

SQL Anywhere サーバ  
文書バージョン: 17 - 2016-05-11

## SQL Anywhere - データベース管理

# 目次

<b>1</b>	<b>SQL Anywhere サーバ - データベース管理</b> .....	<b>5</b>
1.1	データベース接続.....	5
	接続の基本.....	6
	接続パラメータと接続文字列.....	34
	接続パラメータのアルファベット順リスト.....	44
	ODBC データソース.....	111
	認証メカニズム.....	122
	通信プロトコル.....	164
	ネットワークプロトコルオプション.....	177
	データベースおよびデータベースサーバに接続できる限界数.....	237
	接続リスナ.....	241
	接続 ID.....	243
	接続プールによるアプリケーションパフォーマンスの向上.....	244
	詳細情報: テンポラリ接続.....	246
	トラブルシューティング: 接続.....	246
1.2	データベースの作成.....	265
	データベースの作成 (SQL Central の場合).....	266
	データベースの作成 (dbinit ユーティリティの場合).....	267
	データベースファイルのタイプ.....	268
	カラムデータ型の考慮事項.....	302
	カラムの圧縮の考慮事項.....	304
	制約の考慮事項.....	305
1.3	データベースサーバ.....	305
	SQL Anywhere データベースサーバの内部構造.....	307
	ネットワークサーバ (dbsrv17) とパーソナルサーバ (dbeng17) の違い.....	307
	単一のデータベースサーバ上での複数のデータベースの実行.....	309
	データベースサーバの起動方法.....	310
	データベースサーバの停止方法.....	311
	データベースの起動と停止.....	314
	データベースサーバ設定.....	319
	最大ページサイズの考慮事項.....	319
	データベースサーバ名とデータベース名.....	320
	設定ファイルとデータベースサーバの起動オプション.....	322
	データベースサーバのログイン.....	322

	Windows イベントログのメッセージ出力の抑制方法	326
	特殊モード	327
	パフォーマンスとメモリの制御方法	328
	スレッド	330
	サービスまたはデーモンとしてデータベースサーバを実行する方法	339
	OEM Edition の認証アプリケーション	354
	Windows での SQL Anywhere	364
1.4	SQL Anywhere データベースサーバ実行プログラム (dbsrv17, dbeng17)	365
	データベースサーバ起動オプション	375
	データベース起動オプション	519
	トラブルシューティング: データベースサーバの検出方法	539
1.5	データベース設定	539
	環境変数	540
	ファイルロケーションとインストール設定	560
	タイムゾーンの管理	568
	国際言語と文字セット	571
	ログインポリシー	613
	データベースオプション	622
	接続、データベース、データベースサーバのプロパティ	838
	データベースのサイズおよび数に関する物理的な制限	891
1.6	データベースのメンテナンス	894
	データベースのバックアップとリカバリ	894
	データベースの検証	942
	スケジュールとイベントの使用によるタスクの自動化	950
	イベントのトレース	964
	データベースのトラブルシューティング	971
1.7	データベース管理ツールとユーティリティ	980
	Interactive SQL	980
	SQL Central	1041
	ソフトウェアの更新	1062
	データベース管理ユーティリティ	1064
1.8	パフォーマンスの向上、診断、モニタリング	1205
	SQL Anywhere モニタ、コックピット、およびプロファイラの比較	1206
	モニタリング	1206
	パフォーマンス	1349
	診断	1411
1.9	ユーザおよびデータベースのセキュリティ	1453
	ユーザのセキュリティ (ロールと権限)	1453
	データのセキュリティ	1594

---

	トランスポートレイヤセキュリティ.....	1640
1.10	高可用性と読み込み専用のスケールアウトシステム.....	1659
	データベースミラーリング.....	1660
	SQL Anywhere Veritas Cluster Server エージェント.....	1722
	読み込み専用スケールアウト.....	1733
1.11	このマニュアルの印刷、再生、および再配布.....	1765

# 1 SQL Anywhere サーバ - データベース管理

このマニュアルでは、SQL Anywhere データベースを実行、管理、構成する方法について説明します。

データベース接続、データベースサーバ、データベースファイル、バックアッププロシージャ、セキュリティ、高可用性、管理ユーティリティとオプションについて説明します。

このセクションの内容:

## [データベース接続 \[5 ページ\]](#)

データベースに接続するためのツールや方法には、いくつかの種類があります。

## [データベースの作成 \[265 ページ\]](#)

データベースを作成するには、データベースに格納されるテーブル (エンティティ)、各テーブルのカラム (属性)、テーブル間の関係 (キーと制約) を定義します。

## [データベースサーバ \[305 ページ\]](#)

SQL Anywhere データベースサーバには、2 つのバージョンがあります。パーソナルサーバ (dbeng17) と、ネットワークサーバ (dbsrv17) です。

## [SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

データベースサーバを起動して実行します。

## [データベース設定 \[539 ページ\]](#)

データベースを設定するためのツールや方法には、いくつかの種類があります。

## [データベースのメンテナンス \[894 ページ\]](#)

データベースを管理するためのツールや方法には、いくつかの種類があります。

## [データベース管理ツールとユーティリティ \[980 ページ\]](#)

データベースを管理するためのツールや方法には、いくつかの種類があります。

## [パフォーマンスの向上、診断、モニタリング \[1205 ページ\]](#)

モニタリング、データベースパフォーマンスの改善、および問題の診断を行う方法に関するヒントを参照してください。

## [ユーザおよびデータベースのセキュリティ \[1453 ページ\]](#)

データへの不正なアクセスとデータベース上での不正なアクティビティを防ぐために、多くの機能が提供されています。

## [高可用性と読み込み専用のスケールアウトシステム \[1659 ページ\]](#)

データベースミラーリングと Veritas Cluster Server がサポートされており、高可用性が実現しています。

## [このマニュアルの印刷、再生、および再配布 \[1765 ページ\]](#)

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。

## 1.1 データベース接続

データベースに接続するためのツールや方法には、いくつかの種類があります。

このセクションの内容:

#### [接続の基本 \[6 ページ\]](#)

データベースへの接続と切断をすばやく行う方法について説明します。

#### [接続パラメータと接続文字列 \[34 ページ\]](#)

クライアントアプリケーションはデータベースに接続するとき、一連の接続パラメータおよび接続文字列を使用して接続を定義します。

#### [接続パラメータのアルファベット順リスト \[44 ページ\]](#)

接続パラメータは、接続文字列に含めます。

#### [ODBC データソース \[111 ページ\]](#)

Microsoft のオープンデータベースコネクティビティ(ODBC) は、クライアントアプリケーションを Windows ベースのデータベース管理システムに接続するための標準のアプリケーションプログラミングインタフェースです。

#### [認証メカニズム \[122 ページ\]](#)

LDAP、PAM、Kerberos、Microsoft Windows 統合化ログインなどの外部認証メカニズムを使用してデータベースに接続できるよう設定することができます。

#### [通信プロトコル \[164 ページ\]](#)

データベースサーバはさまざまなプロトコルを使用して通信を行います。サポートしているプロトコルの動作をカスタマイズするためのオプションも多数用意されています。

#### [ネットワークプロトコルオプション \[177 ページ\]](#)

ネットワークプロトコルオプションを使用すると、異なるネットワークプロトコルの実装の問題を回避できます。

#### [データベースおよびデータベースサーバに接続できる限界数 \[237 ページ\]](#)

データベースに接続するときは、データベースサーバ経由で接続します。データベースへの同時接続の最大数は、いくつかの要素によって異なります。

#### [接続リスナ \[241 ページ\]](#)

接続リスナを使用すると、特定のネットワークプロトコルやアドレス、ポートを使用してデータベースサーバに接続できます。

#### [接続 ID \[243 ページ\]](#)

ユーザがデータベースに接続すると、データベースサーバはその接続にユニークな接続 ID を割り当てます。

#### [接続プールによるアプリケーションパフォーマンスの向上 \[244 ページ\]](#)

接続プールによって、データベースサーバに対して複数の短時間接続を行うアプリケーションのパフォーマンスが向上する可能性があります。

#### [詳細情報: テンポラリ接続 \[246 ページ\]](#)

データベースサーバでは、テンポラリ接続を使用して、バックアップの実行やデータベースの初期化などの操作が行われます。

#### [トラブルシューティング: 接続 \[246 ページ\]](#)

接続の確立方法を理解すれば、接続の問題の解決に役立ちます。

## 1.1.1 接続の基本

データベースへの接続と切断をすばやく行う方法について説明します。

このセクションの内容:

[データベースへの接続 \(Interactive SQL、SQL Central の場合\) \[7 ページ\]](#)

接続ウィンドウを使用して、ローカルまたはリモートデータベースに接続します。

[ローカルサーバおよびリモートサーバ上のデータベースへの接続 \[10 ページ\]](#)

接続文字列を使用してデータベースに接続します。

[接続されたユーザの管理 \(SQL Central\) \[15 ページ\]](#)

SQL Central を使用して、データベースに接続しているすべてのユーザを管理します。

[接続プロファイル \(SQL Central の場合\) \[19 ページ\]](#)

SQL Central を使用するときの時間の節約と接続プロセスの簡略化のため、接続プロファイルを作成して、各データベースの接続パラメータを保存できます。

[チュートリアル: サンプルデータベースへの接続 \[21 ページ\]](#)

サンプルデータベースを起動して接続し、データやオブジェクトを探索します。

[チュートリアル: SQL Anywhere データベースの作成 \[28 ページ\]](#)

SQL Anywhere サンプルデータベースの Products、SalesOrderItems、SalesOrders、Customers の各テーブルを使用してモデル化した単純なデータベースを SQL Central で作成します。

### 1.1.1.1 データベースへの接続 (Interactive SQL、SQL Central の場合)

接続ウィンドウを使用して、ローカルまたはリモートデータベースに接続します。

#### コンテキスト

このタスクでは、[接続ウィンドウ](#)を使用してデータベースに接続する方法について説明します。[接続ウィンドウ](#)には、実行中のデータベースに接続するオプションとデータベースを起動して接続するオプションがあります。実行中のデータベースへの接続例では、データベースが実行中であることが前提です。

このタスクの一部の選択で、すでに起動しているデータベースへの接続方法について学びます。これらの選択では、データベースがすでに起動していると見なされます。起動していないデータベースでステップを試みた場合、失敗します。

#### 手順

1. [接続ウィンドウ](#)を開きます。
2. 認証ドロップダウンリストで、[データベース](#)をクリックします。
3. [ユーザ ID](#) フィールドと [パスワード](#) フィールドに、ユーザ名とパスワードを入力します。
4. 次のオプションのうちの 1 つを選択してください。

オプション	アクション
ローカルコンピュータのデータベースを起動して接続	<p>このオプションは、ローカルコンピュータ上のパーソナルデータベースサーバでデータベースを起動します。</p> <ol style="list-style-type: none"> <li>1. <b>アクション</b>ドロップダウンリストで、<b>このコンピュータのデータベースを起動して接続</b>をクリックします。</li> <li>2. <b>データベースファイル</b>フィールドで、データベースファイルパス、ファイル名、ファイル拡張子を指定します。たとえば、C:\Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db と入力します。</li> <li>3. 次回以降の接続のために、ファイル名とは異なるデータベース名を作成する場合は、<b>データベース名</b>フィールドに名前を入力します。ファイルパスや拡張子は指定しないでください。</li> <li>4. <b>サーバ名</b>フィールドにデータベースサーバの名前を指定します。たとえば、<b>demo17</b> と入力します。コンピュータシステムでは複数のデータベースサーバを実行できるため、データベースへの接続時には必ず、データベースサーバ名を指定してください (ServerName=server-name)。</li> </ol>
ローカルコンピュータ上で稼働しているデータベースに接続	<p>このオプションは、ローカルコンピュータ上のパーソナルデータベースサーバで実行されているデータベースに接続します。</p> <ol style="list-style-type: none"> <li>1. <b>アクション</b>ドロップダウンリストで、<b>このコンピュータで稼働しているデータベースに接続</b>をクリックし、コンピュータ上で実行されているデータベースに接続します。</li> <li>2. <b>サーバ名</b>フィールドに、実行中のデータベースサーバの名前を入力します。たとえば、<b>demo17</b> と入力します。</li> <li>3. コンピュータシステムでは複数のデータベースサーバを実行できるため、データベースへの接続時には必ず、データベースサーバ名を指定してください (ServerName=server-name)。</li> <li>4. <b>データベース名</b>フィールドにデータベースの名前を入力します。たとえば、<b>demo</b> と入力します。</li> </ol>
別のコンピュータのデータベースに接続	<p>このオプションは、ネットワークデータベースサーバで実行されているデータベースに接続します。</p> <ol style="list-style-type: none"> <li>1. <b>アクション</b>ドロップダウンリストで、<b>別のコンピュータで稼働しているデータベースに接続</b>をクリックし、別のコンピュータ上で実行されているデータベースに接続します。</li> <li>2. <b>ホスト</b>フィールドに、データベースサーバが実行されているコンピュータのホスト名を入力します。</li> <li>3. <b>サーバ名</b>フィールドに、サーバの名前を入力します。たとえば、<b>demo17</b> と入力します。コンピュータシステムでは複数のデータベースサーバを実行できるため、データベースへの接続時には必ず、データベースサーバ名を指定してください (ServerName=server-name)。</li> <li>4. <b>データベース名</b>フィールドにデータベースの名前を入力します。たとえば、<b>demo</b> と入力します。</li> </ol>
ODBC データソースを使用して接続	<p>このオプションは、ODBC データソースを使用してデータベースに接続します。</p> <ol style="list-style-type: none"> <li>1. <b>アクション</b>ドロップダウンリストで、<b>ODBC データソースを使用した接続</b>をクリックします。</li> <li>2. 次のいずれかを実行します。 <ul style="list-style-type: none"> <li>○ <b>ODBC データソース名</b>をクリックし、データソースの名前を入力します。たとえば、<b>SQL Anywhere 17 Demo</b> と入力します (このデータソースでは、パスワード <b>sql</b> も指定する必要があります)。</li> <li>○ <b>ODBC データソースファイル</b>をクリックし、ファイルに格納されているデータソースを参照する FileDataSourceName (FILEDSN) 接続パラメータを入力します。</li> </ul> </li> </ol> <p><b>SQL Anywhere 17 Demo</b> データソースは、データベースサーバとデータベースがまだ稼働していない場合、データベースサーバとデータベースを起動するように設定されています。ODBC データソースがデータベースとデータベースサーバを起動するように設定されていない場合は、ユーザがそれらを起動する必要があります。</p>
接続文字列を使用して接続	<p>このオプションは、データベースに接続します。</p> <ol style="list-style-type: none"> <li>1. <b>アクション</b>ドロップダウンリストで、<b>接続文字列を使用して接続</b>をクリックします。</li> </ol>

オプション	アクション
	<p>ODBC データソースを使用し、接続時に追加のまたは異なるパラメータを指定する場合に、このオプションは便利です。</p> <p>接続文字列と ODBC データソースのどちらも同じ接続パラメータが指定されている場合は、接続文字列の値が使用され、データソースの値は無視されます。</p> <p>2. <b>パラメータフィールド</b>で、"パラメータ=値" のペアをセミコロンで区切ったリストの形式で、接続パラメータを入力します。例:</p> <pre data-bbox="528 562 1473 618">DSN=SQL Anywhere 17 Demo;PWD=sql;ServerName=SampleServer</pre> <p>データベースサーバとデータベースが起動して、管理ツールはデータベースに接続します。データベースサーバの名前は SampleServer です。</p> <p>コンピュータシステムでは複数のデータベースサーバを実行できるため、データベースへの接続時には必ず、データベースサーバ名を指定してください (ServerName=server-name)。</p> <p><i>SQL Anywhere 17 Demo</i> データソースは、データベースサーバとデータベースがまだ稼働していない場合、データベースサーバとデータベースを起動するように設定されています。ODBC データソースがデータベースとデータベースサーバを起動するように設定されていない場合は、ユーザがそれらを起動する必要があります。</p>
別のコンピュータのデータベースを起動して接続	<p>このオプションは、ネットワークデータベースサーバでデータベースを起動した後、データベースに接続します。</p> <p>1. 別のコンピュータ上にあるデータベースサーバを起動します。-gd オプションまたは -gp オプションを指定します。</p> <div data-bbox="528 1061 1473 1200" style="background-color: #fff9c4; padding: 5px;"> <p><b>警告</b></p> <p>データベースサーバオプション -gd、-gp を指定すると、どのユーザでもそのサーバ上でデータベースを起動できるようになります。</p> </div> <p>たとえば、次のコマンドを実行して、データベースサーバを起動します。</p> <pre data-bbox="528 1263 1473 1319">dbsrv17 -n myserver -gd all</pre> <p>2. 1. <b>アクションドロップダウンリスト</b>で、別のコンピュータのデータベースを起動して接続をクリックします。</p> <p>2. <b>データベースファイルフィールド</b>で、データベースサーバに相対的なデータベースファイルのパスと拡張子を入力します。</p> <p>たとえば、<b>C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db</b> と入力します。</p> <p>3. 次回以降の接続のために、ファイル名とは異なるデータベース名を作成する場合は、<b>データベース名フィールド</b>に名前を入力します。ファイルパスや拡張子は指定しないでください。</p> <p>たとえば、<b>mydemo</b> と入力します。</p> <p>4. <b>サーバ名フィールド</b>にサーバの名前を指定します。</p> <p>この例では、上で指定したサーバ名 <b>myserver</b> を使用します。</p> <p>ホストコンピュータ上で複数のデータベースサーバが実行される場合があるので、必ずデータベースサーバ名を指定することをおすすめします。</p> <p>5. <b>ホストフィールド</b>に、データベースサーバが実行されているコンピュータのホスト名を入力します。</p>

5. オプション: このウィンドウで作成される接続文字列をコピーするには、**ツール** ▶ **接続文字列をクリップボードにコピー** ▶ をクリックします。
6. **接続**をクリックします。

## 結果

管理ツールがデータベースに接続します。

## 関連情報

[チュートリアル: サンプルデータベースへの接続 \[21 ページ\]](#)

### 1.1.1.2 ローカルサーバおよびリモートサーバ上のデータベースへの接続

接続文字列を使用してデータベースに接続します。

## 前提条件

データベースサーバに接続すると、実際にはデータベースに接続していることになります。

通常、最初のクライアントが接続する前に、1つ以上のデータベースと同時にデータベースサーバを起動します。たとえば、次のコマンドはデータベースサーバで SQL Anywhere 17 サンプルデータベースを起動します (`server-name` を適切なサーバ名に置き換えます)。

```
dbsrv17 -n server-name "C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

データベースサーバがまだ起動していない場合は、起動する必要があります。サーバコンピュータとクライアントコンピュータが同じ場合、クライアント接続プロセスの一部としてサーバを自動的に起動できます。

同様に、データベースがデータベースサーバでまだ起動していない場合は、起動する必要があります。サーバコンピュータとクライアントコンピュータが同じ場合、クライアント接続プロセスの一部としてデータベースを自動的に起動できます。一定の条件下では、クライアントがリモートサーバコンピュータでデータベースを起動することもできます。

#### ➔ ヒント

コンピュータシステムでは複数のデータベースサーバを実行できるため、データベースへの接続時には必ず、データベースサーバ名を指定してください (`ServerName=server-name`)。

## コンテキスト

次の各選択肢は、固有の接続シナリオを表しています。状況に応じて適切なものを選択してください。Interactive SQL はデモンストレーションに使用されます。

## 手順

- ローカルコンピュータ上で稼働しているデータベースに接続します。

```
dbisql -c "UID=user-id;PWD=password;ServerName=server-name;DBN=database-name"
```

サーバとデータベースの両方がすでに起動している必要があります。

- ローカルコンピュータのデータベースを起動して接続します。

```
dbisql -c "UID=user-id;PWD=password;ServerName=server-name;DBF=database-file"
```

指定されたデータベースサーバがまだ実行されていない場合は、起動されます。データベースがまだ起動していない場合は、サーバで起動されます。

- ネットワークデータベースサーバで実行されているデータベースに接続します。

```
dbisql -c "UID=user-ID;PWD=password;ServerName=server-name;Host=host-name"
```

指定されたサーバがまだ実行されていない場合でも、起動されません。サーバが実行されている場合、サーバで起動された最初のデータベースに接続します。DatabaseName (DBN) を使用して、接続先のデータベースを指定します。

- ODBC データソースを使用して接続します。

```
dbisql -c "DSN=SQL Anywhere 17 Demo;PWD=sql"
```

指定されたデータベースサーバがまだ実行されていない場合は、起動されます。データベースがまだ起動していない場合は、サーバで起動されます。

SQL Anywhere 17 Demo データソースには、サーバ名 (ServerName) とデータベースファイル名 (DBF) が含まれません。

この接続では、ユーザ ID (UID=DBA) はすでにデータソースにあるので指定する必要はありません。パスワードはデータソースに含まれていないため指定する必要があります。

- ODBC データソースと追加の接続パラメータを使用して接続します。

```
dbisql -c "DSN=SQL Anywhere 17 Demo;PWD=sql;ServerName=SampleServer"
```

指定されたデータベースサーバがまだ実行されていない場合は、起動されます。データベースがまだ起動していない場合は、サーバで起動されます。

SQL Anywhere 17 Demo データソースには、サーバ名 (ServerName) とデータベースファイル名 (DBF) が含まれません。サーバ名は、コマンドラインで指定した ServerName 接続パラメータにより上書きされます。

この接続では、ユーザ ID (UID=DBA) はすでにデータソースにあるので指定する必要はありません。パスワードはデータソースに含まれていないため指定する必要があります。

- 別のコンピュータのデータベースを起動して接続します。

- 別のコンピュータ上にあるデータベースサーバを起動します。-gd データベースサーバオプションを指定します。-gp データベースサーバオプションを指定して、4096 より大きいデフォルトページサイズを設定します。指定されたページサイズより大きいページサイズのデータベースファイルを起動することはできません。

```
dbsrv17 -n SampleServer -gd all -gp 8k
```

### ⚠ 警告

データベースサーバオプション `-gd all` を指定すると、どのユーザでもそのサーバ上でデータベースを起動できるようになります。

- b. 次のコマンドを実行します (`host-name` を、データベースサーバを実行しているコンピュータのホスト名に置き換えます)。

```
dbisql -c "Host=host-name;Server=SampleServer;DBF=C:\Users\Public\Documents\SQL Anywhere 17\Samples\demo.db;UID=DBA;PWD=sql"
```

データベースファイルは、サーバコンピュータにあります。

## 結果

Interactive SQL がデータベースに接続します。

このセクションの内容:

[UNIX での接続 \[12 ページ\]](#)

SQL Anywhere をインストールするには、いくつかの基準を満たす必要があります。

### 1.1.1.2.1 UNIX での接続

SQL Anywhere をインストールするには、いくつかの基準を満たす必要があります。

#### Linux

Linux *Applications* メニューをサポートするバージョンの Linux を使用している必要があります。メニュー項目は、SQL Anywhere インストール環境の一部としてインストールされる必要があります。

#### UNIX

SQL Anywhere ユーティリティは、事前にソース化されている必要があります。

#### Mac OS X

SQL Anywhere ユーティリティは、事前にソース化されている必要があります。

### i 注記

#### Mac OS X

Finder で、`/Applications/SQLAnywhere17` の *SQL Central* をダブルクリックします。

#### UNIX

ターミナルセッションで、次のコマンドを実行します。

```
scjview
```

#### Linux

▶ [アプリケーション](#) ▶ [SQL Anywhere17](#) ▶ [管理ツール](#) ▶ [SQL Central](#) ▶ をクリックします。

このセクションの内容:

[UNIX でのサンプルデータベースへの接続 \[13 ページ\]](#)

UNIX 上で SQL Anywhere のサンプルデータベースに接続し、SQL Anywhere を探索します。

[Mac OS X でのサンプルデータベースへの接続 \[14 ページ\]](#)

Mac OS X 上で SQL Anywhere のサンプルデータベースに接続し、SQL Anywhere を探索します。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[セキュリティ: パスワード \[1602 ページ\]](#)

[sa\\_config.sh、sample\\_config32.sh、sample\\_config64.sh のソース指定 \[Unix および Mac OS X\] \[543 ページ\]](#)

### 1.1.1.2.1.1 UNIX でのサンプルデータベースへの接続

UNIX 上で SQL Anywhere のサンプルデータベースに接続し、SQL Anywhere を探索します。

## 前提条件

次の手順では、Interactive SQL を使用してデータベースに接続します。Java Runtime Environment (JRE) がインストールされていることを確認してください。

一般的な UNIX インストールでは、サンプルデータベースを含めた SQL Anywhere ソフトウェアは、書き込みパーミッションのないディレクトリにインストールされます。まず、ターミナルウィンドウで書き込み可能なフォルダに移動します。次のコマンドを実行して、サンプルデータベースをフォルダにコピーします。

```
cp "$SQLANY17/demo.db" .
```

ターミナルウィンドウで次のコマンドを実行して、データベースサーバ上でサンプルデータベースを起動します。

```
dbeng17 demo.db
```

## 手順

1. ターミナルウィンドウで、Interactive SQL を起動します。

```
dbisql
```

デフォルトで、接続ウィンドウが表示されます。

2. 接続ウィンドウに次の情報が表示されます。
  - a. ユーザ ID フィールドに、**DBA** と入力します。
  - b. パスワードフィールドに、**sql** と入力します。
  - c. アクションドロップダウンリストで、このコンピュータで稼働しているデータベースに接続をクリックします。
  - d. サーバ名フィールドに **demo** と入力します。
  - e. オプション: このウィンドウで作成される接続文字列をコピーするには、**ツール** > 接続文字列をクリップボードにコピー をクリックします。
  - f. 接続をクリックします。

## 結果

Interactive SQL が起動し、サンプルデータベースに接続します。

## 関連情報

[UNIX での ODBC データソース \[121 ページ\]](#)

[Mac OS X でのサンプルデータベースへの接続 \[14 ページ\]](#)

### 1.1.1.2.1.2 Mac OS X でのサンプルデータベースへの接続

Mac OS X 上で SQL Anywhere のサンプルデータベースに接続し、SQL Anywhere を探索します。

## 前提条件

- *Finder* で、SQL Anywhere サンプルデータベースを検索して、ファイルを読み込み/書き込みアクセス権がある場所にコピーします。デフォルトでは、サンプルデータベースは `/Applications/SQLAnywhere17/demo.db` にあります。
- データベースサーバを起動します。*Finder* で、*DBLauncher* をダブルクリックします。デフォルトでは、*DBLauncher* は `/Applications/SQLAnywhere17` にあります。
  1. *Database* フィールドで、SQL Anywhere サンプルデータベースのロケーションを参照します (たとえば、`/Applications/SQLAnywhere17/System/demo.db` のように入力します)。
  2. サーバ名フィールドに **demo** と入力します。
  3. *Local Server* をクリックします。  
*Local Server* オプションを選択すると、ネットワーク経由でクライアント/サーバ通信を行えなくなります。
  4. *Start* をクリックして、**demo** という名前のパーソナルデータベースサーバを起動します。

## 手順

1. *Finder* で、/Applications/SQLAnywhere17 にある *Interactive SQL* をダブルクリックします。
2. 接続ウィンドウに次の情報を入力して、SQL Anywhere サンプルデータベースに接続します。
  - 認証ドロップダウンリストで、**データベース**をクリックします。
  - ユーザ ID フィールドに、**DBA** と入力します。
  - パスワードフィールドに、**sql** と入力します。
  - アクションドロップダウンリストで、このコンピュータで稼働しているデータベースに接続をクリックし、コンピュータ上で実行されているデータベースに接続します。
  - サーバ名フィールドに **demo** と入力します。
  - データベース名フィールドに、**demo** と入力します。
  - (オプション) このウィンドウで作成される接続文字列をコピーするには、**ツール** ▶ **接続文字列をクリップボードにコピー** ▶ をクリックします。
  - **接続** をクリックします。

## 結果

Interactive SQL が起動し、サンプルデータベースに接続します。

## 関連情報

[UNIX でのサンプルデータベースへの接続 \[13 ページ\]](#)

[Mac OS X での ODBC データソースの作成 \[116 ページ\]](#)

### 1.1.1.3 接続されたユーザの管理 (SQL Central)

SQL Central を使用して、データベースに接続しているすべてのユーザを管理します。

## 前提条件

DROP CONNECTION、MONITOR、または SERVER OPERATOR システム権限が必要です。

## コンテキスト

SQL Central を使用して、接続中のユーザのプロパティを表示したり、接続を切断したりすることも可能です。

また、SQL Anywhere モニタの [接続](#) ウィジェットからデータベースに接続しているすべてのユーザのリストを表示できます。

## 手順

1. SQL Central では、[SQL Anywhere 17](#) プラグインを使用してデータベースに接続します。
2. データベースに接続している全ユーザのリストを表示します。
  - a. 左ウィンドウ枠のデータベースを選択し、右ウィンドウ枠の [接続](#) タブをクリックします。

このタブは、接続に使用しているアプリケーション (SQL Central、Interactive SQL、カスタムクライアントアプリケーションなど) に関係なく、データベースに現在接続しているすべてのユーザを表示します。
3. ユーザのデータベースへの接続に関するプロパティを調べます。
  - a. [接続](#) タブで、ユーザを右クリックし、[プロパティ](#) をクリックします。
  - b. ユーザのプロパティを確認し、[OK](#) をクリックします。

## 結果

接続しているユーザのプロパティを確認できます。

## 次のステップ

ユーザをデータベースから切断することもできます。

このセクションの内容:

[データベースとの接続の切断 \(SQL Central\) \[17 ページ\]](#)

SQL Central を使用して、ユーザをデータベースから切断します。

[データベースとの接続の切断 \(SQL\) \[18 ページ\]](#)

DISCONNECT 文、または DROP CONNECTION 文を使用して、データベースとの接続を切断します。

## 関連情報

[ウィジェットの管理 \[1220 ページ\]](#)

### 1.1.1.3.1 データベースとの接続の切断 (SQL Central)

SQL Central を使用して、ユーザをデータベースから切断します。

#### 前提条件

自分自身の接続をデータベースから切断するには、権限は必要ありません。

他のユーザを切断する場合は、DROP CONNECTION システム権限が必要です。

#### 手順

データベースに接続します。

オプション	アクション
現在のユーザの接続の切断	▶ <b>ファイル</b> ▶ <b>切断</b> ▶ をクリックします。
他のユーザの接続の切断	右ウィンドウ枠で、 <b>接続タブ</b> をクリックします。 ユーザを右クリックし、 <b>切断</b> を選択します。

#### 結果

ユーザがデータベースから切断されます。

#### 関連情報

[データベースとの接続の切断 \(SQL\) \[18 ページ\]](#)

[モニタを使用した接続の切断 \[1224 ページ\]](#)

## 1.1.1.3.2 データベースとの接続の切断 (SQL)

DISCONNECT 文、または DROP CONNECTION 文を使用して、データベースとの接続を切断します。

### 前提条件

現在のユーザをデータベースから切断するには、権限は必要ありません。

他のユーザを切断する場合は、DROP CONNECTION システム権限が必要です。

### 手順

データベースに接続します。

オプション	アクション
現在のユーザの接続の切断	DISCONNECT 文を実行します。
他のユーザの接続の切断	sa_conn_info システムプロシージャを使用して、接続するユーザの接続 ID を決定します。 DROP CONNECTION 文を実行します。

### 結果

ユーザがデータベースから切断されます。

#### 例

次の文は、Interactive SQL によって現在の接続 conn1 を切断します。

```
DISCONNECT conn1;
```

次の文は、Embedded SQL 内の DISCONNECT の使用法を示します。

```
EXEC SQL DISCONNECT :conn-name
```

次の文は、接続番号 4 を削除します。

```
DROP CONNECTION 4;
```

## 関連情報

[データベースとの接続の切断 \(SQL Central\) \[17 ページ\]](#)

[モニタを使用した接続の切断 \[1224 ページ\]](#)

### 1.1.1.4 接続プロファイル (SQL Central の場合)

SQL Central を使用するときの時間の節約と接続プロセスの簡略化のため、接続プロファイルを作成して、各データベースの接続パラメータを保存できます。

データベースサーバまたはデータベースに最初に接続するときには、ユーザ ID やパスワードなどの接続パラメータを指定します。この情報は、その後の接続時に再び入力する必要があります。SQL Central のデータベースへの接続パラメータを保存する接続プロファイルを作成します。接続プロファイルは、SQL Central に特有のもので、ODBC アプリケーションを開発している場合は、ODBC データソースを使用すれば、接続プロファイルと同様の機能を実現できます。

接続プロファイルを使用したり、管理したりするには、SQL Central で、**接続 > 接続プロファイル** をクリックして、**接続プロファイル** ウィンドウを開きます。**接続プロファイル** ウィンドウを使用して、次の操作を実行します。

- 新しい接続プロファイルの作成
- SQL Central の起動時に自動的に接続するプロファイルの設定  
SQL Central と同じ接続プロファイルを使用することが多い場合は、デフォルトでその接続プロファイルが使用されるように SQL Central を設定できます。SQL Central を起動すると、接続プロファイルを使用してデータベースに自動的に接続します。
- 既存のプロファイルの編集
- 接続プロファイルを使用した接続
- 接続プロファイルの削除
- 接続プロファイルのインポートとエクスポート  
接続プロファイルをファイルにエクスポートすれば、他のコンピュータのユーザと共有したり、バックアップをとったりすることができます。また、他のコンピュータで作成された接続プロファイルを利用して、接続プロファイルのインポートを行うこともできます。

このセクションの内容:

[接続プロファイルの作成 \(SQL Central\) \[20 ページ\]](#)

SQL Central のデータベースへの接続パラメータを保存する接続プロファイルを作成します。

[接続プロファイルを使用した接続 \(SQL Central の場合\) \[21 ページ\]](#)

SQL Central からデータベースサーバやデータベースに接続するには、接続プロファイルを使用します。

## 関連情報

[ODBC データソース \[111 ページ\]](#)

## 1.1.1.4.1 接続プロファイルの作成 (SQL Central)

SQL Central のデータベースへの接続パラメータを保存する接続プロファイルを作成します。

### コンテキスト

ユーザ ID、パスワード (暗号化の有無は不問)、データベースキーを接続プロファイルに保存することはお奨めしません。

### 手順

1. SQL Central で、**接続 > 接続プロファイル** をクリックします。
2. **新規** をクリックします。
3. **名前** フィールドに新しいプロファイルの名前を入力します。
4. **新しい接続プロファイル** をクリックし、ドロップダウンリストから適切なプラグインを選択します。  
既存のプロファイルに基づいて新しい接続プロファイルを作成する場合は、**接続プロファイルをコピー** をクリックし、**既存の接続プロファイル** リストからプロファイルを選択します。
5. (オプション) 他のユーザがプロファイルにアクセスできるようにするには、**この接続プロファイルを他のユーザと共有する** をクリックします。この設定は UNIX などのマルチユーザプラットフォームの場合に便利です。
6. **OK** をクリックします。
7. **接続プロファイルの編集** ウィンドウで、必要な値を入力し、**保存** をクリックしてウィンドウを閉じます。
8. **閉じる** をクリックします。

### 結果

接続プロファイルが作成され、それを使用してデータベースに接続することができます。

### 次のステップ

SQL Central の起動時に、この接続プロファイルを使用して自動的に接続が行われるよう、SQL Central を設定することができます。

### 関連情報

[ODBC データソース \[111 ページ\]](#)

接続プロファイルを使用した接続 (SQL Central の場合) [21 ページ]

## 1.1.1.4.2 接続プロファイルを使用した接続 (SQL Central の場合)

SQL Central からデータベースサーバやデータベースに接続するには、接続プロファイルを使用します。

### 前提条件

接続プロファイルを定義する必要があります。

### 手順

1. SQL Central で、▶ **接続** ▶ **接続プロファイル** ▶ をクリックします。

接続プロファイルウィンドウが表示されます。

2. プロファイルを選択し、**接続**をクリックします。

#### → ヒント

SQL Central が接続プロファイルを使用して自動的に接続を行うようにするには、**スタートアップの設定**をクリックし、**起動時に使用カラム**をはいに変更します。

3. **閉じる**をクリックします。

### 結果

接続すると、SQL Central のメインウィンドウにアイコンが表示されます。

## 1.1.1.5 チュートリアル: サンプルデータベースへの接続

サンプルデータベースを起動して接続し、データやオブジェクトを探索します。

### 前提条件

デフォルトの DBA ユーザ名は `dba`、パスワードは `sql` です。

## コンテキスト

SQL Anywhere サンプルデータベースは、限られた種類のスポーツ衣料品を製造する小企業を想定して作られています。データベースには、この企業の内部情報（従業員、部署、経理データ）、製品情報（製品）、販売情報（受注、顧客、連絡先）が入っています。サンプルデータベースに入っている情報は、すべて架空のものです。

1. [レッスン 1: サンプルデータベースへの接続 \(SQL Anywhere 17 Demo\) \[22 ページ\]](#)  
データベースサーバを起動してサンプルデータベースに接続し、データベースサーバメッセージウィンドウを表示してデータベースサーバに関する情報を確認します。
2. [レッスン 2: データベースへの接続 \(SQL Central の場合\) \[25 ページ\]](#)  
サンプルデータベースに接続し、マニュアルに記載されたさまざまな例を試してみます。
3. [レッスン 3: サンプルデータベースへの接続 \(SQL の場合\) \[26 ページ\]](#)  
コマンドプロンプトを使用して、Interactive SQL を起動し、サンプルデータベースに接続します。
4. [レッスン 4: データベースサーバの停止 \(dbstop ユーティリティ\) \[27 ページ\]](#)  
Stop Server ユーティリティを使用して、データベースサーバを停止します。

### 1.1.1.5.1 レッスン 1: サンプルデータベースへの接続 (SQL Anywhere 17 Demo)

データベースサーバを起動してサンプルデータベースに接続し、データベースサーバメッセージウィンドウを表示してデータベースサーバに関する情報を確認します。

## 前提条件

このチュートリアル冒頭に一覧されているロールと権限を持っている必要があります。

## コンテキスト

サンプルデータベースでは、次のデフォルトのユーザ ID とパスワードを使用してください。

- ユーザ ID: DBA
- パスワード: sql (SQL Anywhere のパスワードは大文字と小文字が区別されます。)

サンプルデータベースでは、ODBC データソース [SQL Anywhere 17 Demo](#) を使用します。

- ユーザ ID は SQL Anywhere 17 Demo データソースにすでに含まれているので入力する必要はありませんが、パスワード `sql` は指定する必要があります。  
運用環境ではパスワードを ODBC データソースに格納しないでください。
- SQL Anywhere 17 Demo データソースがデータベースサーバを自動的に開始するので、接続の前にデータベースサーバが実行中でなくともかまいません。

## 手順

1. Interactive SQL で、**接続**ウィンドウを開きます。

▶ **スタート** ▶ **プログラム** ▶ **SQL Anywhere17** ▶ **管理ツール** ▶ **Interactive SQL** ▶ をクリックします。

▶ **SQL** ▶ **接続** ▶ をクリックします。

2. 認証ドロップダウンリストで、**データベース**をクリックします。
3. **ユーザ ID** フィールドに、**dba** と入力します。**パスワード**フィールドに、**sql** と入力します。
4. 次のオプションのうちの 1 つを選択してください。

アクション	説明
デフォルトの ODBC データソースを使用した接続	<p>SQL Anywhere 17 Demo ODBC データソースは、データベースサーバとデータベースがまだ稼働していない場合、データベースサーバとデータベースを起動するように設定されています。</p> <ol style="list-style-type: none"> <li>1. <b>アクション</b>ドロップダウンリストで、<b>ODBC データソースを使用した接続</b>をクリックします。</li> <li>2. 次のいずれかを実行します。 <ul style="list-style-type: none"> <li>○ <b>ODBC データソース名</b>をクリックし、<b>SQL Anywhere 17 Demo</b> と入力します。</li> </ul> </li> <li>3. (オプション)このウィンドウで作成される接続文字列をコピーするには、▶ <b>ツール</b> ▶ <b>接続文字列をクリップボードにコピー</b> ▶ をクリックします。</li> <li>4. <b>接続</b>をクリックします。</li> </ol>
Interactive SQL を起動してサンプルデータベースに接続	<p>次のコマンドを実行します。</p> <pre>dbisql -c "UID=DBA;PWD=sql;DBF=%SQLANYAMP17%¥demo.db"</pre> <p>新しい Interactive SQL ウィンドウが開きます。</p>
データベースサーバを起動してサンプルデータベースに接続	<p><b>接続</b>ウィンドウで、サンプルデータベースに接続するための情報を以下のフィールドに入力します。</p> <ol style="list-style-type: none"> <li>1. <b>アクション</b>ドロップダウンリストで、<b>このコンピュータで稼働しているデータベースを起動して接続</b>を選択します。<b>サーバ名</b>フィールドに <b>demo17</b> と入力します。</li> <li>2. <b>ファイル名</b>フィールドに、<b>%SQLANYWHERE17%¥demo.db</b> と指定します。</li> <li>3. <b>接続</b>をクリックします。</li> </ol>

Windows の場合、データベースサーバはシステムトレイにアイコンとして表示されます。これで、サンプルデータベースを実行するデータベースサーバが正常に起動されたこととなります。ただし、データベース内のデータを表示したり操作したりすることはまだできません。データベースサーバのアイコンが単に表示されているだけです。

5. システムトレイでデータベースサーバアイコンをダブルクリックして、データベースサーバメッセージウィンドウを表示します。

データベースサーバメッセージウィンドウには、次の情報が表示されます。

## サーバ名

タイトルバーに表示されている名前 (この例では demo 17) がサーバ名です。このチュートリアルでは、-n サーバオプションを使用してサーバ名を割り当てています。サーバ名を指定しない場合は、最初に起動されたデータベースの名前になります。この名前は、アプリケーションがデータベースに接続するときに使用します。

## バージョンとビルド番号

サーバ名に続く数字は、バージョン番号とビルド番号です。バージョン番号は SQL Anywhere の特定のリリースを表し、ビルド番号はコンパイル済みソフトウェアの特定のインスタンスを表します。

## 起動情報

データベースサーバは、起動時に、データベース要求を処理するときに使用するメモリを別に設定します。これをキャッシュと呼びます。キャッシュメモリの量は、このウィンドウに表示されます。キャッシュは固定サイズのページで構成されていますが、このページのサイズもウィンドウ内に表示されます。この例では、起動時のキャッシュサイズとページのサイズはデフォルト値になっています。このチュートリアルの場合も含め、多くの場合デフォルトの起動オプションが適しています。

## データベース情報

データベースファイルの名前とそのトランザクションログファイルがウィンドウに表示されます。

## 結果

サンプルデータベースを実行するデータベースサーバが起動し、データベースサーバメッセージウィンドウが表示されます。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: サンプルデータベースへの接続 \[21 ページ\]](#)

次のタスク: [レッスン 2: データベースへの接続 \(SQL Central の場合\) \[25 ページ\]](#)

## 関連情報

[データベースサーバの起動方法 \[310 ページ\]](#)

[データベースサーバ \[305 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

## 1.1.1.5.2 レッスン 2: データベースへの接続 (SQL Central の場合)

サンプルデータベースに接続し、マニュアルに記載されたさまざまな例を試してみます。

### 前提条件

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に記載されているロールと権限を持っている必要があります。

### コンテキスト

SQL Central は、データベースサーバ、データベース、そこに含まれるオブジェクトを管理するため、SQL 文とコマンドラインユーティリティの代わりとして使用できるグラフィックツールです。

### 手順

1. SQL Central を起動します。▶ **スタート** ▶ **プログラム** ▶ **SQL Anywhere17** ▶ **管理ツール** ▶ **SQL Central** ▶ をクリックします。
2. ▶ **接続** ▶ **SQL Anywhere17に接続** ▶ をクリックします。
3. サンプルデータベースに接続するための情報を以下のフィールドに入力します。
  - a. **ユーザ ID** 項目に、**DBA** と入力します。
  - b. **パスワード** 項目に、**sql** と入力します。
  - c. **アクションドロップダウンリスト**で、**このコンピュータで稼働しているデータベースに接続**を選択します。
  - d. **サーバ名**フィールドに **demo17** と入力します。

コンピュータシステムでは複数のデータベースサーバを実行できるため、データベースへの接続時には必ず、データベースサーバ名を指定してください (ServerName=server-name)。

4. (オプション) このウィンドウで作成される接続文字列をコピーするには、▶ **ツール** ▶ **接続文字列をクリップボードにコピー** ▶ をクリックします。
5. **接続** をクリックします。SQL Central がデータベースに接続します。
6. (オプション) Interactive SQL から Customers テーブルへのクエリを実行します。
  - a. Interactive SQL を開きます。SQL Central で、▶ **ファイル** ▶ **Interactive SQL を開く** ▶ をクリックします。
  - b. Interactive SQL ウィンドウで、次の SQL 文を入力します。

```
SELECT * FROM Customers;
```

- c. 文を実行します。F5 キーを押すか、または ▶ **SQL** ▶ **実行** ▶ をクリックします。

結果ウィンドウ枠に結果セットが表示されます。

## 結果

SQL Central が起動し、サンプルデータベースに接続します。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: サンプルデータベースへの接続 \[21 ページ\]](#)

前のタスク: [レッスン 1: サンプルデータベースへの接続 \(SQL Anywhere 17 Demo\) \[22 ページ\]](#)

次のタスク: [レッスン 3: サンプルデータベースへの接続 \(SQL の場合\) \[26 ページ\]](#)

### 1.1.1.5.3 レッスン 3: サンプルデータベースへの接続 (SQL の場合)

コマンドプロンプトを使用して、Interactive SQL を起動し、サンプルデータベースに接続します。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

次のコマンドを実行して、Interactive SQL を起動し、サンプルデータベースに接続します。

```
dbisql -c "UID=DBA;PWD=sql;DBF=%SQLANY%SAMP17%¥demo.db"
```

## 結果

Interactive SQL が起動し、サンプルデータベースに接続します。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: サンプルデータベースへの接続 \[21 ページ\]](#)

前のタスク: [レッスン 2: データベースへの接続 \(SQL Central の場合\) \[25 ページ\]](#)

次のタスク: [レッスン 4: データベースサーバの停止 \(dbstop ユーティリティ\) \[27 ページ\]](#)

### 1.1.1.5.4 レッスン 4: データベースサーバの停止 (dbstop ユーティリティ)

Stop Server ユーティリティを使用して、データベースサーバを停止します。

#### 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

#### コンテキスト

Windows では、データベースサーバメッセージウィンドウでシャットダウンをクリックしてデータベースサーバを停止できます。

#### 手順

1. 次のコマンドを実行して、サンプルデータベースを実行するデータベースサーバを停止します。

```
dbstop demo17 -c "uid=dba;pwd=sql"
```

2. データベースに接続があるけれども停止を実行するかどうか、と尋ねるプロンプトが表示されたら、**y** と入力します。

#### 結果

データベースサーバが停止し、サンプルデータベースへの接続はすべて終了します。

タスクの概要: [チュートリアル: サンプルデータベースへの接続 \[21 ページ\]](#)

前のタスク: [レッスン 3: サンプルデータベースへの接続 \(SQL の場合\) \[26 ページ\]](#)

## 関連情報

[データベースサーバ \[305 ページ\]](#)

[サーバ停止ユーティリティ \(dbstop\) \[1163 ページ\]](#)

### 1.1.1.6 チュートリアル: SQL Anywhere データベースの作成

SQL Anywhere サンプルデータベースの Products、SalesOrderItems、SalesOrders、Customers の各テーブルを使用してモデル化した単純なデータベースを SQL Central で作成します。

## 前提条件

新しいデータベースでは、CREATE ANY OBJECT システム権限が必要です。

1. [レッスン 1: データベースファイルの作成 \[28 ページ\]](#)  
データを保持するデータベースファイルを作成します。
2. [レッスン 2: データベースへのテーブルの追加 \[30 ページ\]](#)  
テーブルを作成して、データベースに情報を格納します。
3. [レッスン 3: カラムへの NOT NULL 制約の設定 \[32 ページ\]](#)  
NOT NULL 制約をカラムに追加して、カラムで NULL 値が許可されないようにします。
4. [レッスン 4: 外部キーの作成 \[33 ページ\]](#)  
外部キーを作成して、テーブル間の関係を作成します。

#### 1.1.1.6.1 レッスン 1: データベースファイルの作成

データを保持するデータベースファイルを作成します。

## 前提条件

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## コンテキスト

データベースファイルには、システムテーブルと、すべてのデータベースに共通するその他のシステムオブジェクトが格納されます。

## 手順

1. SQL Central を起動します。▶ [スタート](#) ▶ [プログラム](#) ▶ [SQL Anywhere17](#) ▶ [管理ツール](#) ▶ [SQL Central](#) ▶ をクリックします。
2. データベース作成ウィザードを起動します。▶ [ツール](#) ▶ [SQL Anywhere17](#) ▶ [データベースの作成](#) ▶ をクリックします。
  - a. ようこそページで、[次へ](#)をクリックします。
  - b. このコンピュータにデータベースを作成を[クリック](#)し、[次へ](#)をクリックします。
  - c. [メインデータベースファイルを保存フィールド](#)に、`c:¥temp¥mysample.db` と入力します。

テンポラリディレクトリとして `c:¥temp` 以外のディレクトリを使用する場合は、正しいパスを指定します。
  - d. [DBA ユーザとパスワードの指定](#)ページで、DBA ユーザのユーザ ID とパスワードを作成します。
  - e. [完了](#)をクリックしてから、[閉じる](#)をクリックします。

自動的に DBA ユーザとしてデータベースに接続されます。

## 結果

データベースが作成され、そのデータベースに接続します。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: SQL Anywhere データベースの作成 \[28 ページ\]](#)

次のタスク: [レッスン 2: データベースへのテーブルの追加 \[30 ページ\]](#)

## 関連情報

[データベースの作成 \(dbinit ユーティリティの場合\) \[267 ページ\]](#)

[データベースの作成 \(SQL Central の場合\) \[266 ページ\]](#)

## 1.1.1.6.2 レッスン 2: データベースへのテーブルの追加

テーブルを作成して、データベースに情報を格納します。

### 前提条件

このチュートリアルの前までのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

### 手順

1. **テーブル作成ウィザード**を起動します。SQL Central の左ウィンドウ枠で、**テーブル**を右クリックし、**新規** > **テーブル** をクリックします。
  - a. **新しいテーブルの名前を指定してください**というフィールドに **Products** と入力します。
  - b. **完了**をクリックします。

データベースサーバによって、デフォルト設定を使用してテーブルが作成され、右ウィンドウ枠に**カラム**タブが表示されます。新しいカラムの**名前**フィールドが選択され、新しいカラムの名前を指定するよう求められます。

2. 新しいカラムの名前として、**ProductID** と入力します。

このカラムがテーブル内の最初のカラムなので、テーブルのプライマリキーであることを示す**プライマリキー**が選択されます。

3. **データ型**リストで、**integer** をクリックします。
4. ... ボタンをクリックします。
5. **値**タブをクリックし、**デフォルト値** > **システム定義** > **autoincrement** をクリックします。

AUTOINCREMENT 値は、テーブルにローが追加されるごとに増加します。この値により、カラムにはユニークな値が設定されます。これはプライマリキーの要件です。

6. **OK** をクリックします。
7. **ファイル** > **新規** > **カラム** をクリックします。
  - a. **名前**フィールドに **ProductName** と入力します。
  - b. **データ型**リストで、**char** をクリックします。
  - c. **サイズ**リストで、**15** をクリックします。
8. 前述の手順を繰り返して、次のテーブルをデータベースに追加します。

#### Customers テーブル

テーブル **Customers** を追加し、テーブル内に以下のカラムを作成します。

#### CustomerID

各顧客の ID 番号です。**プライマリキー**が選択されていることを確認し、**データ型**を **integer** に、**デフォルト値**を **autoincrement** にそれぞれ設定します。

### CompanyName

各会社の名前です。データ型を *char* に設定し、最大長を 35 文字にします。

### SalesOrders テーブル

テーブル **SalesOrders** を追加し、テーブル内に以下のカラムを作成します。

#### SalesOrderID

各注文書の ID 番号です。プライマリキーが選択されていることを確認し、データ型を *integer* に、デフォルト値を *autoincrement* にそれぞれ設定します。

#### OrderDate

注文日です。データ型を *date* に設定します。

#### CustomerID

発注した顧客の ID 番号です。データ型を *integer* に設定します。

### SalesOrderItems テーブル

テーブル **SalesOrderItems** を追加し、テーブル内に以下のカラムを作成します。

#### SalesOrderItemsID

項目が含まれている注文書の ID 番号です。プライマリキーが選択されていることを確認し、データ型を *integer* に設定します。

#### LineID

各注文書の ID 番号です。プライマリキーが選択されていることを確認し、データ型を *integer* に設定します。

プライマリキーは、SalesOrderItemsID と LineID の両方に対して設定されているため、テーブルのプライマリキーは、この 2 つのカラムの連結値で構成されます。

#### ProductID

受注製品の ID 番号です。データ型を *integer* に設定します。

9.  **ファイル** > **保存**  をクリックします。

## 結果

データベースに 4 つのテーブルが作成されました。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: SQL Anywhere データベースの作成 \[28 ページ\]](#)

前のタスク: [レッスン 1: データベースファイルの作成 \[28 ページ\]](#)

次のタスク: [レッスン 3: カラムへの NOT NULL 制約の設定 \[32 ページ\]](#)

## 関連情報

[ER \(実体関連\) 図の表示 \[1057 ページ\]](#)

### 1.1.1.6.3 レッスン 3: カラムへの NOT NULL 制約の設定

NOT NULL 制約をカラムに追加して、カラムで NULL 値が許可されないようにします。

## 前提条件

このチュートリアル of これまでのレッスンを完了している必要があります。

このチュートリアル of 冒頭に一覧されているロールと権限を持っている必要があります。

## コンテキスト

デフォルトではカラムで NULL 値が許可されますが、NULL 値を許可する理由がないかぎり、カラムで NOT NULL を宣言するようにしてください。

## 手順

1. SQL Central の左ウィンドウ枠で、**テーブル**をダブルクリックします。
2. *Products* をクリックし、右ウィンドウ枠で**カラム**タブをクリックします。
3. *ProductName* カラムをクリックします。
4. **ファイル** > **プロパティ** をクリックします。
5. **制約**タブをクリックし、**NULL 値を禁止**をクリックします。
6. **OK** をクリックします。
7. **ファイル** > **保存** をクリックします。

## 結果

この制約により、*Products* テーブルに追加されたローごとに、*ProductName* カラムに値が必要になります。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: SQL Anywhere データベースの作成 \[28 ページ\]](#)

前のタスク: [レッスン 2: データベースへのテーブルの追加 \[30 ページ\]](#)

次のタスク: [レッスン 4: 外部キーの作成 \[33 ページ\]](#)

## 関連情報

[制約の考慮事項 \[305 ページ\]](#)

### 1.1.1.6.4 レッスン 4: 外部キーの作成

外部キーを作成して、テーブル間の関係を作成します。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

1. SQL Central の左ウィンドウ枠で、[テーブル](#)をダブルクリックします。
2. *SalesOrderItems* テーブルをクリックします。
3. 右ウィンドウ枠で、[制約タブ](#)をクリックします。
4. [外部キー作成ウィザード](#)を起動します。▶ [ファイル](#) ▶ [新規](#) ▶ [外部キー](#) ▶ をクリックします。

- a. この外部キーが参照するテーブルを指定してくださいというリストで、*Products* テーブルをクリックします。
- b. 新しい外部キーの名前を指定してくださいというフィールドに **ProductIDkey** と入力します。
- c. 次へをクリックします。
- d. この外部キーが参照する主キー制約または一意性制約を指定してくださいというフィールドで、**プライマリキー** をクリックします。
- e. **外部カラム** リストで、*SalesOrderItemsID* をクリックします。
- f. **完了** をクリックします。

## 結果

SalesOrderItems テーブルの外部キーが作成されました。

これで、データベースの作成のチュートリアルを完了しました。

## 次のステップ

データベースに情報を挿入してください。

タスクの概要: [チュートリアル: SQL Anywhere データベースの作成 \[28 ページ\]](#)

前のタスク: [レッスン 3: カラムへの NOT NULL 制約の設定 \[32 ページ\]](#)

## 1.1.2 接続パラメータと接続文字列

クライアントアプリケーションはデータベースに接続するとき、一連の接続パラメータおよび接続文字列を使用して接続を定義します。

接続パラメータは、データベースサーバ名、データベース名、ユーザ ID などの情報を指定します。

各接続パラメータでは、キーワードと値の組み合わせ (`parameter=value` の形式) を指定します。次の例では、パスワード接続パラメータ、PWD を指定しています。

```
PWD=passwd
```

### 接続文字列として渡される接続パラメータ

接続パラメータを組み合わせて接続文字列を作ります。接続文字列では、次のように各接続パラメータをセミコロンで区切りません。

```
parameter1=value1;parameter2=value2;...
```

例:

```
UID=DBA;PWD=passwd;Host=myhost;DatabaseName=mydemo;ServerName=myserver
```

通常、アプリケーションが構築してインタフェースライブラリに渡す接続文字列は、ユーザが情報を入力する方法と直接は対応していません。ユーザが接続ウィンドウの複数のフィールドから入力することもあれば、アプリケーションが初期化ファイルから接続情報を読み込むこともあります。

ユーティリティの多くは、接続文字列を `-c` オプションとして認識し、変更を加えないでインタフェースライブラリに渡します。次の例は、バックアップユーティリティ (dbbackup) の典型的なコマンドラインを示します。

```
dbbackup -c "HOST=myhost-pc;DBN=mydemo;UID=DBA;PWD=passwd;ServerName=myserver"  
SQLAnybackup
```

アプリケーションでは、接続文字列は、各パラメータ設定をセミコロンで区切って入力する必要があります。

このセクションの内容:

#### [一般的な接続シナリオ \[35 ページ\]](#)

データベースへの効率的な接続方法を理解するため、一般的な接続シナリオをいくつか取り上げます。

#### [接続パラメータの構文ルール \[39 ページ\]](#)

接続パラメータの構文にはいくつかのルールが適用されます。

#### [組み込みデータベースで使用する接続パラメータ \[41 ページ\]](#)

組み込みデータベースは、通常単一アプリケーションで使用するために設計されており、アプリケーションと同じコンピュータ上で稼働し、ユーザからは見えないようになっています。

#### [データベースユーティリティによる接続パラメータの取得方法 \[43 ページ\]](#)

すべてのデータベースユーティリティで、データベースサーバとの通信に Embedded SQL が使用されます。

#### [ODBC データソースでの接続パラメータ \[43 ページ\]](#)

ODBC データソースを使用してデータベースに接続することができます。

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[ネットワークプロトコルオプション \[177 ページ\]](#)

[文字セット変換 \[581 ページ\]](#)

### 1.1.2.1 一般的な接続シナリオ

データベースへの効率的な接続方法を理解するため、一般的な接続シナリオをいくつか取り上げます。

データベース接続に必要な接続パラメータは、複数のデータベースサーバが同じコンピュータで実行されているか、複数のデータベースが同じデータベースサーバで実行されているか、クライアントアプリケーションがデータベースサーバとして同じコンピュータで実行されているかによって異なります。

## i 注記

1つのコンピュータでは、複数のデータベースサーバを同時に実行できます。したがって、データベースサーバに対して認証を行う際には、必ずデータベースサーバ名を指定することをお奨めします (ServerName=server-name)。

各データベースサーバでは、複数のデータベースを同時に実行できます。したがって、データベースサーバに対して認証を行う際には、必ずデータベース名を指定することをお奨めします (DatabaseName=database-name)。

次の表は、いくつかの一般的なシナリオと、それぞれの場合に必要なクライアント接続パラメータを示します。これらの接続パラメータを組み合わせることで、接続先となるデータベースを正しく識別できます。次のリストの他にも、一般的ではない接続シナリオに対応するため、多くの接続パラメータがサポートされています。

## 共有メモリによるローカルサーバ接続

表中の X は、一番上の段に記載されたシナリオに必要な接続パラメータを示しています。

接続パラメータ	1つのサーバが1つのデータベースをホスト	1つのサーバが複数のデータベースをホスト	複数のサーバがそれぞれ1つのデータベースをホスト	複数のサーバがそれぞれ複数のデータベースをホスト
UserID (UID)	X	X	X	X
Password (PWD)	X	X	X	X
DatabaseName (DBN)		X		X
ServerName(Server)	1	1	X	X
Host <sup>2</sup>	1	1		

<sup>1</sup> -xd オプションを使用してサーバを実行している場合、サーバ名 (ServerName=server-name) が必要です。

<sup>2</sup> ホスト接続パラメータ (Host=host-name:port-number) が使用された場合、共有メモリではなく TCP/IP を使用した接続が試行されます。パーソナルデータベースサーバの場合、-x オプションで TCP/IP 通信プロトコルを有効にしていない限り、このオプションは適切ではありません。また、TCP/IP を使用してパーソナルデータベースサーバに接続する場合は、ホスト名 localhost を使用する必要があります。

## ネットワークサーバへの接続

表中の X は、一番上の段に記載されたシナリオに必要な接続パラメータを示しています。

接続パラメータ	1つのサーバが1つのデータベースをホスト	1つのサーバが複数のデータベースをホスト	複数のサーバがそれぞれ1つのデータベースをホスト	複数のサーバがそれぞれ複数のデータベースをホスト
UserID (UID)	X	X	X	X
Password (PWD)	X	X	X	X
DatabaseName (DBN)		X		X

接続パラメータ	1つのサーバが1つのデータベースをホスト	1つのサーバが複数のデータベースをホスト	複数のサーバがそれぞれ1つのデータベースをホスト	複数のサーバがそれぞれ複数のデータベースをホスト
ServerName(Server)	3	3	X	X
Host	X <sup>3</sup>	X <sup>3</sup>	X	X

<sup>3</sup> サーバがデフォルトの TCP/IP ポート (2638) を使用していない場合には、ポート番号 (Host=host-name:port-number) またはサーバ名 (ServerName=server-name) が必要です。デフォルト以外のポート番号を使用するデータベースサーバを Mac OS X 上で実行している場合や、-sb オプションを使用して実行している場合には、接続のためのポート番号 (Host=host-name:port-number) を指定する必要があります。

### 例

ただ1つのデータベースサーバでただ1つのデータベースが実行されており、クライアントアプリケーションと同じコンピュータで実行されている場合。

```
dbisql -c "UID=DBA;PWD=sql"
```

ただ1つのデータベースサーバで複数のデータベースが実行されており、クライアントアプリケーションと同じコンピュータで実行されている場合。

```
dbisql -c "UID=DBA;PWD=sql;DBN=demo"
```

複数のデータベースサーバでそれぞれ1つのデータベースが実行されており、クライアントアプリケーションと同じコンピュータで実行されている場合。

```
dbisql -c "UID=DBA;PWD=sql;ServerName=demo17"
```

複数のデータベースサーバでそれぞれ複数のデータベースが実行されており、クライアントアプリケーションと同じコンピュータで実行されている場合。

```
dbisql -c "UID=DBA;PWD=sql;ServerName=demo17;DBN=demo"
```

ただ1つのデータベースサーバでただ1つのデータベースが実行されており、クライアントアプリケーションとは異なるコンピュータで実行されていて、データベースサーバを実行しているコンピュータは myhost-pc という名前となります。

```
dbisql -c "UID=DBA;PWD=sql;Host=myhost-pc"
```

複数のデータベースサーバで複数のデータベースが実行されており、クライアントアプリケーションとは異なるコンピュータで実行されていて、データベースサーバを実行しているコンピュータは myhost-pc という名前となります。

```
dbisql -c "UID=DBA;PWD=sql;Host=myhost-pc;ServerName=demo17;DBN=demo"
```

データベースサーバまたはデータベースが実行中かどうかはわからず、それらを接続できるように、クライアントアプリケーションと同じコンピュータ上で起動したいと思っている場合。

```
dbisql -c "UID=DBA;PWD=sql;ServerName=demo17;DBF=%SQLANYSAMPL7%demo.db"
```

このセクションの内容:

#### [デフォルト接続パラメータ \[38 ページ\]](#)

いくつかの接続パラメータを指定しないで、デフォルトの動作で接続を確立できます。

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[Userid \(UID\) 接続パラメータ \[109 ページ\]](#)

[Password \(PWD\) 接続パラメータ \[99 ページ\]](#)

[DatabaseName \(DBN\) 接続パラメータ \[68 ページ\]](#)

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[-xd データベースサーバオプション \[504 ページ\]](#)

### 1.1.2.1.1 デフォルト接続パラメータ

いくつかの接続パラメータを指定しないで、デフォルトの動作で接続を確立できます。

ただし、アプリケーションが他の SQL Anywhere アプリケーションとともにインストールされている場合は、運用環境でデフォルトの動作を使用すると問題が発生する場合があります。接続で問題がある場合、デフォルトに依存するのではなく、明示的に値を指定することで問題を解決できる場合があります。

組み込みの SQL Anywhere サーバを実行している場合は特に、常に ServerName 接続パラメータを使用してサーバ名を指定することをお奨めします。

#### デフォルトを使用しない場合 (推奨方法)

次の接続文字列を使用して、名前を指定したデータベースで、名前を指定したローカルサーバに接続します。

```
ServerName=server-name;DBN=db-name;UID=user-id;PWD=password
```

#### デフォルトのデータベースサーバとデータベース

ローカルコンピュータでただ 1 つのデータベースサーバだけが実行中であることが確実なときに、接続にデフォルトパラメータを使用します。

```
UID=user-id;PWD=password
```

## デフォルトのデータベースサーバ

ローカルコンピュータでただ1つのデータベースサーバだけが実行中であることが確実なときには、デフォルトサーバ設定を使用して、接続するデータベースを指定できます。

```
DBN=db-name;UID=user-id;PWD=password
```

## デフォルトのデータベース

ローカルコンピュータ上で複数のデータベースサーバが実行中である場合、接続する先のサーバを指定します。そのデータベースサーバに1つのデータベースだけが実行中である場合は、データベース名を指定する必要はありません。次の接続文字列は、指定されたサーバに接続して、デフォルトデータベースを使用します。

```
ServerName=server-name;UID=user-id;PWD=password
```

## ネットワークコンピュータのデフォルトのデータベースサーバ

次の接続文字列は、ネットワークコンピュータ上で実行中のデフォルトデータベースサーバで実行中の、指定のデータベースに接続します。この接続文字列は、データベースサーバがデフォルトの TCP/IP ポート 2638 上で実行中であると想定しています。

```
Host=host-name;DBN=db-name;UID=user-id;PWD=password
```

データベースサーバがデフォルトの TCP/IP ポート (2638) を使用していない場合は、データベースサーバ名 (ServerName=server-name) またはポート番号 (Host=host-name:port-number) を指定する必要があります。サーバ名を指定すると、サーバが Mac OS X 上にある場合や、またはデータベースサーバが -sb オプションを使って起動された場合には、接続できない可能性があります。このような場合には、ポート番号 (Host=host-name:port-number) を指定して接続する必要があります。

## 関連情報

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

### 1.1.2.2 接続パラメータの構文ルール

接続パラメータの構文にはいくつかのルールが適用されます。

スペースが含まれた接続文字列

コマンドラインで接続文字列を指定するときに、接続パラメータ値にスペースが含まれている場合は接続文字列全体を二重引用符で囲む必要があります。

## ブール値

ブール (true または false) 引数は、true の場合は YES、ON、1、TRUE、Y、T のいずれかであり、false の場合は NO、OFF、0、FALSE、N、F のいずれかです。

## 大文字と小文字の区別

値に大文字と小文字が区別されるもの (UNIX のファイル名など) が含まれている場合でも、接続パラメータ名は大文字と小文字を区別しません。

## 文字セット制限

データベースサーバ名 (ServerName 接続パラメータで指定) は、32 ~ 126 の範囲の ASCII 文字セットで構成してください。この制限は、他の接続パラメータの値には適用されません。

## 優先度

### 警告

接続文字列内では 1 つのパラメータを複数回指定しないでください。パラメータを複数回指定した場合、結果は保証されず、一貫しなくなる可能性があります。その上、動作はソフトウェアの将来のリリースで変更する可能性があります。

次のルールによって、接続パラメータの優先度が決まります。

- 接続文字列のエントリは、左から右に読み込まれます。同じ接続パラメータが複数回指定された場合は、文字列の最後のパラメータを適用します。ODBC と OLE DB のクライアントアプリケーション (SQL Central と Interactive SQL を除く) はこの動作の例外です。これらに対して同じパラメータが複数回指定された場合は、文字列の最初の値が適用されます。
- 順番に、インタフェースライブラリは、次のソースから接続パラメータを取得します。

#### 接続文字列

パラメータを接続文字列に明示的に渡すことができます。

#### SQLCONNECT 環境変数

SQLCONNECT 環境変数に接続文字列を格納できます。

#### データソース

ODBC データソースに接続パラメータを格納できます。

接続文字列と ODBC データソースがどちらも同じ接続パラメータを指定する場合、接続文字列の値が使用され、データソースからの値は無視されます。

たとえば、データベースサーバ demo17 がデータベース demo および demo2 を実行中であるとして、次のコマンドは demo データベースのデータソースを作成します。

```
dbdsn -w demodsn -c "ServerName=demo17;Host=myhost-pc;DBN=demo" -y
```

接続文字列 DBN 値 (DBN=demo2) には、データソースの DBN 値よりも高い優先度があります。

```
dbping -d -c "UID=DBA;PWD=sql;DBN=demo2;DSN=demodsn"
```

接続に使用される接続パラメータは次のとおりです。

```
UID=DBA;PWD=sql;DBN=demo2;ServerName=demo17;Host=myhost-pc
```

## 接続文字列の解析

接続文字列の解析中に問題が発生した場合、問題の発生源が接続パラメータであることを示すエラーが生成されます。

空の接続パラメータ

指定されている値が空の接続パラメータは、長さ 0 の文字列として扱われます。

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[接続文字列と文字セット \[582 ページ\]](#)

### 1.1.2.3 組み込みデータベースで使用する接続パラメータ

組み込みデータベースは、通常単一アプリケーションで使用するために設計されており、アプリケーションと同じコンピュータ上で稼働し、ユーザからは見えないようになっています。

アプリケーションが組み込みデータベースを使用する場合、アプリケーションの接続時には、通常、パーソナルデータベースサーバは動作していません。データベースを起動するには、接続文字列を使用し、接続文字列の DatabaseFile (DBF) パラメータでデータベースファイルを指定します。

自動的に起動されるデータベースのクエリのパフォーマンスを向上するには、ユーザがすぐに接続しない場合でも、データベースをできるだけ早く起動します。このようにすることで、データベースに対してクエリが実行される前にキャッシュの準備が完了します。

#### ServerName (Server) 接続パラメータ

組み込みデータベースを使用するときには、ServerName (Server) 接続パラメータを使用します。これにより、同じコンピュータ上で SQL Anywhere データベースサーバを実行している別のアプリケーションが存在する場合でも、データベースが適切なデータベースサーバに接続されます。

データベースサーバが代替サーバ名を持っている場合、クライアントアプリケーションは、代替サーバ名を指定して起動したデータベースへの接続にのみ代替サーバ名を使用できます。そのデータベースサーバ上で実行中の他のデータベースに接続するのに代替サーバ名を使用することはできません。

#### DatabaseFile (DBF) 接続パラメータ

DBF 接続パラメータは、使用するデータベースファイルを指定します。データベースファイルは、ServerName 接続パラメータに一致するデータベースサーバで (まだ起動していない場合は) 自動的に起動し、データベースサーバは (まだ起動していない場合は) 自動的に起動されます。

データベースへの接続がなくなったら (通常は接続を開始したアプリケーションが切断したら)、データベースは自動的に停止します。データベースサーバが自動的に起動された場合、データベースサーバはデータベースが停止すると停止します。

次の例では、サンプルデータベースを組み込みデータベースとして起動しています。

```
"DBF=%SQLANYSAMP17%¥demo.db"
```

## StartLine [START] 接続パラメータの使用

次の接続パラメータは、組み込みデータベースとしてサンプルデータベースが開始されるのをカスタマイズする方法を示しています。StartLine 接続パラメータは、データベースサーバオプションを使用する場合に便利です。次に例を示します。

```
"START=dbeng17 -xd -c 8M;DBF=%SQLANYSAMP17%¥demo.db"
```

-xd オプションを指定すると、データベースサーバがデフォルトのデータベースサーバになりません。

ユーザが組み込みデータベースの起動や停止を行ったり、データベースサーバの別のデータベースを起動したりしないようにするには、**none** に -gd オプションを設定します。

接続パラメータには、データベースサーバの起動方法に影響を与えるものが多数存在します。StartLine (START) 接続パラメータで対応するサーバオプションを使用するのではなく、次の接続パラメータを使用することをお奨めします。

- ServerName (Server)
- DatabaseFile (DBF)
- DatabaseSwitches (DBS)
- DatabaseName (DBN)

## Elevate 接続パラメータの使用

接続文字列に ELEVATE=YES を指定すると、昇格されていないクライアントプロセスが昇格したサーバを自動的に起動できるようになります。データベースファイルへのアクセスに管理権限が要求される場合、この指定が必要となる場合があります。データベースサーバが自動的に起動されていない場合、このパラメータは無視されます。

### 例

次に示すのは、組み込みデータベースを使ったアプリケーションの完全な接続文字列の例です。

```
"UID=DBA;PWD=passwd;DBF=c:¥myapp¥mydb.db;ServerName=myapp;START=dbeng17 -c 12M -xd"
```

## 関連情報

[キャッシュウォーミング \[1375 ページ\]](#)

[接続 ID \[243 ページ\]](#)

[DatabaseFile \(DBF\) 接続パラメータ \[65 ページ\]](#)

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

[StartLine \(START\) 接続パラメータ \[107 ページ\]](#)

[-sn データベースオプション \[535 ページ\]](#)

[-gd データベースサーバオプション \[414 ページ\]](#)

[-xd データベースサーバオプション \[504 ページ\]](#)

[Elevate 接続パラメータ \[73 ページ\]](#)

## 1.1.2.4 データベースユーティリティによる接続パラメータの取得方法

すべてのデータベースユーティリティで、データベースサーバとの通信に Embedded SQL が使用されます。

### データベースユーティリティが接続パラメータの値を取得する方法

多くの管理ユーティリティは、接続パラメータの値を次の方法で取得します。

1. コマンドラインで指定された値を使用します。たとえば、次のコマンドは、ユーザ ID DBA とパスワード passwd を使用して、デフォルトサーバ上の MyTestDB データベースのバックアップを開始します。

```
dbbackup -c "UID=DBA;PWD=passwd;DBN=MyTestDB" c:¥backup
```

2. 値がない場合は、SQLCONNECT 環境変数の設定を使用します。SQL Anywhere では、この変数は自動的に設定されるわけではありません。

## 関連情報

[データベース管理ユーティリティ \[1064 ページ\]](#)

[SQLCONNECT 環境変数 \[557 ページ\]](#)

## 1.1.2.5 ODBC データソースでの接続パラメータ

ODBC データソースを使用してデータベースに接続することができます。

ODBC データソースはクライアントコンピュータ上にあります。

ODBC データソースには、一連の接続パラメータが格納されています。一連の接続パラメータを、ODBC データソースとしてシステムレジストリまたはファイルに保存できます。

SQL Anywhere では、ODBC データソースを使用できるのは ODBC アプリケーションだけではありません。ODBC データソースは、SAP Open Client と jConnect を除く、すべてのサポート済みクライアントインタフェースで使用できます。

データソースがある場合、使用するデータソースを次のように接続文字列で指定できます。

データソース名

DataSourceName (DSN) 接続パラメータを使用して、Windows レジストリ内のデータソースを参照します。

```
DSN=my-data-source
```

ファイルデータソース

FileDataSourceName (FILEDSN) 接続パラメータを使用して、ファイルに格納されているデータソースを参照します。

```
FileDSN=mysource.dsn
```

### i 注記

接続文字列と ODBC データソースのどちらも同じ接続パラメータが指定されている場合は、接続文字列の値が使用され、データソースの値は無視されます。

## 関連情報

[Windows でのファイルデータソース \[119 ページ\]](#)

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[接続パラメータの構文ルール \[39 ページ\]](#)

## 1.1.3 接続パラメータのアルファベット順リスト

接続パラメータは、接続文字列に含めます。

次の場所で入力できます。

- アプリケーションの接続文字列。
- SQLConnect 環境変数。
- ODBC データソース。
- SQL Anywhere [接続ウィンドウ](#)。

### i 注記

- 値に大文字と小文字が区別されるもの (UNIX のファイル名など) が含まれている場合でも、接続パラメータは大文字と小文字を区別しません。
- ブールパラメータは、YES、Y、ON、TRUE、T、1 のいずれかによってオンになり、NO、N、OFF、FALSE、F、0 のいずれかによってオフになります。パラメータの値は大文字と小文字を区別しません (たとえば、YES、Yes、yes はどれも同じです)。
- 各接続パラメータの使用法のところで、パラメータが使用される状況を説明します。一般的に使用法にあげる項目は以下のとおりです。

#### 組み込みデータベース

パーソナルデータベースサーバ (dbeng17) は、通常組み込みアプリケーションとして使用します。アプリケーションがデータベースから切断されると、データベースはアンロードされ、サーバが停止します。

#### ローカルデータベースサーバ

ローカルデータベースサーバとは、自分のパソコン上で動作しているサーバのことです。パーソナルサーバ (dbeng17) とネットワークサーバ (dbsrv17) のいずれも使用できます。

ネットワークデータベースサーバ

ネットワークサーバ (dbsrv17) は、自分のパソコン上でも、ネットワークにある別のコンピュータ上でも実行できます。他のユーザもデータベースに接続できるようにするには、ネットワークサーバを使用します。

- dbping ユーティリティを使用して、接続文字列をテストできます。-c オプションを使用して、接続パラメータを指定します。たとえば、demo17 という名前のデータベースサーバがサンプルデータベース (コマンド `dbsrv17 -n demo17 "%SQLANY%SAMP17%¥demo.db"` で起動可能) を実行しているとします。次のコマンドは、データベースへの接続が成功したというメッセージを返します。

```
dbping -d -c "Server=demo17;DBN=demo;UID=DBA;PWD=sql"
```

ローカルコンピュータ上で other-server という名前のデータベースサーバが実行中ではない場合、次のコマンドは、`Ping server failed - Database server not found` というメッセージを返します。

```
dbping -c "Server=other-server"
```

このセクションの内容:

[AppInfo \(APP\) 接続パラメータ \[48 ページ\]](#)

データベースサーバからの特定のクライアント接続の開始を、管理者が容易に識別できるようにします。

[AutoStart \(ASTART\) 接続パラメータ \[50 ページ\]](#)

接続可能なデータベースサーバがない場合にパーソナルデータベースサーバを起動するかどうかを制御します。

[AutoStop \(ASTOP\) 接続パラメータ \[51 ページ\]](#)

オープンな非 HTTP 接続がなくなったときにデータベースを停止するかどうかを制御します。

[CharSet \(CS\) 接続パラメータ \[53 ページ\]](#)

この接続で使用する文字セットエンコーディングを指定します。

[ClientFileValidator \(CFV\) 接続パラメータ \[54 ページ\]](#)

ローカルファイルの読み取りまたは書き込み要求が信頼できるソースから生成されていることをデータベースサーバが検証する方法を制御します。

[CommBufferSize \(CBSIZE\) 接続パラメータ \[55 ページ\]](#)

通信パケットの最大サイズを設定します。

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

クライアント側のネットワークプロトコルオプションを指定します。

[Compress \(COMP\) 接続パラメータ \[59 ページ\]](#)

接続の圧縮をオンまたはオフに設定します。

[CompressionThreshold \(COMPTR\) 接続パラメータ \[60 ページ\]](#)

パケットの圧縮サイズの制限値を増減します。

[ConnectionName \(CON\) 接続パラメータ \[62 ページ\]](#)

接続に名前を付け、マルチ接続アプリケーションで簡単に切り替えができるようにします。

[ConnectionPool \(CPOOL\) 接続パラメータ \[63 ページ\]](#)

クライアントの接続プールの動作を制御します。

[DatabaseFile \(DBF\) 接続パラメータ \[65 ページ\]](#)

実行していないデータベースを起動したときにロードして接続するデータベースファイルを指定します。

#### DatabaseKey (DBKEY) 接続パラメータ [67 ページ]

暗号化されたデータベースを接続要求で起動します。

#### DatabaseName (DBN) 接続パラメータ [68 ページ]

実行中のデータベースを識別したり、起動中のデータベースに名前を割り当てたりします。

#### DatabaseSwitches (DBS) 接続パラメータ [70 ページ]

データベース起動時に、データベース固有のオプションを提供します。

#### DataSourceName (DSN) 接続パラメータ [71 ページ]

ODBC ドライバマネージャまたは Embedded SQL ライブラリに対して、レジストリ内またはシステム情報ファイル (デフォルト名は `.odbc.ini`) での ODBC データソース情報の検索場所を指示します。

#### DisableMultiRowFetch (DMRF) 接続パラメータ [72 ページ]

ネットワーク上での複数ローフェッチをオフにします。

#### Elevate 接続パラメータ [73 ページ]

Elevates は、Windows 7 以降でデータベースサーバの実行プログラムを自動的に起動します。

#### EncodedPassword (ENP) 接続パラメータ [74 ページ]

パスワード文字列がエンコード形式であることを示します。

#### Encryption (ENC) 接続パラメータ [77 ページ]

クライアントアプリケーションとデータベースサーバとの間で送信されるパケットのエンコードを指定します。これには、エンコードなし、単純難読化、TLS (Transport Layer Security) 暗号化があります。

#### EngineName (ENG) 接続パラメータ (旧式) [80 ページ]

この接続パラメータは使用されなくなりました。代わりに、ServerName (Server) 接続パラメータを使用します。

#### FileDataSourceName (FILEDSN) 接続パラメータ [80 ページ]

接続するデータベースに関する情報を ODBC ファイルデータソースが保有していることをクライアントライブラリに知らせます。

#### ForceStart (FORCE) 接続パラメータ [81 ページ]

実行しているサーバがあるかどうかをチェックすることなくデータベースを起動します。

#### Host 接続パラメータ [82 ページ]

ホスト名または IP アドレスとオプションのポート番号を受け入れ、データベースサーバがある場所をクライアントに知らせます。

#### IdleTimeout (IDLE) 接続パラメータ [85 ページ]

接続のアイドルタイムアウト時間を指定します。

#### Integrated (INT) 接続パラメータ [86 ページ]

統合化ログインを試行できるかどうかを指定します。

#### Kerberos (KRB) 接続パラメータ [87 ページ]

データベースサーバへの接続時に Kerberos 認証を使用できるかどうかを指定します。

#### Language (LANG) 接続パラメータ [89 ページ]

接続の言語を指定します。

#### LazyClose (LCLOSE) 接続パラメータ [90 ページ]

カーソル要求を次の要求が発生するまでキューイングするか、またはただちに実行するかを制御します。

#### LivenessTimeout (LTO) 接続パラメータ [91 ページ]

不完全な接続の終了を制御します。

#### [LogFile \(LOG\) 接続パラメータ \[92 ページ\]](#)

ファイルにクライアントエラーメッセージとデバッグメッセージを送信します。

#### [NewPassword \(NEWPWD\) 接続パラメータ \[94 ページ\]](#)

パスワードの有効期限が切れている場合でも、DBA の介入を必要とせずユーザが各自でパスワードを変更できます。

#### [MatView \(MATVIEW\) 接続パラメータ \[95 ページ\]](#)

ODBC 関数の SQLTables が呼び出されたときに、マテリアライズドビューに返される文字列を指定します。

#### [NodeType \(NODE\) 接続パラメータ \[96 ページ\]](#)

ミラーリング構成に参加している、接続先のサーバのタイプを選択します。

#### [Password \(PWD\) 接続パラメータ \[99 ページ\]](#)

接続時のパスワードを指定します。

#### [PrefetchBuffer \(PBUF\) 接続パラメータ \[100 ページ\]](#)

ローをバッファするためのメモリの最大容量を、バイト単位で設定します。

#### [PrefetchOnOpen 接続パラメータ \[102 ページ\]](#)

パラメータが有効であるときに、カーソルを開く要求を含むプリフェッチ要求が送信されます。

#### [PrefetchRows \(PROWS\) 接続パラメータ \[102 ページ\]](#)

データベースのクエリ時にプリフェッチされる、最初に推奨されるローの数を指定します。

#### [RetryConnectionTimeout \(RetryConnTO\) 接続パラメータ \[104 ページ\]](#)

クライアントライブラリ (DBLIB、ODBC、ADO など) に、サーバが見つからない場合は指定した時間が経過するまで接続を試行し続けるよう指示します。

#### [ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

接続する実行中のデータベースサーバ名を指定します。

#### [StartLine \(START\) 接続パラメータ \[107 ページ\]](#)

アプリケーションからローカルデータベースサーバを起動します。

#### [Unconditional \(UNC\) 接続パラメータ \[109 ページ\]](#)

データベースサーバへの接続があるときでも、db\_stop\_engine 関数を使用してデータベースサーバを停止します。または、db\_stop\_database 関数を使用してデータベースを停止します。

#### [Userid \(UID\) 接続パラメータ \[109 ページ\]](#)

データベースへのログインに使用するユーザ ID を指定します。

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[接続パラメータの構文ルール \[39 ページ\]](#)

[ODBC データソース \[111 ページ\]](#)

[Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

[トラブルシューティング: 接続の確立方法 \[249 ページ\]](#)

### 1.1.3.1 AppInfo (APP) 接続パラメータ

データベースサーバからの特定のクライアント接続の開始を、管理者が容易に識別できるようにします。

#### 構文

```
{ AppInfo | APP }=keyword=value
```

#### 使用法

特に制限なし

#### 指定可能な値

クライアントは、固有の文字列を指定できます。指定した文字列は、生成された文字列に追加されます。AppInfo プロパティ文字列は、セミコロンで区切られた `key=value` ペアのシーケンスです。有効なキーは次のとおりです。

##### API

DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerIDBD、DBEXPRESS のいずれか。

##### APPINFO

接続文字列に AppInfo を指定したときに入力される文字列。

##### EXE

クライアント実行プログラムの名前 (Windows、Linux、Solaris)。

##### HOST

クライアントコンピュータのホスト名。

##### IP

クライアントコンピュータの IP アドレス。

##### OS

オペレーティングシステムの名前とバージョン番号。

##### OSUSER

クライアントプロセスに関連付けられたオペレーティングシステムユーザ名。クライアントプロセスで別のユーザを同一化している場合 (UNIX の場合は、セット ID ビットが設定されている場合)、同一化されたユーザ名が返されます。バージョン 10.0.1 以前のクライアントと HTTP や TDS クライアントでは、空の文字列を返します。

##### PID

クライアントのプロセス ID (Windows と UNIX のみ)。

##### THREAD

クライアントのスレッド ID (Windows と UNIX のみ)。

##### TIMEZONEADJUSTMENT

接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある分数。

## VERSION

主要なバージョン番号、それ以外のバージョン番号、ビルド番号を含む、使用しているクライアントライブラリのバージョン (17.0.4.1691 など)。

## デフォルト

空の文字列

## 備考

この接続パラメータは、Embedded SQL、ODBC、OLE DB、または ADO.NET クライアント、および JDBC ドライバを使用するアプリケーションから、データベースサーバに送信されます。SAP Open Client または jConnect アプリケーションから使用することはできません。

このパラメータは、クライアントコンピュータの IP アドレスや実行されているオペレーティングシステムなどの、クライアントプロセスについての情報を保持するように生成された文字列で構成されています。文字列は、接続するデータベースサーバに関連付けられており、次の文を使用して検索できます。

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

クライアント接続パラメータにデバッグログファイルを指定すると、そのファイルに AppInfo 文字列が追加されます。

### 例

Interactive SQL からサンプルデータベースに接続します (デフォルトで JDBC ドライバを使用します)。

```
dbisql -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP17%¥demo.db"
```

Interactive SQL で、アプリケーション情報を表示します。

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

結果は、次のとおりです (1 つの文字列で表示されます)。

```
IP=ip-address;
HOST=computer-name;
OSUSER=user-name;
OS='operating-system-name';
EXE='C:¥Program Files¥SQL Anywhere 17¥Bin64¥dbisql.exe';
PID=0x1170;
THREAD=0x24a4;
VERSION=17.0.4.1691;
API=iAnywhereJDBC;
TIMEZONEADJUSTMENT=-240
```

AppInfo プロパティにユーザ固有の情報を追加して、Interactive SQL からサンプルデータベースに接続します。

```
dbisql -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP17%¥demo.db;APP=Interactive SQL connection"
```

アプリケーション情報を表示します。

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

結果は、次のとおりです (1つの文字列で表示されます)。指定した APPINFO 値が最後に表示されていることに注意してください。

```
IP=ip-address;  
HOST=computer-name;  
OSUSER=user-name;  
OS='operating-system-name';  
EXE='C:\Program Files\SQL Anywhere 17\Bin64\dbisql.exe';  
PID=0x1170;  
THREAD=0x24a4;  
VERSION=17.0.4.1691;  
API=iAnywhereJDBC;  
TIMEZONEADJUSTMENT=-240  
APPINFO='Interactive SQL connection'
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[request\\_timeout オプション \[772 ページ\]](#)

### 1.1.3.2 AutoStart (ASTART) 接続パラメータ

接続可能なデータベースサーバがない場合にパーソナルデータベースサーバを起動するかどうかを制御します。

#### 構文

```
{ AutoStart | ASTART }={ YES | NO }
```

## 使用方法

特に制限なし

## デフォルト

YES

## 備考

データベースファイル、データベース名、START 接続パラメータのいずれかが指定されていて、接続時にデータベースサーバが検出できない場合、デフォルトでは、パーソナルデータベースサーバは同一のコンピュータで起動されます。接続文字列の AutoStart (ASTART) 接続パラメータを NO に設定することによって、このような動作を無効にできます。Host 接続パラメータが指定されているか、CommLinks (LINKS) パラメータに TCPIP が含まれている場合、データベースサーバは自動的に起動しません。

自動的に起動されるデータベースのクエリのパフォーマンスを向上するには、ユーザがすぐに接続しない場合でも、データベースをできるだけ早く起動します。このように起動することで、データベースに対してクエリが実行される前にキャッシュの準備が完了します。

### i 注記

AutoStart 接続パラメータは、データベースサーバを自動的に起動するかどうかを制御します。データベースを自動的に起動するかどうかを制御するものではありません。DatabaseName 接続パラメータを使用すると、すでに起動しているデータベースに接続します。DatabaseFile 接続パラメータを使用すると、すでに起動しているデータベースに接続するか、あるいは必要に応じてデータベースを自動的に起動して接続します。

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[キャッシュウォーミング \[1375 ページ\]](#)

[DatabaseName \(DBN\) 接続パラメータ \[68 ページ\]](#)

[DatabaseFile \(DBF\) 接続パラメータ \[65 ページ\]](#)

[トラブルシューティング: データベースサーバの検出方法 \[251 ページ\]](#)

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

[Elevate 接続パラメータ \[73 ページ\]](#)

[AutoStop \(ASTOP\) 接続パラメータ \[51 ページ\]](#)

### 1.1.3.3 AutoStop (ASTOP) 接続パラメータ

オープンな非 HTTP 接続がなくなったときにデータベースを停止するかどうかを制御します。

#### 構文

```
{ AutoStop | ASTOP }={ YES | NO }
```

## 使用方法

特に制限なし

## デフォルト

YES

## 備考

デフォルトでは、接続文字列で起動したすべてのデータベースサーバは、非 HTTP 接続がなくなると停止します。また、接続文字列からロードしたすべてのデータベースも、非 HTTP 接続がなくなるとアンロードされます。この動作は、AutoStop=YES と同じです。

AutoStop=NO を指定すると、その接続で起動したすべてのデータベースは、非 HTTP 接続がなくなっても実行を続けます。したがって、データベースサーバは操作可能な状態を維持します。

データベースへの唯一の接続が HTTP 接続で、自動的に停止するようにデータベースが設定されている場合、HTTP 接続が切断したときにデータベースは自動的に停止しません。また、自動的に停止するように設定されているデータベースに HTTP 接続と Command Sequence 接続または TDS 接続がある場合は、最後の Command Sequence 接続または TDS 接続が切断したときにデータベースが停止します。このときに HTTP 接続がまだあった場合は切断されます。

AutoStop (ASTOP) 接続パラメータは、現在実行していないデータベースに接続するときのみ使用されます。データベースがすでに起動されている場合は、無視されます。

.NET アプリケーションで AutoStop 接続パラメータを使用する場合は注意が必要です。接続を閉じると関連するアプリケーションも閉じられますが、接続プールが有効な場合、アクティブな接続は開いたままになります。その結果、サーバの停止を予期していても、そのように動作しません。

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[データベースの起動と停止 \[314 ページ\]](#)

[-ga データベースサーバオプション \[412 ページ\]](#)

[AutoStart \(ASTART\) 接続パラメータ \[50 ページ\]](#)

## 1.1.3.4 CharSet (CS) 接続パラメータ

この接続で使用する文字セットエンコーディングを指定します。

### 構文

```
{ CharSet | CS }={ NONE | character-set }
```

### 使用法

特に制限なし

### 指定可能な値

#### NONE

CharSet=NONE を指定すると、接続でデータベース CHAR 文字セットエンコーディングの使用することを要求します。これはデフォルトの動作です。

#### character-set

使用する文字セットエンコーディングのラベルを指定します。

### デフォルト

ANSI モードの DBLIB (ESQL、Perl、PHP、Python、Ruby)、ODBC: ローカル (クライアント) オペレーティングシステム文字セットエンコーディング。

UNICODE モードの ADO.NET、JDBC、および ODBC と OLE DB: データベース CHAR 文字セットエンコーディング。

JavaScript: データベース NCHAR 文字セットエンコーディング。

### 備考

CharSet に値を指定すると、指定された文字セットエンコーディングが現在の接続に使用されます。データベース CHAR 文字セットエンコーディングから指定された接続文字セットエンコーディングへの変換が試みられます。NCHAR 文字セットエンコーディングは、この接続パラメータの影響を受けません。

ADO.NET、JDBC、OLE DB は、Unicode アプリケーションプログラミングインタフェースのため、CharSet 設定は無視されません。

推奨ラベルの代わりとして文字セットエンコーディングラベルを指定することもできます。この場合、推奨ラベルは CharSet 接続プロパティにより報告されます。たとえば、次の Interactive SQL コマンドは接続文字列で cp1251 文字セットエンコーディングを指定します。

```
dbisql -c "UID=DBA;PWD=passwd;CharSet=cp1251" "SELECT
connection_property('CharSet')"
```

返される結果は次のとおりです。

```
connection_property('charset')
-----
windows-1251
```

一部の代替文字列セットエンコーディングラベルは一意でないため、使用しないでください (ibm-943\_P15A-2003 など)。

データをアンロードする場合は、CharSet 接続パラメータを使用して文字セットを指定します。

Unicode クライアント API を使用している場合は、文字セット変換によるデータの損失を避けるために、CharSet 接続パラメータを設定することはおすすりできません。Unicode クライアント API には、ADO.NET、OLE DB、SQL Anywhere JDBC ドライバが含まれます。ODBC も、ワイド (Unicode) 関数が使用されている場合は、Unicode クライアント API です。

#### 例

次の接続文字列フラグメントは、Windows 1252 文字セットエンコーディングを接続に使用することを指定します。

```
CharSet=windows-1252
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[ロケール情報 \[590 ページ\]](#)

[ロケール文字セット \[589 ページ\]](#)

[SACHARSET 環境変数 \[549 ページ\]](#)

[推奨文字セットと照合 \[607 ページ\]](#)

### 1.1.3.5 ClientFileValidator (CFV) 接続パラメータ

ローカルファイルの読み取りまたは書き込み要求が信頼できるソースから生成されていることをデータベースサーバが検証する方法を制御します。

#### 構文

```
{ ClientFileValidator | CFV } = { TrustedServer | Callback | Never }
```

## 使用法

特に制限なし

## 指定可能な値

### Never

ファイル転送を許可しません。

### Callback

DB\_CALLBACK\_VALIDATE\_FILE\_TRANSFER 関数で定義されています。コールバックが定義されており、TRUE を返す場合のみファイルを転送します。

### TrustedServer

要求がクライアントから直接生成されており、間接的な要求の結果でないとデータベースサーバが判断した場合、またはコールバックが TRUE を返す場合にファイルを転送します。

## デフォルト

TrustedServer

## 備考

この接続パラメータは、ファイルを制御する他のすべてのオプションより優先されます。

### 例

次の例では、クライアントからデータベースサーバへのすべてのファイル転送が禁止されます。

```
CFV=Never
```

## 1.1.3.6 CommBufferSize (CBSIZE) 接続パラメータ

通信パケットの最大サイズを設定します。

### 構文

```
{ CommBufferSize | CBSIZE }=size[ k ]
```

## 使用法

特に制限なし

## 指定可能な値

### size

この整数は、通信パケットの最大サイズを指定します。デフォルト値はバイト数で指定します。キロバイト単位で指定するには K を使用します。CommBufferSize の最小値は 1000 バイト、最大値は 65535 バイトです。

## デフォルト

CommBufferSize 値を設定しないと、サーバ側の設定 (デフォルトは 7300 バイト) によって CommBufferSize が制御されます。

ミラーサーバとの接続のための CommBufferSize のデフォルト値は 64240 バイトです。

## 備考

ネットワーク上のパケットの最大サイズは、プロトコルスタックによって設定されます。CommBufferSize をネットワークで許可されているサイズより大きく設定すると、通信パケットがネットワークソフトウェアによって分割されます。デフォルトのサイズは、標準 Ethernet TCP/IP の最大パケットサイズ (1460 バイト) の倍数です。

パケットサイズを大きくすると、複数のローのフェッチと長いローのフェッチのパフォーマンスが向上しますが、クライアントとサーバのメモリ使用量が増加します。

クライアント側で CommBufferSize の指定がないと、接続ではデータベースサーバのバッファサイズが使用されます。クライアント側で CommBufferSize の指定がある場合、接続では CommBufferSize 値が使用されます。

-p データベースサーバオプションを使用して CommBufferSize を設定すると、CommBufferSize を指定していないすべてのクライアントで -p データベースサーバオプションで指定されたサイズが使用されます。

### 例

次の文では、バッファサイズを 1460 バイトに設定します。

```
...  
CommBufferSize=1460  
...
```

またこのパラメータは、[SQL Anywhere の ODBC 設定ウィンドウの詳細タブ](#)で、`CommBufferSize` テキストボックスに値を入力して設定することもできます。

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[-p データベースサーバオプション \[457 ページ\]](#)

### 1.1.3.7 CommLinks (LINKS) 接続パラメータ

クライアント側のネットワークプロトコルオプションを指定します。

#### 構文

```
{ CommLinks | LINKS }={ [ SharedMemory | ShMem ] | ALL | [ TCPIP | TCP ] }  
[, ... ] string
```

## 使用法

CommLinks (LINKS) 接続パラメータは、ネットワークデータベースサーバ (dbsrv17) への接続に使用します。オプションで、パーソナルデータベースサーバ (dbeng17) への接続にも使用します。

## 指定可能な値

CommLinks (LINKS) 接続パラメータの値は、大文字と小文字を区別しません。次の値が含まれます。

### SharedMemory (ShMem)

同一コンピュータ通信の共有メモリプロトコルを使用して接続します。これはデフォルト設定です。共有メモリが CommLinks(LINKS) 接続パラメータによって指定されたプロトコルのリストに含まれている場合は、リストでの順序に関係なく、クライアントは最初に共有メモリを使用しようとします。

### ALL

最初に共有メモリプロトコルを使用して接続しようとし、次に TCP/IP を使用します。

### TCPIP (TCP)

TCP/IP 通信プロトコルを使用して接続します。TCP/IP は、すべてのオペレーティングシステムでサポートされています。

TCPIP 用の追加のネットワークプロトコルオプションを指定できます。

## デフォルト

SharedMemory

## 備考

### i 注記

CommLinks (LINKS) 接続パラメータは、HOST または ServerPort (PORT) 以外の TCP/IP プロトコルオプションを指定する必要がある場合にのみ使用してください。これ以外の場合は、Host 接続パラメータを使用します。

接続文字列に CommLinks と Host の両方は指定できません。

TCP/IP 接続に CommLinks (LINKS) 接続パラメータを使用する場合は、ServerName (Server) 接続パラメータを使用してサーバ名を指定してください。

Host 接続パラメータまたは CommLinks (LINKS) 接続パラメータを指定しないと、クライアントは現在のコンピュータにあるデータベースサーバしか検索せず、共有メモリ接続のみを使用します。これはデフォルトの動作であり、CommLinks=ShMem を指定するのと同じです。共有メモリプロトコルは、パーソナルデータベースサーバに接続するアプリケーションでの標準的な使い方、同じコンピュータで実行されているクライアントとデータベースサーバ間で最速の通信リンクです。

TCPIP と SharedMemory の両方、または CommLinks=ALL を指定すると、最初に共有メモリが使用され、共有メモリでデータベースサーバが検出できない場合に TCP/IP が使用されます。たとえば、次の接続文字列では、共有メモリが先に使用されます。

```
"UID=DBA;PWD=sql;Server=demo;DBN=demo;LINKS=TCPIP,SHMEM"
```

Host 接続パラメータが指定されているか、CommLinks (LINKS) パラメータに TCP/IP が含まれている場合、パーソナルデータベースサーバは自動的に起動しません。

TCP/IP 通信プロトコルをデータベースサーバ上で有効にするには -x オプションを使用します。

### 例

次の接続文字列フラグメントは、現在のサブネット上で実行中の demo という名前のデータベースサーバに接続しようとしています。

```
CommLinks=tcPIP;Server=demo;UID=DBA;PWD=sql;
```

次の接続文字列フラグメントは、demo という名前のデータベースサーバに接続しようとしています。最初は、共有メモリを介して接続しようとしています。共有メモリを使用してもデータベースサーバが見つからない場合、TCP/IP を使用して接続しようとしています。

```
CommLinks=tcPIP,shmem;Server=demo;UID=DBA;PWD=sql
```

次の接続文字列は、kangaroo という名前のホスト上で実行している demo という名前のサーバに接続しようとしています。

```
CommLinks=tcPIP (HOST=kangaroo) ;Server=demo;UID=DBA;PWD=sql
```

その他の方法として、Host 接続パラメータを使用して、kangaroo という名前のホスト上で実行している demo という名前のサーバに接続することもできます。

```
Host=kangaroo;Server=demo;UID=DBA;PWD=sql
```

## 関連情報

[ネットワークプロトコルオプション \[177 ページ\]](#)

[通信プロトコル \[164 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

[セキュリティに関する一般的なヒント \[1598 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

[-x データベースサーバオプション \[500 ページ\]](#)

[トラブルシューティング: 高速接続のための sasrv.ini へのデータベースサーバアドレス情報のキャッシュ方法 \[261 ページ\]](#)

### 1.1.3.8 Compress (COMP) 接続パラメータ

接続の圧縮をオンまたはオフに設定します。

#### 構文

```
{ Compress | COMP }={ YES | NO }
```

## 使用方法

TDS 接続以外。他には特に制限なし。TDS 接続 (jConnect を含む) では、SQL Anywhere 通信圧縮はサポートされません。

## デフォルト

NO

## 備考

クライアントとデータベースサーバの間でやり取りされるパケットは、Compress (COMP) 接続パラメータを使用して圧縮できます。状況によっては、接続を圧縮することでパフォーマンスが向上する場合があります。大幅に圧縮可能なデータの大規模なデータ転送では、圧縮率が高くなります。

Compress 接続パラメータに値が設定されていない場合、圧縮ステータスはデータベースサーバ側の設定によって制御されます。デフォルトでは、圧縮を行いません。クライアントとデータベースサーバで設定が異なる場合は、クライアントの設定が適用されます。

特定のアプリケーションを使用してネットワークのパフォーマンス分析を行ってから、運用環境で通信の圧縮を使用することをお奨めします。

データベースサーバのすべてのリモート接続で圧縮を有効にするには、`-pc` サーバオプションを使用します。

`-pc` オプションまたは `COMPRESS=YES` パラメータを指定しても、使用する通信リンクにかかわらず、同一コンピュータ接続では圧縮は有効になりません。

#### 例

次の接続文字列フラグメントは、パケット圧縮をオンに設定します。

```
Compress=YES
```

次の接続文字列フラグメントは、パケット圧縮をオフに設定します。

```
Compress=NO
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[通信の圧縮設定 \[175 ページ\]](#)

[-pc データベースサーバオプション \[457 ページ\]](#)

### 1.1.3.9 CompressionThreshold (COMPTH) 接続パラメータ

パケットの圧縮サイズの制限値を増減します。

#### 構文

```
{ CompressionThreshold | COMPTH }=size[ k ]
```

## 使用法

TDS 以外。他には特に制限なし。圧縮接続にのみ適用。

## 指定可能な値

**size**

この整数は、パケットの圧縮が適用される最小パケットサイズを指定します。デフォルト値はバイト単位ですが、kを使用してキロバイトの単位を指定できます。クライアントとデータベースサーバで圧縮スレッシュホールドの設定が異なる場合は、クライアントの設定が適用されます。サポートされている最小値は1バイト、最大値は32767バイトです。80バイト未満の値はお奨めしません。

## デフォルト

120

CompressionThreshold 値が設定されていない場合、圧縮スレッシュホールド値は、サーバ側の設定によって制御されます。デフォルトは120バイトです。

## 備考

圧縮した場合に転送速度が速くなるパケットのみを圧縮するように圧縮スレッシュホールドを変更して、圧縮接続のパフォーマンスを向上できます。

圧縮が有効な場合、パケットは、各々のサイズに応じて圧縮するかどうかを決定します。たとえば、データベースサーバでは、圧縮のスレッシュホールドよりも小さいパケットは、通信の圧縮が有効な場合でも圧縮されません。同様に、小さなパケット(100バイト未満)は、通常はまったく圧縮されません。パケットの圧縮にはCPU時間が必要なため、小さなパケットを圧縮しようとすると、実際にパフォーマンスが低下することがあります。

圧縮スレッシュホールドの値を小さくすると、非常に低速なネットワークではパフォーマンスが向上し、値を大きくするとCPU使用率の減少によってパフォーマンスが向上する場合があります。ただし、圧縮のスレッシュホールド値を小さくするとクライアントとサーバの両方でCPU使用率が増加するので、パフォーマンス分析を行って、圧縮のスレッシュホールドを変更することでパフォーマンスが向上するかどうかを判断してください。

### 例

圧縮スレッシュホールド値を100バイトに設定して接続します。

```
CompressionThreshold=100
```

## 関連情報

[通信の圧縮設定 \[175 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

[-pt データベースサーバオプション \[461 ページ\]](#)

## 1.1.3.10 ConnectionName (CON) 接続パラメータ

接続に名前を付け、マルチ接続アプリケーションで簡単に切り替えができるようにします。

### 構文

```
{ ConnectionName | CON }=connection-name
```

### 使用方法

特に制限なし

### 指定可能な値

#### connection-name

この文字列は、確立中の特定の接続に名前を指定します。

### デフォルト

接続名なし

### 備考

この接続パラメータはオプションです。複数の接続を確立しても切り替えを行わないときは、この値を指定する必要はありません。

接続名はデータソース名とは異なります。

接続をプールするには、接続名は異なってもかまいませんが、他のすべての接続パラメータは同じである必要があります。

データベースサーバによって作成されるテンポラリ接続には次の名前が使用されます。

- INT:ApplyRecovery
- INT:BackupDB
- INT:Checkpoint
- INT:Cleaner
- INT:CloseDB
- INT>CreateDB

- INT:CreateMirror
- INT:DelayedCommit
- INT:DiagRcvr
- INT:DropDB
- INT:EncryptDB
- INT:Exchange
- INT:FlushMirrorLog
- INT:FlushStats
- INT:HTTPReq
- INT:PromoteMirror
- INT:PurgeSnapshot
- INT:ReconnectMirror
- INT:RecoverMirror
- INT:RedoCheckpoint
- INT:RefreshIndex
- INT:ReloadTrigger
- INT:RenameMirror
- INT:RestoreDB
- INT:StartDB
- INT:VSS

#### 例

first-con という名前の接続を指定して、データベースに接続します。

```
CON=first-con
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

### 1.1.3.11 ConnectionPool (CPOOL) 接続パラメータ

クライアントの接続プールの動作を制御します。

#### 構文

```
ConnectionPool={ NO | YES [ ( [ Timeout=timeout-sec; ] [ MaxCached=max-cached-conn ] ) ] }
```

## 使用法

非スレッド化 UNIX クライアントを除くすべてのプラットフォーム。

## 指定可能な値

### timeout-sec

接続プールのアイドルタイムアウト時間 (秒単位)。デフォルト値は 60 秒です。キャッシュされた接続は、`timeout-sec` で指定された時間内に再利用しないと、再利用できなくなります。

### max-cached-conn

各アプリケーションのキャッシュされた接続の最大数。デフォルトの接続数は 5 です。接続が切断され、`max-cached-conn` で指定された最大接続数に達していない場合は、接続がキャッシュされます。接続が再初期化されます。接続はアプリケーションによって切断されますが、キャッシュされた接続はデータベースサーバに接続したままになります。

## デフォルト

YES

## 備考

接続プールによって、データベースサーバに対して複数の短時間接続を行うアプリケーションのパフォーマンスが向上する可能性があります。接続が切断されると自動的にキャッシュされ、アプリケーションが再接続した時点で再利用できます。接続をプールするには、接続名は異なってもかまいませんが、他のすべての接続パラメータは同じである必要があります。

.NET アプリケーションでの接続プールには、.NET の [Pooling](#) 接続パラメータを使用する必要があります。ConnectionPool (CPOOL) 接続パラメータは無視されます。.NET の接続プールの詳細については、.NET 接続プールのトピックを参照してください。

### 例

次の接続文字列フラグメントは、接続プールをオフに設定します。

```
ConnectionPool=NO;
```

次の接続文字列フラグメントは、キャッシュされた接続の最大数に 10 を使用して、接続プールをオンに設定します。

```
CPOOL=YES (MaxCached=10);
```

## 関連情報

[接続プールによるアプリケーションパフォーマンスの向上 \[244 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

### 1.1.3.12 DatabaseFile (DBF) 接続パラメータ

実行していないデータベースを起動したときにロードして接続するデータベースファイルを指定します。

#### 構文

```
{ DatabaseFile | DBF }=filename
```

## 使用法

ローカルデータベースサーバ

## 指定可能な値

### filename

この文字列は、起動するデータベースのパスとファイル名を指定します。

ファイル名に拡張子が含まれていない場合、データベースサーバは .db 拡張子の付いたファイルを検索します。

ファイルのパスは、データベースサーバの作業ディレクトリの相対パスです。データベースサーバをコマンドプロンプトから起動すると、コマンド実行時の (現在の) ディレクトリが作業ディレクトリになります。データベースサーバをアイコンかショートカットから起動すると、アイコンかショートカットを指定したディレクトリが作業ディレクトリになります。完全なパスとファイル名を指定することをお奨めします。

UNC ファイル名がサポートされています。

## デフォルト

デフォルト設定なし

## 備考

データベースが実行中かどうか分からない場合には、DatabaseFile (DBF) 接続パラメータを使用することで、データベースを起動して接続することができます。

DatabaseFile (DBF) と DatabaseName (DBN) の両方の接続パラメータを指定した場合、指定したデータベース名を持つ実行中のデータベースに接続しようとします (DatabaseFile 接続パラメータは無視されます)。この試行が失敗した場合は、DatabaseFile 接続パラメータで指定されたパスを使用して自動的にデータベースを起動し、DatabaseName 接続パラメータで指定された名前を割り当てようとします。

DatabaseFile (DBF) 接続パラメータを指定して DatabaseName (DBN) 接続パラメータを指定しなかった場合、指定されたファイル (パスおよび拡張子を除く) と同じ名前を持つ実行中のデータベースに接続しようとします。この試行が失敗した場合は、DatabaseFile 接続パラメータで指定されたパスを使用して、自動的にデータベースを起動しようとします。

ネットワークサーバ上でのデータベースの自動的な起動は、-gd データベースサーバオプションによって明示的に許可されない限り、行えません。

実行中のデータベースサーバが見つからない場合、デフォルトでは、パーソナルデータベースサーバが自動的に起動されます。

配備されたアプリケーションでは、実行されていないデータベースファイルを自動的に起動する場合、ServerName (Server) パラメータを使用してデータベースサーバ名を指定することをお奨めします。そうしないと、アプリケーションが別のデータベースサーバに接続する可能性があります。たとえば、データベースサーバが、組み込みアプリケーションの一部としてすでに実行中の、バージョンの異なる SQL Anywhere データベースサーバに接続してしまうこともありえます。

### 警告

データベースファイルは、データベースサーバと同じコンピュータ上に置いてください。ネットワークドライブにあるデータベースファイルを起動すると、ファイルが破損することがあります。

### 例

次の例の DatabaseFile (DBF) 接続パラメータは、サンプルデータベース *demo.db* をロードして接続します。

```
"DBF=%SQLANYSAMPl7%¥demo.db"
```

次の 2 つの例では、*cities.db* というデータベースファイルをすでに起動していて、次のように名前を Kitchener に変更したものと仮定しています。

```
dbeng17 cities.db -n Kitchener
```

データベースを正常に起動して接続し、Kitchener という名前をつけるには、DBN および DBF 接続パラメータの値を次のように指定します。

```
DBN=Kitchener;DBF=cities.db
```

DBF=cities.db と指定すると、Kitchener という名前の実行中のデータベースへの接続が失敗します。

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[組み込みデータベースで使用する接続パラメータ \[41 ページ\]](#)

[-gd データベースサーバオプション \[414 ページ\]](#)

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

[DatabaseName \(DBN\) 接続パラメータ \[68 ページ\]](#)

### 1.1.3.13 DatabaseKey (DBKEY) 接続パラメータ

暗号化されたデータベースを接続要求で起動します。

#### 構文

```
{ DatabaseKey | DBKEY }=key
```

#### 使用法

特に制限なし

#### 指定可能な値

##### key

暗号化キーは、大文字、小文字、数字、文字、特殊記号を含む文字列です。データベースキーには、前後のスペースやセミコロンを含めることはできません。

#### デフォルト

なし

#### 備考

接続要求で暗号化データベースを起動するときには、このパラメータを指定します。

クライアントアプリケーションとデータベースサーバ間の通信パケットの安全性を保つため、-ec サーバオプションとトランスポートレイヤセキュリティを使用します。

#### 例

次のフラグメントは、DatabaseKey (DBKEY) 接続パラメータの使用方法を示します。

```
"UID=DBA;PWD=sql;Server=myeng;DBKEY=V3moj3952B;DBF=%SQLANY17%demo.db"
```

## 関連情報

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

[トランスポートレイヤセキュリティを使用するようにクライアントアプリケーションを設定する \[1655 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

[-ek データベースオプション \[527 ページ\]](#)

[-ep データベースサーバオプション \[406 ページ\]](#)

[-es データベースサーバオプション \[407 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

### 1.1.3.14 DatabaseName (DBN) 接続パラメータ

実行中のデータベースを識別したり、起動中のデータベースに名前を割り当てたりします。

#### 構文

```
{ DatabaseName | DBN }=database-name
```

## 使用法

特に制限なし

## 指定可能な値

### database-name

この文字列は、データベースサーバですでに実行しているデータベースの名前や、起動中のデータベースの名前を指定します。

## デフォルト

デフォルト設定なし

## 備考

サーバ上でデータベースが起動している場合は、データベース名が割り当てられています (管理者が `-n` オプションを使用して割り当てるか、データベースサーバが拡張子とパスを削除したファイルのベース名を使用して割り当てます)。

### 実行中のデータベースへの接続

接続するデータベースがすでに実行中である場合、DatabaseName (DBN) 接続パラメータを使用します。実行中のデータベースの名前が `database-name` と一致する場合のみ、接続できます。

ユーティリティデータベースに接続するには、データベース名 `utility_db` を使用します。

データベースが実行中でない場合に、起動されないようにするためには、DatabaseFile (DBF) 接続パラメータよりも DatabaseName (DBN) 接続パラメータを使用することをお奨めします。

### 実行されていないデータベースを起動

接続パラメータ DatabaseName (DBN) と DatabaseFile (DBF) を両方とも指定した場合、指定したデータベース名を使用して実行中のデータベースに接続しようとします (DatabaseFile 接続パラメータは無視されます)。これに失敗した場合、DatabaseFile 接続パラメータが示すパスを使用してデータベースを自動的に起動し、そのデータベースに、DatabaseName 接続パラメータで指定した名前を割り当てようとします。

データベースに名前を割り当てるには、DatabaseSwitches (DBS) 接続パラメータに `-n` オプションを指定して使用したり、START 接続パラメータに `-n` オプションを指定して使用するよりも、DatabaseName (DBN) 接続パラメータを使用することをお奨めします。

## 例

`cities.db` という名前のデータベースを起動して、このデータベースに `Kitchener` という名前を付けるには、次の接続文字列を使用します。

```
"DBF=cities.db;DBN=Kitchener;UID=DBA;PWD=passwd"
```

`Kitchener` という名前の実行中のデータベースに接続するには、次の接続文字列を使用します。

```
"DBN=Kitchener;UID=DBA;PWD=passwd"
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

[一般的な接続シナリオ \[35 ページ\]](#)

[DatabaseName \(DBN\) プロトコルオプション \[191 ページ\]](#)

## 1.1.3.15 DatabaseSwitches (DBS) 接続パラメータ

データベース起動時に、データベース固有のオプションを提供します。

### 構文

```
{ DatabaseSwitches | DBS }=database-options
```

### 使用法

特に制限なし

### 指定可能な値

#### database-options

この文字列は、データベースファイルに適用するデータベースオプションを指定します。

データベースサーバオプションを提供するには、StartLine 接続パラメータを使用します。

### デフォルト

オプションはありません。

### 備考

この接続パラメータは、データベースサーバに接続し、そのサーバ上のデータベースを起動するために使用します。データベースサーバが実行中でない場合には、この接続パラメータは指定されたデータベースとオプションにより自動的にデータベースサーバを起動します。

接続しようとしたときに、すでにデータベースが実行中だった場合には、DatabaseSwitches 接続パラメータは無視されます。

DatabaseFile で指定したデータベースをデータベースサーバが起動するとき、サーバは提供された DatabaseSwitches を使用して、データベースの起動オプションを決定します。

### 注記

データベースに名前を割り当てるには、DatabaseSwitches (DBS) 接続パラメータに `-n` オプションを指定して使用するよりも、DatabaseName (DBN) 接続パラメータを使用することをお奨めします。

### 例

次のコマンドは、デフォルトのデータベースサーバに接続し、データベースファイル `demo.db` (DatabaseFile (DBF) 接続パラメータ) をロードし、そのファイルに `my-db` (DatabaseName (DBN) 接続パラメータ) と名前を付け、これを読み込み専用モード (`-r` オプション) で起動します。

```
dbisql -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMPl7%demo.db;DBN=my-db;DBS=-r"
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

[データベース起動オプション \[519 ページ\]](#)

[DatabaseName \(DBN\) 接続パラメータ \[68 ページ\]](#)

[DatabaseKey \(DBKEY\) 接続パラメータ \[67 ページ\]](#)

[StartLine \(START\) 接続パラメータ \[107 ページ\]](#)

## 1.1.3.16 DataSourceName (DSN) 接続パラメータ

ODBC ドライバマネージャまたは Embedded SQL ライブラリに対して、レジストリ内またはシステム情報ファイル (デフォルト名は `.odbc.ini`) での ODBC データソース情報の検索場所を指示します。

### 構文

```
{ DataSourceName | DSN }=data-source-name
```

## 使用法

特に制限なし

## 指定可能な値

### **data-source-name**

この文字列は、データベースの接続情報が含まれる ODBC データソースの名前を指定します。

## デフォルト

デフォルトのデータソース名はありません。

## 備考

データソース名だけを ODBC に送信するのは、ODBC アプリケーションの一般的な手法です。ODBC ドライバマネージャと ODBC ドライバは、接続パラメータの残りの部分を含んだデータソースを探します。

Embedded SQL アプリケーションも ODBC データソースを使用して接続パラメータを保存できます。

### 例

次のパラメータは、データソース名を使用します。

```
DSN=My Database
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[UNIX での ODBC データソース \[121 ページ\]](#)

[ODBC データソース \[111 ページ\]](#)

[FileDataSourceName \(FILEDSN\) 接続パラメータ \[80 ページ\]](#)

### 1.1.3.17 DisableMultiRowFetch (DMRF) 接続パラメータ

ネットワーク上での複数ローフェッチをオフにします。

### 構文

```
{ DisableMultiRowFetch | DMRF }={ YES | NO }
```

## 使用法

特に制限なし

## デフォルト

NO

## 備考

デフォルトでは、データベースサーバが単純なフェッチ要求を受信すると、アプリケーションは追加のローを要求します。このパラメータを YES に設定すると、この動作を無効にできます。

DisableMultiRowFetch (DMRF) 接続パラメータを YES に設定するのと、prefetch データベースオプションを Off に設定するのは、同じ効果があります。

### 例

次の接続文字列フラグメントは、プリフェッチを回避します。

```
DMRF=YES
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[prefetch オプション \[757 ページ\]](#)

### 1.1.3.18 Elevate 接続パラメータ

Elevates は、Windows 7 以降でデータベースサーバの実行プログラムを自動的に起動します。

### 構文

```
Elevate={ YES | NO }
```

## 使用方法

Windows 7 以降

## デフォルト

NO

## 備考

接続文字列に ELEVATE=YES を指定すると、昇格されていないクライアントプロセスが昇格したサーバを自動的に起動できるようになります。Windows 7 以降で、データベースファイルへのアクセスに管理権限が要求される場合、この指定が必要となる場合があります。データベースサーバが自動的に起動されていない場合、このパラメータは無視されます。

### 例

次の接続文字列フラグメントは、データベースサーバを自動的に起動して昇格します。

```
"Elevate=YES;START=dbsrv17"
```

## 1.1.3.19 EncodedPassword (ENP) 接続パラメータ

パスワード文字列がエンコード形式であることを示します。

### 構文

```
{ EncodedPassword | ENP }=password
```

## 使用法

特に制限なし

## 指定可能な値

**password**

バイナリエンコードされたパスワードの 16 進表記。

## デフォルト

なし

## 備考

標準ユーザ認証を使用してデータベースに接続する際、ユーザ ID とパスワードをデータベースサーバに提示します。Kerberos、LDAP (Lightweight Directory Access Protocol)、PAM (Pluggable Authentication Module)、Microsoft Windows 統合化ログインなどの、他のサポート済みユーザ認証フレームワークでは、データベースへの接続にパスワードを必要としません。

パスワードが要求されたとき (プロンプトが表示されたときなど) にユーザが入力することが望ましいのですが、場合によってはユーザのクレデンシャルをクライアントコンピュータに保存しておかなければならないこともあります。クライアントコンピュータにパスワードを保存すると、セキュリティに関するリスクが生じます。ユーザ ID とパスワードの両方が発見されると、そのデータベースサーバへのアクセス権を持つユーザは、そのユーザ ID とパスワードを使用して認証することが可能です。このような事態に対応するため、クライアントソフトウェアでは、パスワードをプレーンテキスト形式またはエンコード形式で保存できるようなメカニズムが提供されています。

このトピックでは、*encode* という用語は、プレーンテキスト形式の文字列をを難読化あるいは暗号化することによってエンコード形式に変換する処理を意味しています。*obfuscate* という用語は、暗号的に安全でない形態であるプレーンテキストの文字列を難読化された形式に変換する処理を意味しています。*encrypt* という用語は、暗号的に安全でない形態であるプレーンテキストの文字列を暗号化された形式に変換する処理を意味しています。

EncodedPassword 接続パラメータは、パスワードがエンコードされていることを示すために、接続文字列の中で使用されます。エンコードされた文字列は、16 進形式となります。

エンコードパスワードを作成するには、*ODBC データソースアドミニストレータの SQL Anywhere の ODBC 設定ウィンドウ* (Microsoft Windows のみ)、またはデータソースユーティリティ (dbdsn) を使用します。

これらのツールを使用することによって、エンコードパスワードがデータソースの ENP (EncodedPassword) パラメータに 16 進文字列として保存されます。

Microsoft Windows では、データソース情報は通常システムレジストリに格納されます。UNIX では、データソース情報はシステム情報ファイル (デフォルト名は `.odbc.ini`) に格納されます。

エンコードパスワードが接続要求の一部として提供されると、接続が行われる前に、クライアントソフトウェアによってデコードされ、プレーンテキストに戻ります。

Password (PWD) 接続パラメータと EncodedPassword (ENP) 接続パラメータの両方を指定した場合、Password (PWD) 接続パラメータが優先されます。

エンコードパスワードには複数のレベルがサポートされています。

1. 難読化パスワードはすべてのコンピュータで使用できます。エンコードパスワード文字列とそれに対応するユーザ ID は、データベースへの接続を開始するために、どのユーザでも、どのコンピュータ上でも使用できます。
2. パスワードは、エンコードが行われたのと同じコンピュータ上でのみデコードできます。エンコードパスワード文字列とそれに対応するユーザ ID は、エンコードパスワードが作成されたコンピュータからデータベースへの認証を行う場合のみ使

用できます。このコンピュータにログインできるユーザであれば誰でも、エンコードパスワードとそれに対応するユーザ ID を使用してデータベースへの認証を実行できます。他のコンピュータでは使用できません。

3. パスワードは、エンコードが行われたのと同じコンピュータ上で、同じユーザによってのみデコードできます。エンコードパスワード文字列とそれに対応するユーザ ID は、エンコードパスワードが作成されたコンピュータから、作成したユーザがデータベースへの認証を行う場合のみ使用できます。同じコンピュータの他のユーザはデコードすることができません。同じユーザであっても他のユーザであっても、他のコンピュータ上ではデコードすることができません。

ODBC データソースに保存するパスワードのエンコードレベルを指定するには、[ODBC データソースアドミニストレータの SQL Anywhere の ODBC 設定ウィンドウ](#) (Microsoft Windows のみ)、またはデータソースユーティリティ (dbdsn) の `-pet` オプションを使用します。

### 警告

データソースを作成する際、ユーザ ID とプレーンテキストのパスワードまたはエンコードパスワードを、定義の一部に含めないようにしてください。この機能は Microsoft Windows [ODBC データソースアドミニストレータの SQL Anywhere の ODBC 設定ウィンドウ](#)と、データソースユーティリティ (dbdsn) の両方にありますが、この情報を含めるとセキュリティに関するリスクが発生します。パスワードを含めなければならない場合には、可能な限り最善のエンコード形式を選択してください。

Microsoft Windows の場合、Microsoft Cryptography API を使用すると、上記のレベル 2 および 3 に対応する強力な暗号化が可能です。Microsoft Windows では、パスワードの暗号化に使用される暗号化キーを、内蔵のキーストアで管理します。暗号化キーは、特定のコンピュータ、または特定のユーザとコンピュータに関連付けられます。レベル 2 のエンコードパスワードは、コンピュータ上のどのユーザでもプレーンテキストにデコードすることができます。レベル 3 のエンコードパスワードは、コンピュータ上の特定のユーザだけがプレーンテキストにデコードすることができます。デコードを行うのに、パスワードは必要ありません (Microsoft Windows にログインするためのパスワードだけです)。

Microsoft Windows System ODBC データソースではレベル 3 を使用できますが、レベル 2 の方が適している場合もあります。レベル 2 のエンコードパスワードは、コンピュータのどのユーザでも使用できるためです。システム ODBC データソースを作成し使用する Microsoft Windows ユーザが 1 人だけであれば、システム ODBC データソースにレベル 3 を使用することも考えられます。

Linux や MacOS X などのシステムでは、キーストアが内蔵されていません。そのため、レベル 2 および 3 では難読化だけが行われます。難読化は、パスワードを奪おうとする攻撃を受けた場合には安全とは言えません。`.odbc.ini` ファイルに適切なパーミッションを設定することで、オペレーティングシステムレベルの保護を行ってください。

エンコードパスワードのバイナリ値は常に 16 進表記で指定されます。次はその例です。

```
ENP=03a17731bca92f9701000000d08c9ddf0115d1118c7a00c04fc297eb01...
```

特定のユーザやコンピュータ用にパスワードをエンコードすることの利点は、仮にパスワードが誤って漏れてしまったとしても、クライアントライブラリを使用して他のコンピュータシステムで使用することができないという点です。ただし、選択したレベルやオペレーティングシステムプラットフォームによっては、攻撃者によってデコードされる危険性があることに注意してください。

## 関連情報

[UNIX での ODBC データソース \[121 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

[データソースユーティリティ \(dbdsn\) \[1089 ページ\]](#)

## 1.1.3.20 Encryption (ENC) 接続パラメータ

クライアントアプリケーションとデータベースサーバとの間で送信されるパケットのエンコードを指定します。これには、エンコードなし、単純難読化、TLS (Transport Layer Security) 暗号化があります。

### 構文

```
{ Encryption | ENC }=  
{ NONE  
| SIMPLE  
| TLS [ (  
  [ FIPS={ ON | OFF }; ]  
  [ TRUSTED_CERTIFICATE={ public-certificate-filename | * }; ]  
  [ CERTIFICATE_COMPANY=organization; ]  
  [ CERTIFICATE_NAME=common-name; ]  
  [ CERTIFICATE_UNIT=organization-unit; ]  
  [ SKIP_CERTIFICATE_NAME_CHECK={ ON | OFF } ]  
  [ IDENTITY=identity-file ]  
  [ IDENTITY_PASSWORD=password ]  
  [ ALLOW_EXPIRED_CERTS= { ON | OFF } ]  
) ]  
}
```

### 使用法

TLS: TCP/IP でのみサポート

NONE または SIMPLE: どこでも使用可

### 指定可能な値

#### NONE

暗号化されていない通信パケットを受け入れます。

#### SIMPLE

単純難読化によってエンコードされた通信パケットを受け入れます。この値は、すべてのプラットフォームで、また以前のバージョンの SQL Anywhere でサポートされます。単純難読化の場合、データベースサーバの検証や強力な暗号化などの TLS の機能は提供されません。

データベースサーバが単純難読化は受け入れるが暗号化されない通信は受け入れない場合、暗号化を使用しない非 TDS 接続では、自動的に単純難読化が使用されます。

-ec SIMPLE を指定してデータベースサーバを起動すると、データベースサーバは単純難読化を使用した接続だけを受け入れます。RSA 暗号化や FIPS 認定の RSA 暗号化を必要とする TLS 接続は失敗し、暗号化を必要としない接続では単純難読化が使用されます。

## TLS

TLS を使用して暗号化された通信パケットを受け入れます。FIPS 認定の TLS の場合は、FIPS=ON と指定します。FIPS 認定 TLS は別の認定ライブラリを使用しますが、SQL Anywhere 9.0.2 以降で非認定 TLS を使用しているデータベースサーバと互換性があります。

TLS 接続を開始する際、クライアントライブラリは、データベースサーバのホスト名をサーバが提供する証明書と照らし合わせます。この照合が行われるのは、certificate\_name、certificate\_company、certificate\_unit のいずれのオプションも指定されていない場合、あるいは、skip\_certificate\_name\_check オプションが有効でない場合です。

certificate\_name、certificate\_company、certificate\_unit のいずれかが指定されていれば、それらのオプションだけが確認されます。skip\_certificate\_name\_check オプションを有効にすると、ホスト名のチェックが無効になります。

ホスト名や IP アドレスは、subjectAltName (サブジェクトの別名、SAN) 拡張、および Common Name (CN) フィールドから取得します。SAN には、ワイルドカードによって複数のホスト名が指定されている場合があります。たとえば、Google 社の証明書には、\*.google.com、\*.google.ca、\*.android.com が指定されているかもしれません。その場合、www.google.ca は有効なホスト名ということになります。

クライアントは、データベースサーバの証明書のフィールド値を検証するための次の引数をサポートしています。使用する証明書がパブリック証明書であってもオペレーティングシステムの証明書ストアから取得した証明書であっても同様です。

- CERTIFICATE\_COMPANY
- CERTIFICATE\_NAME
- CERTIFICATE\_UNIT

TLS と TLS(TRUSTED\_CERTIFICATE=\*) は同等です。これらのいずれかが指定されると、ソフトウェアはオペレーティングシステムの証明書ストアを使用し、ストアにリストされたすべての証明書を信頼します。

SKIP\_CERTIFICATE\_NAME\_CHECK=ON を指定すると、ホスト名のチェックが無効になります。このオプションを有効にすると、クライアントはデータベースサーバの認証を完全に行うことができません。

暗号化証明書のパスワードを指定するには、IDENTITY\_PASSWORD オプションを使用します。

ルート証明書を信頼し、期限切れになったかまだ有効になっていないデータベースサーバ証明書を受け入れるには、クライアントライブラリに対して ALLOW\_EXPIRED\_CERTS=ON を指定します。デフォルトは OFF です。

## デフォルト

NONE

## 備考

このパラメータを使用すると、クライアントアプリケーションとデータベースサーバとの間の通信を、トランスポートレイヤセキュリティ (TLS) または単純難読化によってエンコードできます。

FIPS 認定の暗号化には別途ライセンスが必要です。

CONNECTION\_PROPERTY システム関数を使用すると、現在の接続の暗号化設定値を取得できます。

```
SELECT CONNECTION_PROPERTY ( 'Encryption' );
```

### 例

次の接続文字列フラグメントは、TLS (RSA 暗号) と、期限切れの場合でも受け入れられる指定した証明書ファイルを使用して、データベースサーバに接続します。

```
"HOST=myhost;SERVER=myserver;ENCRYPTION=TLS (FIPS=OFF;TRUSTED_CERTIFICATE=rsaroot.crt;ALLOW_EXPIRED_CERTS=ON) "
```

次の接続文字列フラグメントは、単純難読化を使用して、データベースサーバに接続します。

```
"HOST=myhost;SERVER=myserver;ENCRYPTION=simple"
```

次の 2 つの接続文字列フラグメントは同等であり、TLS (RSA 暗号) およびオペレーティングシステムの証明書ストアから取得した証明書を使用して、データベースサーバに接続します。

```
"HOST=myhost;SERVER=myserver;ENCRYPTION=TLS (FIPS=OFF;TRUSTED_CERTIFICATE=*) "
```

```
"HOST=myhost;SERVER=myserver;ENCRYPTION=TLS "
```

次の接続文字列フラグメントは、myrootcert.crt がデータベースサーバの証明書署名チェーンのルートであることと、証明書の通称フィールドに MyCertificateName が設定されていることを確認します。

```
"HOST=myhost;SERVER=myserver;ENCRYPTION=TLS (TRUSTED_CERTIFICATE=myrootcert.crt;CERTIFICATE_NAME=MyCertificateName) "
```

次の接続文字列フラグメントは、myrootcert.crt がデータベースサーバの証明書署名チェーンのルートであることを確認します。他の TLS 証明書オプションは指定されていないので、データベースサーバの証明書内のホスト名は接続先のホスト名に一致している必要があります。

```
"HOST=myhost;SERVER=myserver;ENCRYPTION=TLS (TRUSTED_CERTIFICATE=myrootcert.crt) "
```

次の接続文字列フラグメントは、myrootcert.crt がデータベースサーバの証明書署名チェーンのルートであることを確認しますが、その他の検証は行いません。

```
"HOST=myhost;SERVER=myserver;ENCRYPTION=TLS (TRUSTED_CERTIFICATE=myrootcert.crt;SKIP_CERTIFICATE_NAME_CHECK=ON) "
```

## 関連情報

[トランスポートレイヤセキュリティを使用するようにクライアントアプリケーションを設定する \[1655 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

[デジタル証明書 \[1644 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

- ek データベースオプション [527 ページ]
- ep データベースサーバオプション [406 ページ]
- es データベースサーバオプション [407 ページ]
- DatabaseKey (DBKEY) 接続パラメータ [67 ページ]
- certificate\_company プロトコルオプション (クライアント側のみ) [186 ページ]
- certificate\_name プロトコルオプション (クライアント側のみ) [187 ページ]
- certificate\_unit プロトコルオプション (クライアント側のみ) [188 ページ]
- trusted\_certificates プロトコルオプション [234 ページ]
- skip\_certificate\_name\_check プロトコルオプション (クライアント側のみ) [228 ページ]
- allow\_expired\_certs プロトコルオプション [181 ページ]

### 1.1.3.21 EngineName (ENG) 接続パラメータ (旧式)

この接続パラメータは使用されなくなりました。代わりに、ServerName (Server) 接続パラメータを使用します。

#### 関連情報

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

### 1.1.3.22 FileDataSourceName (FILEDSN) 接続パラメータ

接続するデータベースに関する情報を ODBC ファイルデータソースが保有していることをクライアントライブラリに知らせます。

#### 構文

```
{ FileDataSourceName | FILEDSN }=file-data-source-name
```

#### 使用法

特に制限なし

#### 指定可能な値

**file-data-source-name**

この文字列は、データベースの接続情報が含まれるファイルデータソースの名前を指定します。

## デフォルト

デフォルト名はありません。

## 備考

ファイルデータソースは、レジストリに保管される ODBC データソースと同じ情報を持ちます。ファイルデータソースは、簡単にエンドユーザに配布できるので、接続情報を各コンピュータ上で再構成する必要はありません。

ODBC と Embedded SQL アプリケーションのいずれもファイルデータソースを使用できます。

## 関連情報

[Windows でのファイルデータソース \[119 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

[Windows でのファイルデータソース \[119 ページ\]](#)

[DataSourceName \(DSN\) 接続パラメータ \[71 ページ\]](#)

### 1.1.3.23 ForceStart (FORCE) 接続パラメータ

実行しているサーバがあるかどうかをチェックすることなくデータベースを起動します。

#### 構文

```
{ ForceStart | FORCE }={ YES | NO }
```

## 使用法

db\_start\_engine 関数と併用する場合のみ

## デフォルト

NO

## 備考

ForceStart=YES を設定することで、db\_start\_engine 機能は、サーバを起動する前に実行しているサーバがあるかどうかをチェックしなくなります。

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

### 1.1.3.24 Host 接続パラメータ

ホスト名または IP アドレスとオプションのポート番号を受け入れ、データベースサーバがある場所をクライアントに知らせます。

#### 構文

```
Host={ hostname | ip-address }[:port-number ] [, ...]
```

## 使用法

特に制限なし。Host 接続パラメータは、ネットワークデータベースサーバとの接続を行う場合に使用が推奨され、TCP/IP を使用することを示します。

## 指定可能な値

### hostname

データベースサーバを実行しているコンピュータの名前。ホスト値のリストはカンマで区切られていますが、オプションとしてポート番号 (コロンで区切る) を含めることができます。現在のコンピュータを識別するには、**localhost** を使用できません。

### ip-address

この文字列は、IP アドレス形式で指定する必要があり、任意でポート番号 (コロンで区切る) を含めることができます。IP アドレスのリストは、カンマで区切られています。

IPv6 アドレスの場合は、アドレスを角カッコまたはカッコで囲む必要があります。たとえば、  
Host=[fd77:ab34:2238::3894]:8933 のように指定します。8933 はポート番号です。

#### port-number

データベースサーバで使用されるポート番号。デフォルトのポート番号は 2638 です。

## デフォルト

なし

## 備考

Host 接続パラメータは、1 つ以上のホスト名 (または IP アドレス) とオプションのポート番号を指定し、データベースサーバが実行している場所をクライアントに知らせます。

Host 接続パラメータによって、データベースサーバを実行している 1 つ以上のコンピュータシステムを識別します。Host 接続パラメータが指定されていない場合、データベースサーバは自動的に起動されません。

1 つのコンピュータシステム上で複数のデータベースサーバが実行されており、ポート番号が指定されていない場合、どのデータベースサーバに接続するかを識別するため ServerName 接続パラメータを使用する必要があります。ServerName パラメータを常に使用することをお奨めします (特に埋め込みアプリケーションの場合)。

オプション	説明
サーバとポートを指定	<pre>host=myhost:1234;servername=myserver</pre> <p>指定したポートで受信を行う指定したサーバ名のデータベースサーバのみ接続が確立されます。</p>
サーバを指定し、ポートを指定しない	<pre>host=myhost;servername=myserver</pre> <p>指定したサーバ名のデータベースサーバとの接続が確立されます。</p>
サーバもポートも指定しない	<pre>host=myhost</pre> <p>ポート 2638 で受信を行うデータベースサーバに対して接続が試行されます。</p>
複数のサーバとポートを指定	<pre>host=myhost1:6871,myhost2:6872</pre> <p>指定されたホスト名とポート番号のうちのいずれかで受信するデータベースサーバとの接続が確立されます。</p>

複数のアドレスが指定された場合、次の順序でアドレスに接続が試行されます。

1. 指定されたアドレスがデータベースサーバアドレスキャッシュ (sasrv.ini) がないかどうか確認します。一致するものが見つかった場合には、データベースサーバキャッシュ内のこのアドレスが最初に試行されます。
2. アドレスは指定された順番で試行されます。

Host パラメータを使用する場合、サーバを一意に特定するための十分な情報 (ホスト名とポート番号) が指定されていれば、UDP パケットは送信されません。ポート番号とデータベースサーバ名がどちらも指定されていない場合は、ポート番号は 2638 であると想定され、クライアントはブロードキャストを実行しません。ただし、クライアントにホスト名とデータベースサーバ名があり、ポート番号がない場合は、指定されたホストのポート 2638 に UDP パケットが送信されて、ポート番号が検索されます。

HOST プロトコルオプションまたは Host 接続パラメータと共に LocalOnly プロトコルオプションを指定することはできません。

### 注記

Host 接続パラメータを指定すると、共有メモリの使用は無効になります。パーソナルデータベースサーバとの接続に Host 接続パラメータを使用するには、-x オプションを使用してデータベースサーバで TCP/IP 接続プロトコルを有効にする必要があります。また、TCP/IP を使用してパーソナルデータベースサーバに接続する場合は、ホスト名 localhost を使用する必要があります。

CommLinks (LINKS) 接続パラメータは、HOST または ServerPort (PORT) 以外の TCP/IP プロトコルオプションを指定する必要がある場合にのみ使用してください。

接続文字列に CommLinks と Host の両方は指定できません。

### 例

SalesDB という名前のデータベースサーバが Elora というコンピュータのデフォルトのポート番号で実行されていることがわかっている場合、次の接続文字列を使用して、データベースサーバに接続できます。

```
UID=DBA;PWD=passwd;Server=SalesDB;Host=Elora:2638
```

データベースサーバが実行されているポート番号が不明の場合は、次の接続文字列を使用して、Elora というコンピュータで実行されている SalesDB データベースサーバに接続します。

```
UID=DBA;PWD=passwd;Server=SalesDB;Host=Elora
```

## 関連情報

[一般的な接続シナリオ \[35 ページ\]](#)

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

[トラブルシューティング: Host 接続パラメータによるデータベースサーバの検出方法 \[255 ページ\]](#)

## 1.1.3.25 IdleTimeout (IDLE) 接続パラメータ

接続のアイドルタイムアウト時間を指定します。

### 構文

```
{ IdleTimeout | IDLE }=timeout-value
```

### 使用法

TDS 接続以外。他には特に制限なし。TDS 接続 (jConnect を含む) では、IdleTimeout 接続パラメータは無視されます。

### 指定可能な値

#### timeout-value

接続のアイドルタイムアウト時間 (分単位)。IdleTimeout 接続パラメータの最小値は 1 分で、サポートされる最大値は 32767 分です。0 を指定すると、接続に対するアイドルタイムアウトのチェックは行われません。

### デフォルト

240 分 (TCP/IP の場合)

0 (共有メモリの場合)

### 備考

IdleTimeout 接続パラメータは、現在の接続に対してのみ適用されます。同じデータベースサーバ上の複数の接続に、異なるタイムアウト値を設定できます。

IdleTimeout 接続パラメータが使用されていない場合、TCP/IP 接続のアイドルタイムアウト値は -ti データベースサーバオプションによって制御されます。-ti データベースサーバオプションと IdleTimeout 接続パラメータの両方が指定された場合、アイドルタイムアウト値は接続パラメータによって制御されます。

### 例

次の接続文字列フラグメントは、この接続のタイムアウト値を 10 分に設定します。

```
"Host=myhost;IDLE=10"
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[スキーマロックの自動解放 \(Interactive SQL\) \[1032 ページ\]](#)

[-ti データベースサーバオプション \[483 ページ\]](#)

[ネットワーク通信のトラブルシューティング \[979 ページ\]](#)

[blocking\\_others\\_timeout オプション \[663 ページ\]](#)

### 1.1.3.26 Integrated (INT) 接続パラメータ

統合化ログインを試行できるかどうかを指定します。

#### 構文

```
{ Integrated | INT }={ YES | NO }
```

## 使用法

統合化ログインは、クライアントマシンとサーバマシンの両方が Windows システムで実行されている場合のみ許可されません。

## 指定可能な値

### YES

統合化ログインを行います。接続の試行が失敗し、login\_mode オプションが 'Standard,Integrated' と設定されている場合は、標準ログインを試行します。

### NO

これはデフォルト設定です。統合化ログインは試行されません。

## デフォルト

NO

## 備考

統合化ログインを使用するクライアントアプリケーションでは、login\_mode データベースオプションを Integrated に設定してサーバを実行してください。

### 例

次のデータソースフラグメントは、統合化ログインを使用します。

```
INT=YES
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[Windows 統合化ログイン \[124 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

## 1.1.3.27 Kerberos (KRB) 接続パラメータ

データベースサーバへの接続時に Kerberos 認証を使用できるかどうかを指定します。

### 構文

```
{ Kerberos | KRB } = { YES | NO | SSPI | GSS-API-library-file }
```

## 使用法

すべてのプラットフォーム

## 指定可能な値

### YES

Kerberos 認証ログインが試行されます。

### NO

Kerberos 認証ログインは試行されません。これはデフォルトです。

### SSPI

Kerberos 認証ログインが試行され、GSS-API ライブラリの代わりに組み込みの Windows SSPI インタフェースが使用されます。SSPI は Windows プラットフォームでのみ使用できます。また、Domain Controller Active Directory KDC 以外のキー配布センター (KDC) で SSPI を使用することはできません。Windows クライアントコンピュータがすでに Windows ドメインにログインされている場合は、Kerberos クライアントをインストールまたは設定せずに、SSPI を使用できます。

#### 注記

SSPI は、クライアントのみが Kerberos 接続パラメータで使用できます。データベースサーバは SSPI を使用できません。SSPI 以外でサポートされている Kerberos クライアントを使用する必要があります。

#### GSS-API-library-file

Kerberos 認証ログインが試行されます。この文字列は、Kerberos GSS-API ライブラリ (または UNIX の共有オブジェクト) のファイル名を指定します。これは、Kerberos クライアントでデフォルトと異なる Kerberos GSS-API ライブラリファイル名が使用されているか、コンピュータに複数の GSS-API ライブラリがインストールされている場合にだけ必要です。

## デフォルト

NO

## 備考

Kerberos 認証ログインを使用している場合、UserID および Password 接続パラメータは無視されます。

Kerberos 認証を使用するには、Kerberos クライアントがインストールおよび設定され (SSPI の場合は不要)、ユーザが Kerberos にログインされ (有効なチケット付与権限付きチケットがある)、データベースサーバで Kerberos 認証ログインが有効になり設定されている必要があります。

#### 例

```
Kerberos=YES  
Kerberos=SSPI  
Kerberos=c:¥Program Files¥MIT¥Kerberos¥bin¥gssapi32.dll
```

## 関連情報

[Kerberos ユーザ認証 \[149 ページ\]](#)

[Windows で Kerberos ログインに SSPI を使用した接続 \[158 ページ\]](#)

[-kl データベースサーバオプション \[441 ページ\]](#)

[-kr データベースサーバオプション \(廃止予定\) \[443 ページ\]](#)

[-krb データベースサーバオプション \[444 ページ\]](#)

## 1.1.3.28 Language (LANG) 接続パラメータ

接続の言語を指定します。

### 構文

```
{ Language | LANG }=language-code
```

### 使用法

特に制限なし

### 指定可能な値

#### language-code

2 文字の組み合わせで言語を表します。たとえば、LANG=DE の場合、デフォルト言語をドイツ語に設定します。

### デフォルト

SALANG 環境変数、dblang ユーティリティ、インストーラの順に指定された言語

### 備考

この接続パラメータは接続用の言語を設定するものです。サーバが指定された言語をサポートしている場合に、サーバからのエラーや警告が指定された言語で配信されます。

言語を指定しない場合は、デフォルトの言語が使用されます。デフォルトの言語は、SALANG 環境変数、dblang ユーティリティ、インストーラの順で指定された言語です。

この接続パラメータは接続のみに影響します。ツールとユーティリティから返されるメッセージはデフォルトの言語で表示されますが、サーバから戻されるメッセージは接続の言語で表示されます。

### 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[言語ラベルの値および言語コード \[588 ページ\]](#)

## 1.1.3.29 LazyClose (LCLOSE) 接続パラメータ

カーソル要求を次の要求が発生するまでキューイングするか、またはただちに実行するかを制御します。

カーソルを閉じる要求をキューイングすると、往復がなくなり、パフォーマンスが向上します。

### 構文

```
{ LazyClose | LCLOSE }={ YES | NO | AUTO }
```

### 使用法

特に制限なし

### 指定可能な値

#### YES

カーソルを閉じる要求を常にキューイングするため往復がなくなりますが、クライアントによってカーソルが閉じられた後に、ロックが発生したり、他のリソースが保留状態になる場合があります。同じ接続でデータベースサーバに次の要求が送信されると、カーソルを閉じる要求が実行されます。CLOSE *cursor-name* データベース要求がキューイングされている間は、独立性レベル 1 のすべてのカーソル安定性ロックがカーソルに適用されます。

#### NO

カーソルをただちに閉じます。

#### AUTO

ロックや大量のサーバリソースの保留状態が持続する時間が変わらない場合のみ、カーソルを閉じる要求をキューイングして往復をなくします。カーソルが独立性レベル 1 のカーソル安定性ロックを使用する場合、またはカーソルが閉じられるまでサーバリソースが解放されないために大量のリソースを消費する可能性がある場合、カーソルはただちに閉じられません。ワークテーブルを必要とするクエリなどは、大量のサーバリソースを消費する可能性の高いカーソルの例です。

### デフォルト

AUTO

### 備考

この接続パラメータを YES または AUTO に設定すると、カーソルは次のデータベース要求が発生するまで閉じられません。

ネットワークの遅延時間が長い場合、またはアプリケーションでカーソルを開く要求と閉じる要求を多数送信する場合は、このオプションを有効にすると、パフォーマンスが向上します。

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

### 1.1.3.30 LivenessTimeout (LTO) 接続パラメータ

不完全な接続の終了を制御します。

#### 構文

```
{ LivenessTimeout | LTO }=timeout-value
```

## 使用法

ネットワークサーバのみ。

非スレッド化 UNIX アプリケーションを除くすべてのプラットフォーム

## 指定可能な値

### **timeout-value**

接続の活性タイムアウト期間 (秒単位)。LivenessTimeout 接続パラメータの最小値は 30 秒で、最大値は 32767 秒です。0 に指定すると、接続に対する活性タイムアウトのチェックがオフになります。最小値より小さい 0 以外の値は最小値に再設定されます。たとえば、"LivenessTimeout=5" が含まれている接続文字列を指定すると、

"LivenessTimeout=30" が使用されます。LivenessTimeout 値が設定されていない場合、LivenessTimeout はサーバ設定で制御されます。デフォルトは 120 秒です。

## デフォルト

なし

## 備考

接続が維持されていることを確認するため、クライアント/サーバの TCP/IP 通信プロトコルを介して、定期的に活性パケットが送信されます。活性要求や応答パケットを検出することなく、指定した LivenessTimeout 期間にわたってクライアントが実行されていると、通信は切断されます。

LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で接続がパケットを送信しない場合に、活性パケットが送信されます。

サーバへの接続数が 200 を超えると、サーバは指定された LivenessTimeout 値に基づいて、それより大きい LivenessTimeout 値を自動的に算出します。これにより、サーバはより多くの接続を効率よく処理できます。

またこのパラメータは、[SQL Anywhere の ODBC 設定ウィンドウの詳細タブ](#)で、[活性タイムアウトテキストボックス](#)に値を入力して設定することもできます。

### 例

次の接続文字列フラグメントでは、LivenessTimeout 値を 10 分に設定します。

```
LTO=600
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[-t データベースサーバオプション \[484 ページ\]](#)

### 1.1.3.31 LogFile (LOG) 接続パラメータ

ファイルにクライアントエラーメッセージとデバッグメッセージを送信します。

#### 構文

```
{ LogFile | LOG }=filename
```

## 使用法

特に制限なし

## 指定可能な値

### filename

この文字列は、クライアントエラーメッセージとデバッグメッセージが保存されるファイルの名前を指定します。ファイル名にパスがない場合、クライアントアプリケーションの現在の作業ディレクトリを基準にします。

## デフォルト

ログファイルはありません。

## 備考

LogFile (LOG) 接続パラメータは接続ごとに固有であるため、単一のアプリケーションで、接続ごとに異なる LogFile 引数を設定できます。

一般的なログファイルの内容は次のとおりです。

```
Tue Jan 22 2013 10:59:3
10:59:31 接続を開始します:
UID=DBA;PWD=*****;DBF='C:\Users\Public\Documents\SQL Anywhere 17\Samples
\demo.db';ServerName=demo17;START='C:\Program Files\SQL Anywhere
17\Bin32\dbeng17.exe';CON=SQL_DBC_10b9d400;ASTOP=YES;LOG=c:\logs\test.txt
10:59:31 実行中のサーバに接続を開始します...
10:59:31 共有メモリに接続を開始します (sasrv.ini にキャッシュされたアドレスはありません)
10:59:31 共有メモリ経由でサーバに接続しました
10:59:31 SQL Anywhere サーババージョン 17.0.4.0.1403 に接続しました
10:59:31 アプリケーション情報:
10:59:31 IP=192.138.151.100;HOST=MyComputer1-PC;OSUSER=user1;OS='Windows 7 Build
7601 Service Pack 1';EXE='C:\Program Files\SQL Anywhere
17\bin32\dbisql.exe';PID=0x145c;THREAD=0x508;VERSION=17.0.4.0.1403;API=iAnywhereJDBC
;TIMEZONEADJUSTMENT=-300
10:59:31 サーバに接続しました。実行中のデータベースに接続を開始します...
10:59:31 [ 205] データベースに正常に接続しました。10:59:31 [ 205] プリフェッチバッファ
のために、プリフェッチローの数が 16 に減りました。10:59:31 [ 205] 制限値に達しました。
PrefetchBuffer 接続パラメータを増やすと、パフォーマンスが向上する場合があります。
```

### 例

次のコマンドラインは、LogFile (LOG) 接続パラメータを使用して、Interactive SQL を起動して、サンプルデータベースに接続します。

```
dbisql -c "DSN=SQL Anywhere 17 Demo;PWD=sql;LOG=c:\logs\test.txt"
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[LogFile \(LOG\) プロトコルオプション \[208 ページ\]](#)

### 1.1.3.32 NewPassword (NEWPWD) 接続パラメータ

パスワードの有効期限が切れている場合でも、DBA の介入を必要とせずユーザが各自でパスワードを変更できます。

#### 構文

```
{ NewPassword | NEWPWD }={ password-string | * }
```

## 使用法

新しいパスワードの入力を要求するクライアントライブラリは、Microsoft Windows でのみサポートされます。

## 指定可能な値

### password-string

ユーザが新しいパスワードを入力すると、データベースサーバはユーザ ID とパスワードの認証を実行し、login\_procedure オプションが呼び出される前にパスワードを変更しようとします。このプロセスにより、ユーザは DBA の介入なく有効期限の切れたパスワードを変更できるようになります。verify\_password\_function オプションを設定した場合は、新しいパスワードの検証が行われます。統合化ログインや Kerberos ログインで認証を実行する場合、元のパスワードは検証されず、データベースサーバは新しいパスワード値を無視し、パスワードは変更されません。

\*

Microsoft Windows の場合、特別値 \* を使用すると、既存のパスワードの有効期限が切れている場合にかぎり、接続の試行中にクライアントライブラリは新しいパスワードの入力を要求します。ユーザは既存のパスワードと新しいパスワードを入力し、さらに確認のために新しいパスワードをもう一度入力する必要があります。ユーザがフィールドに値を入力して OK をクリックすると、古いパスワードが認証され、データベースサーバはパスワードを変更しようとします。

verify\_password\_function オプションを設定した場合は、新しいパスワードの検証が行われます。ユーザのパスワードの有効期限切れを確認し、新しいパスワードの入力を要求し、パスワードの認証と変更を行うプロセスは、クライアントライブラリに対する 1 回の接続呼び出しで実行されます。

## デフォルト

パスワードは変更されず、クライアントライブラリは新しいパスワードを要求しません。

## 備考

この接続パラメータは、password\_life\_time オプションや password\_expiry\_on\_next\_login オプションを使用してログインポリシーを実装する場合に非常に便利です。また、login\_procedure で Password has expired エラーを伝えることで、パスワードの有効期限ポリシーを実装することもできます。

動作環境でパスワードの入力要求がサポートされていない場合、ユーザは Password has expired error を受信します。Microsoft Windows 環境の場合、呼び出し元のアプリケーションに最上位レベルのウィンドウが複数あったり、最小化されている場合、プロンプトウィンドウでは呼び出し元のアプリケーションウィンドウとの対話を正しく防げない (モーダルでなくなったり、正しい親ウィンドウがない) ことがあります。

Windows 環境では、ODBC SQLDriverConnect 関数を使用して DriverCompletion 引数が SQL\_DRIVER\_NOPROMPT 以外に設定されている場合、パスワードの有効期限が切れていると、接続時に新しいパスワードの入力を要求されます。DBPROP\_INIT\_PROMPT プロパティが DBPROMPT\_NOPROMPT 以外に設定されている場合、OLE DB で接続時に新しいパスワードの入力を要求されることがあります。いずれの場合も、接続パラメータ NewPassword=\* が指定されているように動作します。

### 例

次の接続文字列では、接続時にユーザ Test1 のパスワードを変更します。

```
"UID=Test1;PWD=welcome;NEWPWD=hello"
```

Windows 環境の場合、次の接続文字列では、既存のパスワードの有効期限が切れている場合に、ユーザ Test1 に新しいパスワードの入力を要求します。

```
"UID=Test1;PWD=welcome;NEWPWD=*"
```

## 関連情報

[login\\_procedure オプション \[720 ページ\]](#)

[verify\\_password\\_function オプション \[826 ページ\]](#)

[post\\_login\\_procedure オプション \[754 ページ\]](#)

### 1.1.3.33 MatView (MATVIEW) 接続パラメータ

ODBC 関数の SQLTables が呼び出されたときに、マテリアライズドビューに返される文字列を指定します。

#### 構文

```
MATVIEW=string
```

## 使用法

ODBC、OLE DB

## デフォルト

VIEW

## 備考

デフォルトでは、ODBC 関数 `SQLTables` を実行したときに、値 `VIEW` がマテリアライズドビューの `TABLE_TYPE` カラムに返されます。異なる戻り値を指定するには、`MatView` 接続パラメータを使用します。指定する文字列は、`SQLTables` 関数が `TableType` フィルタで呼び出されるときにも返されます。

### 例

`SQLTables` 関数から、タイプ `MATERIALIZED VIEW` というマテリアライズドビューが返されるようにするには、次のように接続します。

```
UID=DBA;PWD=passwd;ServerName=server-name;MATVIEW=MATERIALIZED VIEW
```

## 1.1.3.34 NodeType (NODE) 接続パラメータ

ミラーリング構成に参加している、接続先のサーバのタイプを選択します。

高可用性設定と、読み込み専用スケールアウト設定で使用します。設定によっては、接続が他のサーバにリダイレクトされ、負荷分散機能も提供されます。

### 構文

```
{ NodeType | NODE }={ DIRECT | PRIMARY | MIRROR | COPY | READONLY }
```

## 使用法

ネットワークサーバのみ。

## 指定可能な値

### DIRECT

これはデフォルト設定です。

NodeType 接続パラメータが DIRECT に設定されている場合、データベースサーバは、負荷分散またはリダイレクションを行わずに接続を受け入れます。

### PRIMARY

プライマリサーバに接続したときに、NodeType 接続パラメータが PRIMARY に設定されている場合、接続は受け入れられます。非プライマリサーバ (ミラーサーバまたはコピーノードなど) に接続した場合、データベースサーバは接続をプライマリサーバにリダイレクトします。

### COPY

NodeType 接続パラメータが COPY にセットされているとき、データベースサーバは負荷分散を行い、接続のコピーノードを選択します。読み込み専用スケールアウトシステムでは、データベースサーバはその分岐内のコピーノード (ルートノードでない場合はサーバ自体も) を検査し、負荷が最も軽いコピーノードを選択します。データベースサーバがサーバ自体を選択しない場合、選択したデータベースサーバにクライアントをリダイレクトします。

### MIRROR

ミラーサーバに接続したときに、NodeType 接続パラメータが MIRROR に設定されている場合、接続は受け入れられません。ミラーでないサーバに接続した場合、データベースサーバは接続をミラーサーバにリダイレクトします。

### READONLY

NodeType 接続パラメータが READONLY に設定されているとき、コピーノードかミラーかに関わらず、任意の読み込み専用サーバに接続されます。コピーノードがない場合は、MIRROR 設定と同様になります。コピーノードのみある場合は、COPY 設定と同様になります。

## デフォルト

DIRECT

## 備考

この接続パラメータは、指定されたロールのタイプを実行するサーバを選択するためにクライアントアプリケーションが使用します。クライアントが最初に接続されたデータベースサーバでは、どのデータベースサーバが接続を処理するかが判断され、必要に応じて、データベースサーバのアドレスを返すことによって、適切なデータベースサーバに接続がリダイレクトされます。クライアントは元のデータベースサーバから自動的に切断され、適切なデータベースサーバに自動的に接続されます。

COPY を指定すると、クライアントが最初に接続されたデータベースサーバが含まれる分岐で負荷分散が発生します。分岐は、そのデータベースサーバとその子で構成されます。この機能は、地理的に離れているサーバ間で負荷分散を実行する場合に便利です。

NodeType が指定されていると、通常、アプリケーションはルートノードに接続され、負荷が最も低いコピーノードがデータベースサーバによって決まります。その時点で、そのコピーノードに接続がリダイレクトされます。このような接続と切断をアプリケーションで短時間に何度も行うと、接続がプールされ、どのコピーサーバを使用するかがルートノードに問われなくなりま

す。接続プールを使用すると、ルートノードでの負荷は削減されますが、予想する結果にならない可能性があります。接続のたびに接続先となるコピーノードをルートサーバで確実に判断するため、接続のプールを禁止するようアプリケーションで指定することができます。

クライアント接続が異なるデータベースサーバにリダイレクトされると、ServerName (Server) 接続パラメータで指定された値と Name データベースサーバプロパティの値は一致しません。

このパラメータを使用して、次のユーザ認証方法のうちのいずれかを使用する接続をリダイレクトしている場合、スケールアウトツリーのすべてのデータベースサーバを次のように設定する必要があります。

#### 統合化ログイン

(Windows のみ) スケールアウトツリーのすべてのデータベースサーバは、同一の Windows ユーザグループの一部であるか、同一の Windows ユーザがアクセスすることを許可するように設定する必要があります。

#### Kerberos ユーザ認証

スケールアウトツリーのすべてのデータベースサーバでは、Kerberos ユーザ認証を有効にしておく必要があります。スケールアウトツリーのすべてのサーバに対する Kerberos サーバプリンシパルは、同一の領域を持つ必要があります。

#### LDAP ユーザ認証

スケールアウトツリーのすべてのデータベースサーバでは、LDAP ユーザ認証を有効にしておく必要があります。LDAP サーバとの通信で TLS 暗号化を有効化するには、各データベースサーバは信頼できる証明書ファイルへアクセスする必要があります。このため、各コンピュータの同じファイルパスを使用して、信頼できる認証ファイルを格納する必要があります。

LDAP サーバでの認証が成功したときは常に、格納された値が異なっていれば、データベースのユーザのパスワードを更新しようとします。これはデータベースが読み込み専用かどうかによって、成功する場合と失敗する場合があります。また、更新したパスワードは他のノードに複製されません。このため、パスワードが変更され LDAP サーバが利用できない場合は、標準のユーザ認証へのフェイルオーバーが失敗することがあります。

#### PAM ユーザ認証

(UNIX のみ) スケールアウトツリーのすべてのデータベースサーバでは、PAM ユーザ認証を有効にしておく必要があります。このため、各コンピュータには適切な PAM 設定が定義されている必要があります。

PAM サーバでの認証が成功したときは常に、格納された値が異なっていれば、データベースのユーザのパスワードを更新しようとします。これはデータベースが読み込み専用かどうかによって、成功する場合と失敗する場合があります。また、更新したパスワードは他のノードに複製されません。このため、パスワードが変更され PAM サービスが利用できない場合は、標準のユーザ認証へのフェイルオーバーが失敗することがあります。

データベースミラーリングと読み取り専用のスケールアウトには、それぞれ別個のライセンスが必要です。

## 関連情報

[データベースミラーリング \[1660 ページ\]](#)

[読み込み専用スケールアウト \[1733 ページ\]](#)

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

## 1.1.3.35 Password (PWD) 接続パラメータ

接続時のパスワードを指定します。

### 構文

```
{ Password | PWD }=password-string
```

### 使用法

特に制限なし

### 指定可能な値

#### password-string

パスワードは、最大長が 255 バイトで、大文字と小文字が区別されます。パスワードには、前後のスペースやセミコロンを含めることができます。

### デフォルト

パスワードの指定なし

### 備考

すべてのデータベースユーザにはパスワードが設定されています。標準認証を使用している場合、データベースへの接続時にユーザがパスワードを入力します。これ以外の認証方式では、パスワードは必要ありません。

デフォルトでは、パスワードが 6 バイトの長さである必要があります。この要件を変更するには、min\_password\_length データベースオプションを設定します。

アプリケーションは、Password (PWD) 接続パラメータを使用して接続文字列の中にプレーンテキストとしてパスワードを含めることもできます。アプリケーションは、EncodedPassword (ENP) 接続パラメータを使用して接続文字列の中にエンコードされたテキストとしてパスワードを含めることもできます。Password (PWD) 接続パラメータと EncodedPassword (ENP) 接続パラメータの両方を指定した場合、Password (PWD) 接続パラメータが優先されます。

パスワードは、データベースサーバに送信される前にクライアントにより暗号化されます。

## 警告

データソースを作成するときに、パスワードを定義の一部として含めないでください。Windows *ODBC データソースアドミニストレータ* の *SQL Anywhere* の *ODBC 設定* ウィンドウと、*SQL Anywhere* データソースユーティリティ (dbdsn) の両方でパスワードを指定できますが、この情報をデータソースに含めるとセキュリティに関するリスクが発生します。

UNIX では、データソース情報はシステム情報ファイル (デフォルト名は `.odbc.ini`) に格納されます。

## 例

次の接続文字列フラグメントは、ユーザ ID とパスワードを指定します。

```
UID=DBA;PWD=passwd
```

## 関連情報

[パスワードおよびユーザ ID の制限事項と考慮事項 \[1554 ページ\]](#)

[セキュリティ: パスワード \[1602 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

[UNIX での ODBC データソース \[121 ページ\]](#)

[Userid \(UID\) 接続パラメータ \[109 ページ\]](#)

[EncodedPassword \(ENP\) 接続パラメータ \[74 ページ\]](#)

[min\\_password\\_length オプション \[736 ページ\]](#)

## 1.1.3.36 PrefetchBuffer (PBUF) 接続パラメータ

ローをバッファするためのメモリの最大容量を、バイト単位で設定します。

### 構文

```
{ PrefetchBuffer | PBUF }=buffer-size[ k | m ]
```

## 使用法

特に制限なし

## 指定可能な値

### buffer-size

この値はバイト単位ですが、k または m を使用して、キロバイト単位またはメガバイト単位で指定することもできます。この接続パラメータには、64 KB ~ 8 MB の値を指定できます。

以前のバージョンとの互換性を保つため、16384 未満の値が指定されるとキロバイト単位だと解釈されます。

PrefetchBuffer 接続パラメータにおいて、k サフィックスを使用しないキロバイト単位での指定は廃止されました。

## デフォルト

512 KB (524288)

## 備考

PrefetchBuffer (PBUF) 接続パラメータは、プリフェッチされたローを格納するためにクライアントで割り付けられる接続ごとの最大メモリを制御します。

状況によっては、プリフェッチされるローの数を増やすと、クエリのパフォーマンスが向上することがあります。プリフェッチされるローの数は、PrefetchRows (PROWS) と PrefetchBuffer (PBUF) 接続パラメータを使用して増やすことができます。

PrefetchBuffer (PBUF) 接続パラメータを増やすと、GET DATA 要求のバッファに使用できるメモリ容量も増えます。多数の GET DATA (SQLGetData) 要求を処理するアプリケーションでは、このように設定するとパフォーマンスが向上します。

### 例

次の接続文字列フラグメントを使用して、PrefetchBuffer によるメモリ制限によってプリフェッチされるローの数が減っているかどうかを判断できます。

```
...PrefetchRows=100;LogFile=c:¥client.txt
```

次の文字列を使用して、メモリ制限を 2 MB に増やすことができます。

```
...PrefetchRows=100;PrefetchBuffer=2M
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[PrefetchRows \(PROWS\) 接続パラメータ \[102 ページ\]](#)

### 1.1.3.37 PrefetchOnOpen 接続パラメータ

パラメータが有効であるときに、カーソルを開く要求を含むプリフェッチ要求が送信されます。

#### 構文

```
PrefetchOnOpen={ YES | NO }
```

#### 使用法

ODBC

#### デフォルト

NO

#### 備考

このオプションを有効にすると、カーソルを開く要求を含むプリフェッチ要求が送信されます。これにより、カーソルを開くたびにローをフェッチするネットワーク要求は行われなくなります。カーソルを開くときにプリフェッチを実行するには、カラムをバインドしておきます。PrefetchOnOpenを使用するときに、カーソルを開いた後で最初のフェッチの前にカラムを再バインドすると、パフォーマンスが低下することがあります。

結果セットを返すクエリまたはストアードプロシージャで ODBC の SQLExecute または SQLExecDirect を呼び出すと、カーソルが開きます。

次の場合は、このオプションを有効にするとパフォーマンスが向上します。

- ネットワークの遅延時間が長い。
- アプリケーションがカーソルを開く要求と閉じる要求を多数送信する。

### 1.1.3.38 PrefetchRows (PROWS) 接続パラメータ

データベースのクエリ時にプリフェッチされる、最初に推奨されるローの数を指定します。

#### 構文

```
{ PrefetchRows | PROWS }=number-of-rows
```

## 使用法

特に制限なし

## 指定可能な値

### number-of-rows

プリフェッチされるローの数は、PrefetchRows (PROWS) 接続パラメータと PrefetchBuffer (PBUF) 接続パラメータの両方によって制限されており、そのため、プリフェッチされたローの格納に使用できるメモリが制限されます。

プリフェッチできるロー数の最大値は 10000 です。

## デフォルト

10

ADO.NET の場合は 200

## 備考

デフォルトでは、プリフェッチローの数を増やすことによってパフォーマンスが向上するカーソルに対して、プリフェッチローの数がクライアントライブラリで動的に増加されます。この動作のため、PrefetchRow のデフォルトを増加しても、ほとんどのアプリケーションではパフォーマンスがあまり向上しません。

PrefetchRow のデフォルトを増加することによってパフォーマンスが向上する可能性があるのは、STATIC または FORWARD ONLY FOR UPDATE カーソルタイプを使用しているほとんどの場合には次の操作をフェッチし、位置付け更新や削除をほとんど行わない ODBC、OLE DB、SQL Anywhere JDBC アプリケーションの場合であり、特にクライアントとデータベースサーバが低速ネットワークまたは広域ネットワークを介して通信している場合です。

プリフェッチされるローの数は、プリフェッチされるローの保管に使用できるメモリ量を制限する PrefetchBuffer (PBUF) 接続パラメータによって制限されます。

プリフェッチできるロー数の最大値は 10000 です。

### 例

次の接続文字列フラグメントは、プリフェッチされるローの数を 100 に設定します。

```
...PrefetchRows=100;...
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[PrefetchBuffer \(PBUF\) 接続パラメータ \[100 ページ\]](#)

### 1.1.3.39 RetryConnectionTimeout (RetryConnTO) 接続パラメータ

クライアントライブラリ (DBLIB、ODBC、ADO など) に、サーバが見つからない場合は指定した時間が経過するまで接続を試行し続けるよう指示します。

#### 構文

```
{ RetryConnectionTimeout | RetryConnTO }=timeout-value
```

## 使用法

特に制限なし

## 指定可能な値

### timeout-value

この接続パラメータで指定される値は、秒単位のタイムアウトです。接続試行のリトライ回数を示すカウンタ値ではありません。デフォルト値の 0 は、接続を 1 度だけ試行することを示します。

## デフォルト

0

## 備考

間隔は 0.5 秒で、接続しようとしてデータベースサーバが見つからなかった場合にだけリトライされます。それ以外のエラーは、すぐに返されます。データベースサーバが見つからなかった場合、少なくとも `RetryConnectionTimeout` 接続パラメータに指定されている期間、接続が試行されます。

デフォルトの TCP タイムアウトは 5 秒です。接続文字列に 5 秒未満に指定された `RetryConnTO` が含まれている場合でも (`Host=host-name;RetryConnTO=3` など)、5 秒間は接続が試行されます。

## 例

次の接続文字列フラグメントは、接続試行を最低 5 秒間リトライし続けるようクライアントライブラリに指示します。

```
...RetryConnTO=5;...
```

## 関連情報

[Timeout \(TO\) プロトコルオプション \[232 ページ\]](#)

### 1.1.3.40 ServerName (Server) 接続パラメータ

接続する実行中のデータベースサーバ名を指定します。

#### 構文

```
{ ServerName | Server }=database-server-name-string
```

## 使用法

特に制限なし

## 指定可能な値

### database-server-name-string

データベースサーバを自動的に起動する場合、このパラメータを使ってサーバ名を指定できます。

サーバ名は、クライアントコンピュータの文字セットに従って解釈されます。サーバ名に非 ASCII 文字を使用することは推奨できません。

この名前は、有効な識別子であることが必要です。データベースサーバ名には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロン、フォワードスラッシュ (/)、またはバックスラッシュ (\) を含む値
- 長さが 250 バイトを超える値
- UNIX 上で実行するときにスペースを含む値

データベースサーバのロングネームは、プロトコルごとに異なる長さにトランケートされます。

Windows と UNIX では、データベースサーバがバージョン 10.0.0 以降で、名前が次の長さを超えている場合、バージョン 9.0.2 以前のクライアントから接続することはできません。

- Windows 共有メモリの場合は、40 バイト
- UNIX 共有メモリの場合は、31 バイト
- TCP/IP の場合は 40 バイト

## デフォルト

デフォルトのローカルデータベースサーバ

## 備考

EngineName と ENG は、下位互換性を保つために受け入れられますが、廃止されました。

データベースサーバは起動するとき、そのコンピュータのデフォルトのデータベースサーバになろうとします。デフォルトのサーバがない場合、最初に起動したデータベースサーバが、デフォルトのデータベースサーバになります。各 Windows セッションには、独自のデフォルトデータベースサーバがあります。そのコンピュータで、データベースサーバ名を明示的に指定しないで共有メモリに接続しようとする、デフォルトのサーバに接続されます。

デフォルトのローカルデータベースサーバに接続する場合は、ServerName は必要ありません。

Host 接続パラメータを使用している場合、ホスト名とポートの両方が指定されているか、あるいはホスト上でデフォルトのポートを使用して実行しているサーバが 1 つだけであれば、ServerName は必要ありません。

接続ウィンドウや *SQL Anywhere* の ODBC 設定ウィンドウの場合、サーバ名フィールドがこれに相当します。

### i 注記

配備されたアプリケーションの接続文字列には、ServerName パラメータを含めることをおすすめします。これにより、コンピュータで複数のデータベースサーバが実行されている場合に、アプリケーションが確実に正しいサーバに接続できるため、タイミングに依存する接続エラーを防ぐことができます。

データベースサーバがデフォルトデータベースサーバとならないようにするには、配備されたアプリケーションにより使用されるデータベースサーバに -xd オプションを使用します。すべてのクライアントは、ServerName 接続パラメータを使用して接続する必要がある接続先のデータベースサーバの名前を明示的に指定する必要があります。このようにすると、ホスト上で複数のデータベースサーバが実行されているときにも、アプリケーションは正しいデータベースサーバに接続します。

### 例

サーバ Guelph に接続します。

```
Server=Guelph
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[組み込みデータベースで使用する接続パラメータ \[41 ページ\]](#)

[-n データベースサーバオプション \[448 ページ\]](#)

[-xd データベースサーバオプション \[504 ページ\]](#)

### 1.1.3.41 StartLine (START) 接続パラメータ

アプリケーションからローカルデータベースサーバを起動します。

#### 構文

```
{ StartLine | START }=local-database-server-command
```

## 使用法

ローカルデータベースサーバ

## 指定可能な値

### local-database-server-command

デフォルトでは、クライアントは、実行中のデータベースサーバに接続しようとします。指定された接続パラメータでサーバが見つからない場合、クライアントは `local-database-server-command` を使用して、新しいローカルデータベースサーバを自動的に起動します。Host 接続パラメータが指定されているか、CommLinks (LINKS) パラメータに TCPIP が含まれている場合、データベースサーバは自動的に起動しません。

## デフォルト

dbeng17

## 備考

StartLine 接続パラメータは、指定のデータベースサーバに接続できない場合にデータベースサーバを起動するためにのみ使用されます。

たとえば、次のように、データベースを実行するデータベースサーバを起動するとします。

```
dbeng17 c:¥mydb.db
```

別のデータベースに接続します (ServerName 接続パラメータを使用したデータベースサーバ名の指定は行いません)。

```
dbisql -c "START=dbsrv17 -x none -c 8M;DBN=seconddb;DBF=c:¥myseconddb.db;UID=DBA;PWD=passwd"
```

この場合、dbsrv17 データベースサーバは起動しません。その代わりに、mydb.db の起動に使用された dbeng17 データベースサーバが、myseconddb.db を起動して接続するために使用されます。

ただし、Server=server-name が指定されており、server-name という名前のデータベースサーバが実行されていない場合は、dbsrv17 データベースサーバが起動されます。

### 注記

データベース名、データベースファイル、サーバ名を指定する場合は、StartLine 接続パラメータでデータベースサーバオプションを指定するのではなく、DatabaseName (DBN)、DatabaseFile (DBF)、ServerName(Server) 接続パラメータを使用することをお奨めします。

次のコマンドでは、推奨される構文が使用されています。

```
START=dbsrv17 -c 8M;Server=myserver;DBF=c:¥sample.db;DBN=mydb
```

次の構文はお奨めしません。

```
START=dbsrv17 -c 8M -n myserver "c:¥sample.db"
```

### 例

次のデータソースフラグメントは、8 MB のキャッシュでパーソナルデータベースサーバを起動します。

```
"DBF=C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db;StartLine=dbeng17 -c 8M"
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[組み込みデータベースで使用する接続パラメータ \[41 ページ\]](#)

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

### 1.1.3.42 Unconditional (UNC) 接続パラメータ

データベースサーバへの接続があるときでも、db\_stop\_engine 関数を使用してデータベースサーバを停止します。または、db\_stop\_database 関数を使用してデータベースを停止します。

#### 構文

```
{ Unconditional | UNC }={ YES | NO }
```

#### 使用法

db\_stop\_engine 関数と db\_stop\_database 関数のみ。

#### デフォルト

NO

#### 備考

db\_stop\_engine 関数は、データベースサーバを停止します。db\_stop\_database 関数は、データベースを停止します。db\_stop\_database 関数または db\_stop\_engine 関数に渡される接続文字列で UNC=YES を指定すると、データベースサーバまたはデータベースはアクティブな接続があるときでも停止されます。UNC が YES に設定されていない場合は、アクティブな接続がないときのみデータベースサーバまたはデータベースが停止されます。

#### 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

### 1.1.3.43 Userid (UID) 接続パラメータ

データベースへのログインに使用するユーザ ID を指定します。

#### 構文

```
{ Userid | UID }=userid
```

## 使用法

特に制限なし

## 指定可能な値

### userid

ユーザ ID には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロンを含む値

## デフォルト

なし

## 備考

データベースに接続するときは、統合化ログインまたは Kerberos ログインを使用している場合を除き、必ずユーザ ID を指定します。

### 例

次の接続文字列フラグメントは、ユーザ ID DBA とパスワード passwd を指定します。

```
UID=DBA;PWD=passwd
```

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

[ユーザのセキュリティ \(ロールと権限\) \[1453 ページ\]](#)

[Password \(PWD\) 接続パラメータ \[99 ページ\]](#)

## 1.1.4 ODBC データソース

Microsoft のオープンデータベースコネクティビティ(ODBC) は、クライアントアプリケーションを Windows ベースのデータベース管理システムに接続するための標準のアプリケーションプログラミングインタフェースです。

多くのクライアントアプリケーションは、アプリケーション開発システムも含め、ODBC インタフェースを使用してデータベースにアクセスします。データベースに接続するときは、ODBC アプリケーションは ODBC データソースを使用するのが一般的です。他にも多くの種類のアプリケーションが ODBC データソースの使用をサポートしています (OLE DB や .NET など)。

SAP Open Client または jConnect とは ODBC データソースを使用できません。

SQL Anywhere ODBC ドライバは *SQL Anywhere17* という名前です。

ODBC データソースは一連の接続パラメータで、レジストリまたはファイルに格納されています。

### 警告

ユーザ ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはお奨めしません。

このセクションの内容:

[ODBC データソースの作成 \(ODBC データソースアドミニストレータの場合\) \[112 ページ\]](#)

ODBC データソースアドミニストレータを使用して ODBC データソースを作成します。

[ODBC データソースの編集 \(ODBC データソースアドミニストレータの場合\) \[113 ページ\]](#)

ODBC データソースを編集します。

[ODBC データソースの作成 \(dbdsn ユーティリティを使用する場合\) \[115 ページ\]](#)

ODBC データソースを作成します。

[Mac OS X での ODBC データソースの作成 \[116 ページ\]](#)

ODBC アドミニストレータツールを使用して ODBC データソースを作成します。

[Windows でのファイルデータソース \[119 ページ\]](#)

ファイルデータソースは、ファイルとして保存されるもので、システムレジストリに保存される ODBC データソースの代わりになります。

[ODBC ファイルデータソースの作成 \(ODBC データソースアドミニストレータの場合\) \[120 ページ\]](#)

ODBC データソースアドミニストレータで、ODBC ファイルデータソースを作成します。

[UNIX での ODBC データソース \[121 ページ\]](#)

UNIX オペレーティングシステムでは、ODBC データソースはシステム情報ファイルに格納されます。

## 関連情報

[セキュリティ: パスワード \[1602 ページ\]](#)

## 1.1.4.1 ODBC データソースの作成 (ODBC データソースアドミニストレータの場合)

ODBC データソースアドミニストレータを使用して ODBC データソースを作成します。

### コンテキスト

Windows では、システムデータソースをコンピュータシステムのすべてのユーザおよび Windows サービスが利用できます。ユーザデータソースは、作成したユーザだけしか利用できません。

#### i 注記

##### 64 ビット Windows でのシステムデータソースの作成

システムデータソースは、コンピュータシステムのすべてのユーザと Windows サービスが利用できます。64 ビットバージョンの Windows では、64 ビットアプリケーション用と 32 ビットアプリケーション用の 2 つのシステムデータソースが Windows レジストリに管理されています。32 ビットアプリケーションにアクセス可能なシステムデータソースを作成するには、32 ビット ODBC データソースアドミニストレータ (%windir%\SysWOW64 フォルダにある) を実行する必要があります。接続の問題を回避するため、64 ビットと 32 ビットの両方のシステムデータソースを作成し、まったく同じ設定にしておくことをお勧めします。

##### 64 ビット Windows でのユーザデータソースの作成

ユーザデータソース定義は、Windows レジストリの一部として保存され、システムに現在ログインしているユーザ用の設定を含んでいます。64 ビットバージョンの Windows では、コンピュータの各ユーザごとに 1 つのユーザデータソースが Windows レジストリに管理されています。ユーザデータソースを作成するには、ODBC データソースアドミニストレータの 32 ビットと 64 ビットのどちらのバージョンも使用できます。このデータソースは、一致する ODBC ドライバがインストールされている限り、64 ビットのクライアントアプリケーションでも 32 ビットのクライアントアプリケーションでも使用できます。

### 手順

1. 32 ビットの ODBC データソースアドミニストレータの場合、**スタート** > **プログラム** > **SQL Anywhere17** > **管理ツール** > **ODBC データソースアドミニストレータ (32 ビット)** をクリックします。

64 ビットの ODBC データソースアドミニストレータの場合、**スタート** > **プログラム** > **SQL Anywhere17** > **管理ツール** > **ODBC データソースアドミニストレータ (64 ビット)** をクリックします。

2. 現在のユーザの ODBC データソースを作成するには、**ユーザ DSN** タブをクリックします。

システムデータソースを作成するには、**システム DSN** タブをクリックします。

3. **追加** をクリックします。
4. **名前リスト** で、**SQL Anywhere17** を選択します。**完了** をクリックします。
5. ODBC データソースの接続パラメータを指定します。

### 警告

ユーザ ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはお奨めしません。

6. **OK** をクリックします。
7. **OK** をクリックして、**ODBC データソースアドミニストレータ** を閉じます。

## 結果

ODBC データソースが作成されます。

64 ビット Windows 上でシステムデータソースの作成を行う場合、同じ接続パラメータを使用して 32 ビットと 64 ビットのシステムデータソースを作成する必要があります。

## 1.1.4.2 ODBC データソースの編集 (ODBC データソースアドミニストレータの場合)

ODBC データソースを編集します。

## 前提条件

ODBC データソースを定義する必要があります。

## コンテキスト

### 注記

#### 64 ビット Windows でのシステムデータソースの編集

システムデータソースは、コンピュータシステムおよび Windows サービスのすべてのユーザが利用できます。64 ビットバージョンの Windows では、64 ビットアプリケーション用と 32 ビットアプリケーション用の 2 つのシステムデータソースが Windows レジストリに管理されています。32 ビットアプリケーションにアクセス可能なシステムデータソースを編集するには、32 ビット ODBC データソースアドミニストレータ (%windir%\SysWOW64 フォルダにある) を実行する必要があります。接続の問題を回避するため、64 ビットと 32 ビットの両方のシステムデータソースを保持し、まったく同じ設定にしておくことをお奨めします。

#### 64 ビット Windows でのユーザデータソースの編集

ユーザデータソース定義は、Windows レジストリの一部として保存され、システムに現在ログインしているユーザ用の設定を含んでいます。64 ビットバージョンの Windows では、コンピュータの各ユーザごとに 1 つのユーザデータソ

スが Windows レジストリに管理されています。ユーザデータソースを編集するには、ODBC データソースアドミニストレータの 32 ビットと 64 ビットのどちらのバージョンも使用できます。このデータソースは、一致する ODBC ドライバがインストールされている限り、64 ビットのクライアントアプリケーションでも 32 ビットのクライアントアプリケーションでも使用できます。

## 手順

1. 32 ビットの ODBC データソースアドミニストレータの場合、**スタート** > **プログラム** > **SQL Anywhere17** > **管理ツール** > **ODBC データソースアドミニストレータ (32 ビット)** をクリックします。  
64 ビットの ODBC データソースアドミニストレータの場合、**スタート** > **プログラム** > **SQL Anywhere17** > **管理ツール** > **ODBC データソースアドミニストレータ (64 ビット)** をクリックします。
2. **ユーザ DSN** または **システム DSN** のいずれかのタブを選択します。
3. **名前リスト** でデータソースをクリックします。
4. **設定** をクリックします。
5. ODBC データソースの接続パラメータを編集します。

### 警告

ユーザ ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはお奨めしません。

6. **OK** をクリックします。
7. **OK** をクリックして、**ODBC データソースアドミニストレータ** を閉じます。

## 結果

ODBC データソースが編集され、保存されます。

64 ビット Windows 上でシステムデータソースの更新を行っている場合、32 ビットと 64 ビットのシステムデータソースに同じ更新が適用されるようにする必要があります。

### 1.1.4.3 ODBC データソースの作成 (dbdsn ユーティリティを使用する場合)

ODBC データソースを作成します。

#### コンテキスト

Windows では、システムデータソースをコンピュータシステムのすべてのユーザおよび Windows サービスが利用できます。ユーザデータソースは、作成したユーザだけしか利用できません。dbdsn ユーティリティではファイルデータソースは作成できません。ファイルデータソースを作成するには、ODBC データソースアドミニストレータを使用してください。

Linux およびその他の Windows 以外のシステムでは、dbdsn ユーティリティを使用して ODBC データソースを作成できません。

#### 注記

##### 64 ビット Windows でのシステムデータソースの作成

システムデータソースは、コンピュータシステムのすべてのユーザと Windows サービスが利用できます。64 ビットバージョンの Windows では、64 ビットアプリケーション用と 32 ビットアプリケーション用の 2 つのシステムデータソースが Windows レジストリに管理されています。32 ビットアプリケーションにアクセスできるシステムデータソースを作成するには、32 ビット dbdsn ユーティリティ (SQL Anywhere インストールディレクトリの bin32 サブディレクトリにある) を実行する必要があります。接続の問題を回避するため、64 ビットと 32 ビットの両方のシステムデータソースを作成し、まったく同じ設定にしておくことをお勧めします。

##### 64 ビット Windows でのユーザデータソースの作成

ユーザデータソース定義は、Windows レジストリの一部として保存され、システムに現在ログインしているユーザ用の設定を含んでいます。64 ビットバージョンの Windows では、コンピュータの各ユーザごとに 1 つのユーザデータソースが Windows レジストリに管理されています。ユーザデータソースを作成するには、dbdsn ユーティリティの 32 ビットと 64 ビットのどちらのバージョンも使用できます。このデータソースは、一致する ODBC ドライバがインストールされている限り、64 ビットのクライアントアプリケーションでも 32 ビットのクライアントアプリケーションでも使用できます。

#### 警告

ユーザ ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはお勧めしません。

#### 手順

コマンドプロンプトからのデータソースユーティリティ (dbdsn) の実行

## 結果

ODBC データソースが作成されます。

### 例

次のコマンドはサンプルデータベースの ODBC データソースを作成します。コマンドは、1 行に入力する必要があります。

```
dbdsn -w "My DSN" -c "DBF=${SQLANY17}/demo.db"
```

## 関連情報

[Windows でのファイルデータソース \[119 ページ\]](#)

[データソースユーティリティ \(dbdsn\) \[1089 ページ\]](#)

### 1.1.4.4 Mac OS X での ODBC データソースの作成

ODBC アドミニストレータツールを使用して ODBC データソースを作成します。

#### 前提条件

バージョン 10.6 以降の Mac OS X システムを使用することはできません。

SQL Anywhere ODBC ドライバをインストールする必要があります。

Mac OS X 用の ODBC アドミニストレータツールをインストールする必要があります。このツールは Apple 社より入手できません。

SQL Anywhere ODBC ドライバを使用するよう、Mac OS X 用 ODBC アドミニストレータツールを設定しておく必要があります。このように設定しておくことで、このドライバを使用するデータソースを作成できるようになります。

#### コンテキスト

ODBC アドミニストレータツールを使用するには、環境変数 ODBCINI または ODBC\_INI のいずれかを使用し、ODBC 初期化ファイル (~/.Library/ODBC/odbc.ini) の標準以外の場所やファイル名を指定する必要があります。

この問題を回避するため、データソースユーティリティ (dbdsn) を使用して、Mac OS X に ODBC データソースを作成することも可能です。

## 手順

1. /Applications/Utilities から Mac OS X 用の ODBC アドミニストレータツールを起動します。
2. ユーザ DSN タブをクリックしてから、追加をクリックします。
3. 名前リストで SQL Anywhere17 をクリックします。
4. OK をクリックします。
5. データソース名フィールドに、使用するデータソースの名前を入力します。
6. 説明フィールドに、使用するデータソースに適した説明を入力します。
7. 有効な接続パラメータと値を指定します。一部の値は大文字と小文字が区別されます。次はその例です。

キーワード	値
DatabaseFile	/Applications/SQLAnywhere17/samples/demo.db
StartLine	dbeng17
ServerName	demo17
ThreadManager	ON

### 警告

ユーザ ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはお勧めしません。

8. OK をクリックします。
9. 適用をクリックします。
10. Command + Q を押して終了します。

## 結果

ODBC データソースが作成されます。

このセクションの内容:

### [Mac OS X 用の ODBC アドミニストレータツールの設定 \[118 ページ\]](#)

Mac OS X 用の ODBC アドミニストレータツールに SQL Anywhere ODBC ドライバを追加し、このドライバを使用する ODBC データソースを作成できるようにします。

## 関連情報

[UNIX での ODBC データソース \[121 ページ\]](#)

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[データソースユーティリティ \(dbdsn\) \[1089 ページ\]](#)

## 1.1.4.4.1 Mac OS X 用の ODBC アドミニストレータツールの設定

Mac OS X 用の ODBC アドミニストレータツールに SQL Anywhere ODBC ドライバを追加し、このドライバを使用する ODBC データソースを作成できるようにします。

### 前提条件

バージョン 10.6 以降の Mac OS X システムを使用することはできません。

SQL Anywhere ODBC ドライバをインストールする必要があります。

Mac OS X 用の ODBC アドミニストレータツールをインストールする必要があります。このツールは Apple 社より入手できません。

### コンテキスト

データソースユーティリティ (dbdsn) を使用して、Mac OS X に ODBC データソースを作成することも可能です。

### 手順

1. /Applications/Utilities から Mac OS X 用の ODBC アドミニストレータツールを起動します。
2. **ドライバタブ**をクリックします。
3. **追加**をクリックします。
4. *Description* フィールドに **SQL Anywhere17** と入力します。
5. **Choose** をクリックし、*Driver file* と *Setup file* の両方のフィールドで SQL Anywhere ODBC ドライバを選択します。デフォルトでは、/Applications/SQLAnywhere17/System/lib64/dbodbc17\_r.bundle. にあります。

バンドル名の *\_r* は、ドライバのスレッドバージョンであることを示します。非スレッドアプリケーションで使用するための非スレッドバージョン (dbodbc17.bundle) もあります。

6. **OK** をクリックします。

ODBC 設定ファイルは、ホームディレクトリの /Library/ODBC にあります。ドライバ情報として *odbcinst.ini* ファイル、データソース情報として *odbc.ini* ファイルがあります。情報はテキストエディタを使用して編集できます。

ODBC 初期化ファイル (~ /Library/ODBC/odbc.ini) の標準以外の場所やファイル名を指定するには、環境変数 ODBCINI または ODBC\_INI のいずれかを使用する必要があります。

### 結果

SQL Anywhere ODBC ドライバが追加されます。

## 関連情報

[ODBC データソースの作成 \(dbdsn ユーティリティを使用する場合\) \[115 ページ\]](#)

### 1.1.4.5 Windows でのファイルデータソース

ファイルデータソースは、ファイルとして保存されるもので、システムレジストリに保存される ODBC データソースの代わりになります。

通常、Windows のファイルデータソースの拡張子は .dsn です。ファイルデータソースは複数のセクションから構成されていて、各セクションは角カッコで囲まれた名前で始まります。

ファイルデータソースを使用して接続するには、FileDataSourceName (FILEDSN) 接続パラメータを使います。1 つの接続文字列内で DataSourceName (DSN) と FileDataSourceName (FILEDSN) の両方は使用できません。

FileDSN の作成や変更を行う場合には、先にデータベースサーバが動作し、データベースが起動されている必要があります。接続は必ず Microsoft ODBC データソースアドミニストレータによって行われます。接続がうまくいかなかった場合、次のいずれかの結果となります。

#### 新しいファイルデータソースの場合

Microsoft ODBC データソースアドミニストレータは、指定のファイルデータソースパラメータを使って接続を確立することができなかったというメッセージを表示します。その後、このメッセージから、検証されていないファイルを保存するかどうか尋ねられます。ファイルデータソースを保存すると選択した場合は、ODBC データソースアドミニストレータによって次の行のみが書き込まれます。

```
[ODBC]
DRIVER=SQL Anywhere 17
```

#### 既存のファイルデータソースの場合

Microsoft ODBC データソースアドミニストレータは、指定のファイルデータソース名が不正であることを示すメッセージを表示します。ファイルデータソースは、ODBC データソースアドミニストレータによっては更新されません。

接続に成功すると、ODBC データソースアドミニストレータによって新規または更新されたファイルデータソースがディスクに書き込まれますが、Password (PWD) 接続パラメータは含まれません (ファイルに以前に存在した場合は削除されます)。EncodedPassword (ENP) 接続パラメータの場合はこの限りではありません。保存が行われます。

#### **i** 注記

ファイルデータソースを使用すると、ファイルをユーザに配布し、複数のユーザ接続の管理を簡単にすることができます。ファイルがファイルデータソースのデフォルト ロケーションにある場合、ODBC によって自動的に選択されます。

## 1.1.4.6 ODBC ファイルデータソースの作成 (ODBC データソースアドミニストレータの場合)

ODBC データソースアドミニストレータで、ODBC ファイルデータソースを作成します。

### 前提条件

ファイルデータソースを正常に作成するには、ファイルデータソースの作成対象となるデータベースに対して接続を確立できる必要があります。

### コンテキスト

#### 警告

ユーザ ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはお奨めしません。

### 手順

1. **スタート** > **プログラム** > **SQL Anywhere17** > **管理ツール** > **ODBC データソースアドミニストレータ** をクリックします。
2. **ファイル DSN** タブをクリックします。
3. **追加** をクリックします。
4. **ドライバリスト** で **SQL Anywhere17** をクリックします。
5. **次へ** をクリックします。
6. **データソースの新規作成ウィザード** の指示に従います。
7. **OK** をクリックします。
8. **OK** をクリックして、**ODBC データソースアドミニストレータ** を閉じます。

### 結果

ODBC データソースファイルが作成されます。

## 1.1.4.7 UNIX での ODBC データソース

UNIX オペレーティングシステムでは、ODBC データソースはシステム情報ファイルに格納されます。

このファイルは通常 `.odbc.ini` という名前です。データベースサーバは、次のロケーションで順番にシステム情報ファイルを検索します。

- ODBCINI 環境変数
- ODBC\_INI 環境変数
- ODBCHOME 環境変数
- HOME 環境変数
- ユーザのホームディレクトリ (~)
- PATH 環境変数

### i 注記

ODBCINI と ODBC\_INI 環境変数は、システム情報ファイル (ファイル名は通常 `.odbc.ini`) の検索に使用されますが、ODBCHOME と HOME 環境変数は `.odbc.ini` ファイルがあるパスの定義に使用されます。

ODBCINI と ODBC\_INI はどちらも、ファイル名を含むフルパスで指定します。システム情報ファイルが ODBCINI または ODBC\_INI で指定されたディレクトリにある場合、ファイル名は `.odbc.ini` である必要はありません。

次にシステム情報ファイルの例を示します。

```
[My Data Source]
Host=hostname
ServerName=myserver
```

システム情報ファイルには、任意の接続パラメータを入力できます。

### ⚠ 警告

ユーザ ID、パスワード (暗号化の有無は不問)、データベースキーをデータソースに保存することはお奨めしません。

UNIX では、`dbdsn` ユーティリティを使用して ODBC データソースを作成、管理します。

### ⚠ 警告

UNIX では、SQL Anywhere データソースだけを使用している場合を除き、ファイル非表示ユーティリティ (`dbfhide`) を使用して、システム情報ファイル (デフォルトのファイル名は `.odbc.ini`) を暗号化しないでください。その他のデータソースを使用する予定がある場合は、システム情報ファイルの内容を暗号化すると、その他のドライバが正しく機能しなくなる場合があります。

また、ODBC 初期化ファイル (デフォルトのファイル名は `.odbc.ini`) には機密情報が含まれている場合があります。ファイルへのパーミッションはできるだけ制限を厳しくしてください。たとえば、ファイルに対して `chmod 600 .odbc.ini` を実行することを検討してください。

## 関連情報

- [接続パラメータのアルファベット順リスト \[44 ページ\]](#)
- [データソースユーティリティ \(dbdsn\) \[1089 ページ\]](#)
- [ODBCHOME 環境変数 \[UNIX\] \[547 ページ\]](#)
- [ODBCINI と ODBC\\_INI 環境変数 \[UNIX\] \[547 ページ\]](#)

## 1.1.5 認証メカニズム

LDAP、PAM、Kerberos、Microsoft Windows 統合化ログインなどの外部認証メカニズムを使用してデータベースに接続できるように設定することができます。

標準的なデフォルトのデータベース接続方法は、データベースに認識されているユーザ ID とパスワードを入力するという方法です。

標準的な認証の他に、データベースサーバは次の認証メカニズムをサポートしています。

- Microsoft Windows 統合化ログイン
- LDAP ユーザ認証
- PAM ユーザ認証
- Kerberos ユーザ認証

これらの認証メカニズムは、1つのセキュリティシステムで、データベース、オペレーティングシステム、ネットワークのセキュリティに対応できるため、ユーザにとっても管理者にとっても便利です。他にも、次のような利点があります。

- ユーザはデータベースに接続するときにデータベースユーザ ID やパスワードを入力しなくてよい
- 認証ユーザはデータベースユーザ ID にマッピングされるので、ユーザ ID とパスワードがデータベースユーザ ID とパスワードに一致していなくてもよい
- 複数のユーザを1つのデータベースユーザ ID にマッピングできる
- グループ全体を1つのデータベースユーザ ID にマッピングできる認証メカニズムもある

これらの認証メカニズムには単一のセキュリティシステムという利便性がある一方、データベース管理者が知っておくべき重要なセキュリティ上の問題があります。無制限データベースアクセスや、データベースファイルのコピーなどです。これらの問題点や対処方法を確実に把握しておく必要があります。

このセクションの内容:

### [ログインモードと login\\_mode オプション \[123 ページ\]](#)

login\_mode オプションは、利用を許可するユーザ認証メカニズムを指定するために使用します。

### [Windows 統合化ログイン \[124 ページ\]](#)

Windows 統合化ログイン機能を使用すると、オペレーティングシステム、ネットワークのログイン、データベース接続を、単一のユーザ ID とパスワードで管理できます。

### [LDAP ユーザ認証 \[133 ページ\]](#)

LDAP (Lightweight Directory Access Protocol) は、ディレクトリサービスへの業界標準のアクセス方法です。

### [チュートリアル: LDAP ユーザ認証環境の作成 \[139 ページ\]](#)

このチュートリアルでは、Interactive SQL または SQL Central で LDAP ユーザ認証環境をセットアップする基本的な手順について説明します。

### [PAM ユーザ認証 \[146 ページ\]](#)

PAM (Pluggable Authentication Module) は、複数の低レベル認証スキームに共通のインタフェースを提供する認証メカニズムであり、ユーザはすべてのシステムで同じクレデンシャルを使用できます。

### [Kerberos ユーザ認証 \[149 ページ\]](#)

Kerberos ログイン機能を使用すると、データベース接続、オペレーティングシステム、ネットワークのログインを、単一のユーザ ID とパスワードで管理できます。

### [セキュリティ: ログインモードを使用したデータベースのセキュリティ保護 \[162 ページ\]](#)

データベースをセキュリティ保護するための方法はいくつかあります。

### [セキュリティ: 統合化ログインを使用すると、無制限データベースアクセスが行われる可能性があります \[163 ページ\]](#)

統合化ログイン機能は、ネイティブな Windows 認証を利用してデータベースへのアクセスを制御する機能です。

## 1.1.5.1 ログインモードと `login_mode` オプション

`login_mode` オプションは、利用を許可するユーザ認証メカニズムを指定するために使用します。

データベースオプションは、それが指定されているデータベースにしか適用されないため、同じサーバ内にロードされて動作している場合でも、データベースが異なればログイン設定も異なります。

`login_mode` データベースオプションに指定できる値は次のとおりです。

#### **Standard**

標準ユーザ認証を許可します。これはデフォルト設定です。ユーザ ID とパスワードを入力する必要があります。

#### **Integrated**

Windows 統合化ログインを許可します。

#### **Kerberos**

Kerberos ログインを許可します。

#### **LDAPUA**

LDAP (Lightweight Directory Access Protocol) ユーザ認証ログインを許可します。

#### **PAMUA**

PAM (Pluggable Authentication Modules) ユーザ認証ログインを許可します。

#### **CloudAdmin**

このログインモードは、クラウドの内部で使用します。

#### **警告**

標準ユーザ認証 (ユーザ ID とパスワード) を許可しないように `login_mode` データベースオプションを設定すると、接続できるのは、統合化、Kerberos、LDAPUA、PAMUA、CloudAdmin のいずれかのログインマッピングを付与されているユーザまたはグループのみに制限されます。MANAGE ANY USER 権限を持つユーザでないかぎり、ユーザ ID とパスワードを使用して接続しようとすると、エラーが発生します。

複数のログインタイプを許可するには、`login_mode` オプションに複数の値を指定します。たとえば、次の SQL 文は、標準ログインと統合化ログインの両方の接続を許可するように `login_mode` データベースオプションの値を設定します。

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

データベースファイルのコピーが可能な場合、一時的なパブリック login\_mode オプション (標準以外のログインモード) を使用してください。これにより、ファイルのコピーが可能な場合に標準以外のログインモードがデフォルトでサポートされません。

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

## 関連情報

[Kerberos ユーザ認証 \[149 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

## 1.1.5.2 Windows 統合化ログイン

Windows 統合化ログイン機能を使用すると、オペレーティングシステム、ネットワークのログイン、データベース接続を、単一のユーザ ID とパスワードで管理できます。

統合化ログインを作成するには、次の手順に従います。

- 統合化ログイン機能を有効にします。
- 統合化ログインのマッピング先となるデータベースユーザを作成します (存在しない場合)。
- Windows ユーザまたはグループプロファイルと既存のデータベースユーザの間に、統合化ログインマッピングを作成します。SQL Central の [ログインマッピング](#) フォルダに、統合化ログイン権限を持つユーザがすべて表示されます。
- クライアントアプリケーションから接続し、統合化ログイン機能をテストします。

## サポートされるオペレーティングシステム

統合化ログイン機能は、Windows ベースのデータベースサーバで使用できます。Windows クライアントは、統合化ログインを使用して、Windows で実行されているネットワークサーバに接続できます。

## 統合化ログインの利点

統合化ログインは、1 つまたは複数の Windows ユーザプロファイルまたは Windows ユーザグループプロファイルから既存のデータベースユーザへのマッピングです。ユーザプロファイルまたはグループのセキュリティを特定し、コンピュータへのログインに成功したユーザは、他のユーザ ID またはパスワードを指定しないでデータベースに接続できます。

そのためには、統合化ログインを使用するようにデータベースを設定し、コンピュータまたはネットワークへのログインに使用するユーザまたはグループのプロファイルとデータベースユーザの間のマッピングを許可します。

統合化ログインの使用は、ユーザにとって便利であると同時に、1 つのセキュリティシステムでデータベースとネットワークの両方のセキュリティを維持できます。統合化ログインには次の利点があります。

- ユーザによるユーザ ID とパスワードの入力は不要です。

- ユーザはオペレーティングシステムによって認証されます。データベースのセキュリティと、コンピュータやネットワークのセキュリティには、単一のシステムが使用されます。
- 複数のユーザまたはグループプロファイルを 1 つのデータベースユーザ ID にマッピングできます。
- Windows コンピュータへのログインに使用する名前とパスワードは、データベースユーザの ID とパスワードと一致している必要はありません。

#### 警告

統合化ログインには単一のセキュリティシステムという利便性がある一方、データベース管理者が知っておくべき重要なセキュリティ上の問題があります。無制限データベースアクセスや、データベースファイルのコピーなどです。これらの問題点や対処方法を確実に把握しておく必要があります。

このセクションの内容:

#### [Windows 統合化ログインマッピングの作成 \(SQL Central の場合\) \[126 ページ\]](#)

Windows 統合化ログインを使用すると、オペレーティングシステムとネットワークのログイン、さらにデータベースへの接続を、単一のユーザ ID とパスワードで管理できます。

#### [統合化ログインの作成 \(SQL の場合\) \[127 ページ\]](#)

Windows 統合化ログインを使用すると、オペレーティングシステムとネットワークのログイン、さらにデータベースへの接続を、単一のユーザ ID とパスワードで管理できます。

#### [統合化ログイン権限の取り消し \(SQL Central\) \[128 ページ\]](#)

SQL Central を使用して、統合化ログイン権限を取り消します。

#### [統合化ログイン権限の取り消し \(SQL\) \[129 ページ\]](#)

REVOKE 文を使用して、統合化ログイン権限を取り消します。

#### [クライアントアプリケーションからの統合化ログイン接続 \[129 ページ\]](#)

統合化ログインを使用して接続するには、次の 2 通りの方法があります。

#### [Windows ユーザグループの統合化ログイン \[130 ページ\]](#)

Windows ユーザグループへのログインを統合できます。

#### [Windows ユーザグループのメンバーによるデータベースへの接続を防ぐ方法 \[132 ページ\]](#)

統合化ログインを持つ Windows ユーザグループのメンバーであるユーザが、グループの統合化ログインを使用してデータベースに接続できないようにするには、2 つの方法があります。

#### [ネットワークから見た統合化ログイン \[133 ページ\]](#)

データベースがネットワークサーバにある場合、統合化ログインを使用するには次の 2 つの条件のどちらかが満たされていなければなりません。

## 関連情報

[セキュリティ: ログインモードを使用したデータベースのセキュリティ保護 \[162 ページ\]](#)

## 1.1.5.2.1 Windows 統合化ログインマッピングの作成 (SQL Central の場合)

Windows 統合化ログインを使用すると、オペレーティングシステムとネットワークのログイン、さらにデータベースへの接続を、単一のユーザ ID とパスワードで管理できます。

### 前提条件

MANAGE ANY USER システム権限が必要です。また、SELECT ANY TABLE システム権限か、SYSLOGINMAP システムビューに対する SELECT 権限のいずれかが必要です。

ログインモードの変更を行う場合、SET ANY SECURITY OPTION システム権限も必要です。

### コンテキスト

ユーザまたはグループのプロファイルとデータベースユーザ ID を 1 対 1 の対応にする必要はありません。同じデータベースユーザ ID に対して複数のユーザプロファイルをマップできます。

ユーザプロファイルは、既存のデータベースユーザ ID にのみマップできます。データベースユーザ ID がデータベースから削除されると、そのデータベースユーザ ID に基づくすべての統合化ログインマッピングが自動的に削除されます。

### 手順

1. *SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. フォルダビューで、ログインマッピングを右クリックし、**新規** > **ログインマッピング** をクリックします。
3. 統合化ログインマッピングを選択し、次へをクリックします。
4. データベースに接続する *Windows ユーザ* を指定してくださいというフィールドに、統合化ログインを作成するユーザまたはグループのプロファイル名を入力します。
5. ログインモードを指定するよう求められたら、標準および統合化を選択します。
6. *Windows ユーザ* に関連付けるデータベースユーザを指定してくださいというリストで、このユーザをマッピングするデータベースユーザ ID を選択します。
7. ログインマッピング作成ウィザードの残りの指示に従います。

### 結果

統合化ログインマッピングが作成されます。

## 1.1.5.2.2 統合化ログインの作成 (SQL の場合)

Windows 統合化ログインを使用すると、オペレーティングシステムとネットワークのログイン、さらにデータベースへの接続を、単一のユーザ ID とパスワードで管理できます。

### 前提条件

MANAGE ANY USER システム権限が必要です。また、SELECT ANY TABLE システム権限か、SYSLOGINMAP システムビューに対する SELECT 権限のいずれかが必要です。

ログインモードの変更を行う場合、SET ANY SECURITY OPTION システム権限も必要です。

### コンテキスト

ユーザまたはグループのプロファイルとデータベースユーザ ID を 1 対 1 の対応にする必要はありません。同じデータベースユーザ ID に対して複数のユーザプロファイルをマップできます。

### 手順

1. データベースに接続します。
2. SET OPTION 文を実行してログインモードを変更し、Windows 統合化ユーザ認証を有効にします。

たとえば、次の文は標準ユーザ認証と統合化ユーザ認証の両方を有効にします。

```
SET OPTION PUBLIC.login_mode='Standard,Integrated';
```

さらにセキュリティを強化するには、SET OPTION 文の代わりに SET TEMPORARY OPTION を使用して login\_mode オプションを設定します。

```
SET TEMPORARY OPTION PUBLIC.login_mode='Standard,Integrated';
```

login\_mode オプションの現在の設定は次のようにして照会できます。

```
SELECT connection_property('login_mode');
```

3. GRANT INTEGRATED LOGIN TO 文を実行します。

### 結果

統合化ログインが作成されます。

## 例

次の SQL 文を使用すると、Window ユーザの `fran_whitney` と `matthew_cobb` は、DBA ユーザ ID またはパスワードを知らなかったり指定しなかったりしても、DBA ユーザとしてデータベースにログインできます。

```
GRANT INTEGRATED LOGIN
TO fran_whitney, matthew_cobb
AS USER DBA;
```

次の SQL 文を使用すると、Windows グループの `mywindowsusers` のメンバーである Windows ユーザは、DBA ユーザのユーザ ID またはパスワードを知らない、あるいは指定しなかった場合でも、DBA ユーザとしてデータベースにログインできます。

```
GRANT INTEGRATED LOGIN
TO mywindowsusers
AS USER DBA;
```

## 関連情報

[Windows ユーザグループの統合化ログイン \[130 ページ\]](#)

### 1.1.5.2.3 統合化ログイン権限の取り消し (SQL Central)

SQL Central を使用して、統合化ログイン権限を取り消します。

## 前提条件

MANAGE ANY USER システム権限が必要です。また、SELECT ANY TABLE システム権限か、SYSLOGINMAP システムビューに対する SELECT 権限のいずれかが必要です。

## 手順

1. [SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[ログインマッピング](#)をクリックします。
3. 右ウィンドウ枠で、削除するログインマッピングを選択し、[削除](#)をクリックします。
4. [はい](#)をクリックします。

## 結果

統合化ログイン権限が取り消されます。

### 1.1.5.2.4 統合化ログイン権限の取り消し (SQL)

REVOKE 文を使用して、統合化ログイン権限を取り消します。

## 前提条件

MANAGE ANY USER システム権限が必要です。

## 手順

1. データベースに接続します。
2. REVOKE INTEGRATED LOGIN FROM 文を実行します。

## 結果

統合化ログイン権限が取り消されます。

### 例

次の SQL 文では、pchin という Windows ユーザから統合化ログイン権限を削除します。

```
REVOKE INTEGRATED LOGIN  
FROM pchin;
```

### 1.1.5.2.5 クライアントアプリケーションからの統合化ログイン接続

統合化ログインを使用して接続するには、次の 2 通りの方法があります。

接続パラメータリストで **Integrated (INT)** パラメータを **YES** に設定する

接続文字列で **Integrated (INT)** パラメータが **YES** に設定されている場合、統合化ログインが試行されます。接続が失敗し、login\_mode データベースオプションが **Standard** に設定されている場合、サーバは標準ログインを試行します。

接続文字列でも、接続ウィンドウでも、ユーザ ID とパスワードを指定しない

ユーザ ID やパスワードを指定しないでデータベースに接続しようとすると、統合化ログインが試行されます。ログインの成否は、現在のユーザプロフィール名がデータベース内の統合化ログインマッピングに一致するかどうかによって決まります。

## Interactive SQL の例

次の例では、デフォルトのデータベースサーバの統合化ログインマッピングと一致するユーザプロフィールでユーザがログインした場合に接続に成功します。

```
CONNECT USING 'INTEGRATED=yes';
```

この CONNECT 文でデータベースに接続できるのは、次の場合です。

- サーバが現在実行中です。
- デフォルトデータベースで、統合化ログイン接続を受け入れるよう login\_mode データベースオプションが設定されています。
- 現在のユーザのユーザプロフィール名と一致する統合化ログインマッピングまたはユーザが所属する Windows ユーザグループの統合化ログインマッピングが作成されています。
- 追加接続情報のプロンプトが表示されたときにユーザが追加情報を入力しないで OK をクリックします。

## 関連情報

[login\\_mode オプション \[717 ページ\]](#)

### 1.1.5.2.6 Windows ユーザグループの統合化ログイン

Windows ユーザグループへのログインを統合できます。

Windows ユーザがログインするとき、明示的な統合化ログインマッピングを持っていないが、統合化ログインマッピングがある Windows ユーザグループに所属している場合、そのユーザは、Windows ユーザグループの統合化ログインマッピングで指定されたデータベースユーザまたはグループとしてデータベースに接続します。

#### 警告

Windows ユーザグループの統合化ログインを作成すると、そのグループに属するすべてのユーザが、ユーザ ID やパスワードを知らなくてもデータベースに接続できます。

## 複数グループのメンバー

Windows ユーザが複数の Windows ユーザグループに属し、コンピュータ上の複数の Windows ユーザグループの統合化ログインマッピングがデータベースに存在する場合、統合化ログインが成功するのは、コンピュータ上の Windows ユーザグ

ループのすべてが同じデータベースユーザ ID に対する統合化ログインマッピングを持っている場合のみです。複数の Windows ユーザグループが異なるデータベースユーザ ID への統合化ログインマッピングを持つ場合は、エラーが返され、統合化ログインは失敗します。

たとえば、dbuserA と dbuserB という 2 つのユーザ ID があるデータベースと、Windows ユーザグループ xpgroupA と xpgroupB に属する Windows ユーザ windowsuser について考えてみます。

SQL 文	許可内容
GRANT INTEGRATED LOGIN TO windowsuser AS USER dbuserA;	windowsuser に対して明示的に設定された統合化ログインマッピングを使用して windowsuser はデータベースに接続します。
GRANT INTEGRATED LOGIN TO xpgroupA AS USER dbuserB;	windowsuser は、xpgroupA に対して付与された統合化ログインマッピングを使用してデータベースに接続します。
GRANT INTEGRATED LOGIN TO xpgroupA AS USER dbuserB; GRANT INTEGRATED LOGIN TO xpgroupB AS USER dbuserB;	windowsuser が属する両方の Windows ユーザグループに同じデータベースユーザへの統合化ログインマッピングがあるため、windowsuser はデータベースに接続できます。
GRANT INTEGRATED LOGIN TO xpgroupA AS USER dbuserA; GRANT INTEGRATED LOGIN TO xpgroupB AS USER dbuserB;	データベースへは接続できません。windowsuser がデータベースに接続しようとした場合、各 Windows ユーザグループには異なるデータベースユーザへの統合化ログインマッピングがあり、windowsuser は両方の Windows ユーザグループのメンバーであるため、統合化ログインは失敗します。

## ドメインコントローラーのロケーション

デフォルトでは、Windows ユーザグループのメンバーシップを確認するために、データベースサーバを実行しているコンピュータが使用されます。ドメインコントローラサーバが、データベースサーバとは異なるコンピュータで実行されている場合は、integrated\_server\_name オプションを使用して、ドメインコントローラサーバの名前を指定できます。次に例を示します。

```
SET OPTION PUBLIC.integrated_server_name = '¥¥¥¥¥myserver-1';
```

## 関連情報

[Windows ユーザグループのメンバーによるデータベースへの接続を防ぐ方法 \[132 ページ\]](#)

[integrated\\_server\\_name オプション \[710 ページ\]](#)

## 1.1.5.2.7 Windows ユーザグループのメンバーによるデータベースへの接続を防ぐ方法

統合化ログインを持つ Windows ユーザグループのメンバーであるユーザが、グループの統合化ログインを使用してデータベースに接続できないようにするには、2つの方法があります。

- パスワードのないデータベースユーザ ID に対して、ユーザの統合化ログインを作成します。
- ユーザがログインを許可されているかどうかを確認し、無許可のユーザが接続しようすると例外を発行するストアードプロシージャを作成します。このストアードプロシージャは、login\_procedure オプションによって呼び出します。

### パスワードなしでユーザ ID への統合化ログインを作成する

ユーザが統合化ログインを持つ Windows ユーザグループのメンバーであるだけでなく、ユーザ ID に対する明示的な統合化ログインも持っている場合、データベースへの接続にはそのユーザの統合化ログインが使用されます。ユーザが Windows ユーザグループ統合化ログインを使用してデータベースへ接続できないようにするには、データベースユーザ ID への Windows ユーザの統合化ログインをパスワードなしで作成します。パスワードを持たないデータベースユーザ ID は、データベースに接続できません。

パスワードを持たないユーザ ID への統合化ログインは SQL で作成できます。

1. データベースにパスワードなしでユーザを追加します。次に例を示します。

```
CREATE USER db_user_no_password;
```

2. パスワードを持たないデータベースユーザにマッピングする Windows ユーザの統合化ログインを作成します。次に例を示します。

```
GRANT INTEGRATED LOGIN TO WindowsUser  
AS USER db_user_no_password;
```

### Windows ユーザの接続を防ぐ手順の作成

login\_procedure オプションは、データベースへの接続が試行されるたびに呼び出すストアードプロシージャを指定します。デフォルトでは dbo.sp\_login\_environment プロシージャが呼び出されます。login\_procedure オプションを設定して、特定のユーザがデータベースに接続できないようにするために作成したプロシージャを呼び出すことができます。

次の例では、login\_procedure オプションによって呼び出される login\_check というプロシージャを作成します。login\_check プロシージャは、データベースに接続できないユーザのリストに照らし合わせて、指定された OS ユーザ名を確認します。指定されたユーザ名がリストに見つかったら、接続は失敗します。この例では、Joe、Harry、または Martha というユーザは接続を許可されていません。ユーザがリストに見つからない場合、データベース接続は通常どおり実行され、sp\_login\_environment プロシージャが呼び出されます。

```
CREATE PROCEDURE DBA.login_check()  
BEGIN  
    DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';  
    // Disallow certain users  
    IF( connection_property( 'OSUser' ) IN ( 'Joe', 'Harry', 'Martha' ) ) THEN
```

```

        SIGNAL INVALID_LOGON;
    ELSE
        CALL sp_login_environment;
    END IF;
END
go
GRANT EXECUTE ON DBA.login_check TO PUBLIC
go
SET OPTION PUBLIC.login_procedure='DBA.login_check'
go

```

## 1.1.5.2.8 ネットワークから見た統合化ログイン

データベースがネットワークサーバにある場合、統合化ログインを使用するには次の 2 つの条件のどちらかが満たされていなければなりません。

- 統合化ログイン接続に使用するユーザプロファイルが、ローカルコンピュータとサーバで同じである必要があります。両方のユーザプロファイルのパスワードも同じである必要があります。  
たとえば、ユーザ DSmith が統合化ログインを使用してネットワークサーバ上にロードされているデータベースに接続しようとした場合、ローカルコンピュータとデータベースサーバを実行しているコンピュータとの両方に、同じユーザプロファイル名とパスワードが存在している必要があります。ユーザ DSmith は両方のコンピュータにログオンできる必要があります。
- ネットワークアクセスが Microsoft ドメインコントローラによって制御されている場合、統合化ログインを試みるユーザは、ドメインコントローラサーバのドメインパーミッションを持っていて、ネットワークにログインしている必要があります。ローカルコンピュータのユーザプロファイルとネットワークサーバのユーザプロファイルが一致する必要はありません。

## 1.1.5.3 LDAP ユーザ認証

LDAP (Lightweight Directory Access Protocol) は、ディレクトリサービスへの業界標準のアクセス方法です。

LDAP ユーザ認証では、クライアントアプリケーションがユーザ ID とパスワード情報をデータベースサーバに送信して、カタログを利用しないで LDAP サーバによって認証を行うことができます。LDAP サーバを使用した認証によって、組織規模のパスワード管理が可能になります。

LDAP ユーザ認証は、既存のコンピューティング環境でユーザ管理を簡素化して集中管理したい組織や、新しいコンピューティング環境で異なるアプリケーションやデータベースのユーザを管理する際の無用な複雑さを回避したいユーザに最適です。

login\_mode データベースオプションには LDAPUA モードが含まれています。これによって、認証メカニズムの候補が拡張され、LDAP ユーザ認証が含まれるようになります。SQL Anywhere LDAP ユーザ認証サポートにより、SQL Anywhere を、LDAP に基づいた既存の全社的ディレクトリアクセスフレームワークに統合することができます。

### LDAP ユーザ認証の仕組み

各ユーザはログインポリシーに関連付けられます。ログインポリシーは、必要に応じてプライマリ LDAP サーバオブジェクト、セカンダリ LDAP サーバオブジェクト、またはそれら両方をユーザ認証のために参照できます。異なるオプションを使用して複数のログインポリシーを定義することにより、複数のドメインを指定できます。

LDAP サーバオブジェクトを指定するログインポリシーを備えたユーザを認証するには、そのユーザの識別名 (DN) と、DN に関連付けられたパスワードを使用して、信頼できる LDAP ディレクトリサーバに接続します。

クライアントアプリケーションは、通常はその LDAP DN を認識しないか、データベースサーバに送信しません。その代わりに、データベースのユーザ ID とパスワードを認識してデータベースサーバに送信します。データベースのユーザ ID とその DN の関連付けは、定義済みの検索文字列を使用して、LDAP サーバ上の特定のユーザ ID の DN を検索することにより行われます。LDAP サーバでの検索からデータベースサーバに返された DN は、ISYSUSER システムテーブル (user\_dn カラム) に、現在の UTC 時刻 (user\_dn\_cached\_at カラム) とともに格納されます。ALTER LOGIN POLICY...LDAP\_REFRESH\_DN=NOW 文を使用して関連するログインポリシーが更新されるまで、格納された DN 値が継続して使用されます。

## LDAP サーバ設定オブジェクト

LDAP サーバに接続するための設定情報は、LDAP サーバ設定オブジェクトに格納されます。ISYSLDAPSERVER システムテーブルの各ローは、LDAP サーバ設定オブジェクトの設定を反映します。それぞれの設定オブジェクトには、特定のユーザ ID の DN を取得する際に使用する LDAP 検索文字列など、接続のための一連の属性が含まれます。これらの属性や LDAP サーバのステータスは、SYSLDAPSERVER ビューを照会して調べることができます。

## 高可用性と LDAP ユーザ認証

ユーザのログインポリシーを介して、特定ユーザのプライマリサーバまたはセカンダリサーバとして LDAP サーバを指定することにより、LDAP サーバの高可用性が実現されます。何らかの理由 (LDAP サーバのメンテナンス、ネットワークの障害、LDAP サーバのクラッシュなど) によってプライマリ LDAP サーバにアクセスできない場合は、セカンダリ LDAP サーバがユーザ認証に使用されます。プライマリからセカンダリにフェイルオーバーした後にセカンダリからプライマリにフェールバックする機能は、SQL 文を使用して手動で管理したり、データベースサーバによって切り替えが適切であると判断されたときに自動的に実行することができます。

## LDAP ログインポリシーオプションの継承

デフォルトのルートログインポリシーからの継承は、LDAP ポリシーオプションに関してグループで行われます。たとえば、ルートログインポリシーでプライマリ LDAP サーバを定義し、LDAP ポリシーオプションを指定しないログインポリシー policy\_X が定義された場合、policy\_X に関連付けられたユーザは、ルートポリシーからプライマリ LDAP サーバと LDAP ポリシーオプションを継承します。

別のログインポリシー policy\_Y がセカンダリ LDAP サーバのみを定義している場合、policy\_Y に関連付けられたユーザは、そのポリシーからセカンダリ LDAP サーバとその関連する設定のみを使用し、ルートポリシーからの LDAPUA ポリシーオプションは何も継承されません。

このセクションの内容:

[LDAP サーバ設定オブジェクトのアクティブ化または中断 \(SQL Central の場合\) \[135 ページ\]](#)

LDAP サーバで使用される LDAP サーバ設定オブジェクトをアクティブにしたり、中断したりします。

[LDAP サーバ設定オブジェクトの変更 \(SQL Central の場合\) \[136 ページ\]](#)

LDAP ユーザ認証用の LDAP サーバに接続するための LDAP サーバ設定オブジェクトを変更します。

[LDAP サーバ設定オブジェクトの検証 \(SQL Central の場合\) \[137 ページ\]](#)

LDAP サーバに接続して、LDAP 設定オブジェクトが正しく設定されていることを確認します。

[LDAP サーバ設定オブジェクトの削除 \(SQL Central の場合\) \[138 ページ\]](#)

LDAP サーバへの接続に使用される LDAP サーバ設定オブジェクトを削除します。

### 1.1.5.3.1 LDAP サーバ設定オブジェクトのアクティブ化または中断 (SQL Central の場合)

LDAP サーバで使用する LDAP サーバ設定オブジェクトをアクティブにしたり、中断したりします。

#### 前提条件

MANAGE ANY LDAP SERVER システム権限が必要です。

#### コンテキスト

LDAP サーバ設定オブジェクトを修正した後は、そのオブジェクトを再び有効にする必要があり、LDAP サーバ設定オブジェクトを修正したり削除するには、そのオブジェクトを無効にする必要があります。

#### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[LDAP サーバ](#)をクリックします。
3. 右ウィンドウ枠で、LDAP サーバ設定を右クリックし、[アクティブ化](#)または[中断](#)をクリックします。
4. [OK](#) をクリックします。

#### 結果

LDAP サーバ設定オブジェクトのステータスが更新されます。

## 1.1.5.3.2 LDAP サーバ設定オブジェクトの変更 (SQL Central の場合)

LDAP ユーザ認証用の LDAP サーバに接続するための LDAP サーバ設定オブジェクトを変更します。

### 前提条件

MANAGE ANY LDAP SERVER システム権限が必要です。

### コンテキスト

SQL Central は、LDAP サーバ設定オブジェクトを変更する前に無効にして、操作が完了したら再び有効にします。

### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[LDAP サーバ](#)をクリックします。
3. 右ウィンドウ枠で、変更する LDAP サーバ設定を右クリックし、[プロパティ](#)をクリックします。
4. [一般タブ](#)と[設定タブ](#)のプロパティを必要に応じて編集して、[OK](#)をクリックします。

### 結果

ISYSLDAPSERVER システムテーブルでの LDAP サーバ設定オブジェクトの設定が更新されます。

### 関連情報

[LDAP サーバ設定オブジェクトのアクティブ化または中断 \(SQL Central の場合\)](#) [135 ページ]

### 1.1.5.3.3 LDAP サーバ設定オブジェクトの検証 (SQL Central の場合)

LDAP サーバに接続して、LDAP 設定オブジェクトが正しく設定されていることを確認します。

#### 前提条件

MANAGE ANY LDAP SERVER システム権限が必要です。

LDAP サーバ設定オブジェクトに関連付けられている LDAP サーバにアクセスできる必要があります。

#### コンテキスト

LDAP 設定オブジェクトの設定を使用してデータベースサーバから LDAP サーバに接続し、それらが有効であることを確認します。

#### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[LDAP サーバ](#)をクリックします。
3. 右ウィンドウ枠で、LDAP サーバ設定を右クリックし、[接続テスト](#)をクリックします。

#### 結果

データベースサーバによって LDAP サーバ設定オブジェクトが検証されます。検証中に問題が発生した場合は、データベースサーバから警告が返されます。

## 1.1.5.3.4 LDAP サーバ設定オブジェクトの削除 (SQL Central の場合)

LDAP サーバへの接続に使用される LDAP サーバ設定オブジェクトを削除します。

### 前提条件

MANAGE ANY LDAP SERVER システム権限が必要です。

### コンテキスト

SQL Central によって、ログインポリシーから LDAP サーバ設定オブジェクトのすべての参照が削除されてから、LDAP サーバ設定オブジェクトが削除されます。

SQL Central では、LDAP サーバ設定オブジェクトが自動的に停止されてから削除されます。

### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[LDAP サーバ](#)をクリックします。
3. 右ウィンドウ枠で、LDAP サーバ設定を右クリックし、[削除](#)をクリックします。
4. [はい](#)をクリックして削除を確定します。

### 結果

データベースサーバによって LDAP サーバ設定オブジェクトが削除され、その参照が ISYSLDAPSERVER システムテーブルから削除されます。

## 1.1.5.4 チュートリアル: LDAP ユーザ認証環境の作成

このチュートリアルでは、Interactive SQL または SQL Central で LDAP ユーザ認証環境をセットアップする基本的な手順について説明します。

### 前提条件

これらのタスクを実行するには、次のシステム権限が必要です。

- SET ANY SECURITY OPTION
- MANAGE ANY USER
- MANAGE ANY LOGIN POLICY
- MANAGE ANY LDAP SERVER

### コンテキスト

このチュートリアルでは、次の作業の方法について説明します。

- LDAP サーバ設定オブジェクトを作成します。
- LDAP サーバを使用するログインポリシーを作成します。
- 定義したログインポリシーを使用して LDAP サーバに認証されるユーザを作成します。

このセクションの内容:

#### [レッスン 1: LDAP サーバ設定オブジェクトの作成 \[140 ページ\]](#)

SQL Central を使用して 2 つの LDAP サーバ設定オブジェクトを作成します。

#### [レッスン 2: LDAP サーバを使用するログインポリシーの作成 \[141 ページ\]](#)

SQL Central を使用して、LDAP サーバを使用するログインポリシーを作成します。

#### [レッスン 3: LDAP サーバに認証されるユーザの作成 \[142 ページ\]](#)

SQL Central を使用して、LDAP サーバに認証されるユーザを作成します。

#### [レッスン 4: \(省略可\) Interactive SQL での LDAP ユーザ認証環境の作成 \[143 ページ\]](#)

LDAP ユーザ認証環境を作成するには、LDAP サーバ設定オブジェクトを作成して、LDAP サーバを使用するログインポリシーを作成し、そのログインポリシーを使用して LDAP サーバに認証されるユーザを作成します。

### 関連情報

[レッスン 2: LDAP サーバを使用するログインポリシーの作成 \[141 ページ\]](#)

## 1.1.5.4.1 レッスン 1: LDAP サーバ設定オブジェクトの作成

SQL Centralを使用して2つのLDAPサーバ設定オブジェクトを作成します。

### 前提条件

このチュートリアル冒頭に記載されているシステム権限を持っている必要があります。

### コンテキスト

各LDAPサーバにはTCP/IP経由でアクセスします。LDAPサーバの定義は、ユーザ認証に使用するLDAPサーバごとに必要です。次に示す応答は、デモンストレーションのみを目的としたものです。適切な同等の応答を指定してください。

### 手順

1. SQL Centralで、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、*LDAP サーバ*を右クリックし、**新規** > *LDAP サーバ* をクリックします。
3. LDAPサーバの名前では、**prim\_ldap**と入力し、**次へ**をクリックします。
4. ネットワーク暗号化のタイプを選択します。**暗号化なし (ldap:)**。
5. LDAPサーバのホスト名とポート番号では、サーバ名とポートを入力します (**iq10web** および **389**)。
6. **認証 URL** に URL が表示されます (ldap://iq10web:389/ など)。**次へ**をクリックします。
7. 検索 URL を **検索 URL** フィールドに直接入力します (**ldap://iq10web:389/dc=sap,dc=com?dn?sub?uid=\***)。
8. **識別名** に識別名を入力します (**cn=Manager,dc=sap,dc=com**)。
9. LDAPサーバでの認証にパスワードが必要な場合、**パスワード**フィールドに入力します (**Not4YourEyes** など)。
10. **次へ**をクリックします。
11. 接続タイムアウトでは、接続タイムアウトの間隔を選択します。
12. 接続リトライ回数では、回数を選択して、**次へ**をクリックします。
13. LDAPサーバをアクティブにするには、**このLDAPサーバを今すぐアクティブ化**を選択します。
14. **ログインモード**を選択します (少なくとも**標準**と**LDAPUA**は選択する必要があります)。
15. **接続テスト**をクリックして、LDAP接続パラメータを検証します。接続パラメータを正しく入力していた場合は、接続に成功します。**閉じる**をクリックしてから、**次へ**をクリックします。
16. (省略可)コメントを追加します。**次へ**をクリックします。
17. LDAPサーバ設定オブジェクトを作成するために実行されるSQLが表示されます。**終了**をクリックします。

## 結果

LDAP サーバ設定オブジェクトが作成され、そのオブジェクトへの参照が ISYSLDAPSERVER システムテーブルに追加されます。

## 次のステップ

次のレッスンに進みます。

### 1.1.5.4.2 レッスン 2: LDAP サーバを使用するログインポリシーの作成

SQL Central を使用して、LDAP サーバを使用するログインポリシーを作成します。

## 前提条件

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に記載されているシステム権限を持っている必要があります。

## コンテキスト

LDAP サーバを使用してユーザを認証する場合は、ログインポリシーが必要です。次に示す応答は、デモンストレーションのみを目的としたものです。適切な同等の応答を指定してください。

## 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、ログインポリシーを右クリックし、**新規** > **ログインポリシー** をクリックします。
3. ログインポリシーの名前 (**ldap\_policy\_both** など) を入力して、**次へ** をクリックします。
4. **LDAP プライマリサーバ** を選択します (**prim\_ldap** など)。
5. (省略可) **LDAP セカンダリサーバ** を選択します (**sec\_ldap** など)。
6. **次へ** をクリックします。
7. (省略可) コメントを追加します。**次へ** をクリックします。
8. ログインポリシーを作成するために実行される SQL が表示されます。**終了** をクリックします。

## 結果

ユーザ認証に LDAP サーバを使用するログインポリシーが作成され、その定義が SYSLOGINPOLICY システムテーブルと SYSLOGINPOLICYOPTION システムテーブルに追加されます。

## 次のステップ

次のレッスンに進みます。

## 関連情報

[レッスン 3: LDAP サーバに認証されるユーザの作成 \[142 ページ\]](#)

### 1.1.5.4.3 レッスン 3: LDAP サーバに認証されるユーザの作成

SQL Central を使用して、LDAP サーバに認証されるユーザを作成します。

## 前提条件

このチュートリアル of これまでのレッスンを完了している必要があります。

このチュートリアル of 冒頭に記載されているシステム権限を持っている必要があります。

## コンテキスト

LDAP サーバに認証されるユーザがデータベースに特定される必要があります。次に示す応答は、デモンストレーションのみを目的としたものです。適切な同等の応答を指定してください。

## 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、**ユーザ**を右クリックし、**新規 ユーザ**をクリックします。
3. 新しいユーザの名前 (**ldap\_user01** など) を入力して、**次へ**をクリックします。

4. このユーザにパスワードを割り当てるが選択されていないことを確認します。
5. ログインポリシーの項目で、ログインポリシーを選択します (`ldap_policy_both` など)。
6. 次へをクリックします。
7. (省略可) コメントを追加します。次へをクリックします。
8. ユーザを作成するために実行される SQL が表示されます。終了をクリックします。

## 結果

LDAP サーバを使用して認証されるデータベースユーザが作成されます。LDAP サーバを使用できない場合は標準認証が行われます。

## 関連情報

[レッスン 2: LDAP サーバを使用するログインポリシーの作成 \[141 ページ\]](#)

### 1.1.5.4.4 レッスン 4: (省略可) Interactive SQL での LDAP ユーザ認証環境の作成

LDAP ユーザ認証環境を作成するには、LDAP サーバ設定オブジェクトを作成して、LDAP サーバを使用するログインポリシーを作成し、そのログインポリシーを使用して LDAP サーバに認証されるユーザを作成します。

## 前提条件

LDAP サーバが必要です。

このチュートリアルのものでのレッスンを完了している必要があります。

このチュートリアルの冒頭に記載されているシステム権限を持っている必要があります。

## 手順

1. (省略可) LDAP サーバとの通信で TLS 暗号化を有効にするには、使用するサーバの証明書ファイルを指定します。例:

```
SET OPTION PUBLIC.trusted_certificates_file='c:¥¥certificates¥¥trusted.txt';
```

以降のステップで TLS 暗号化を有効にするには、それぞれの例で出現する TLS OFF 句を TLS ON 句に変更します。

2. VALIDATE LDAP SERVER 文を実行して、LDAP サーバへの接続をテストします。

たとえば、次の文は既存の LDAP サーバの接続属性を検証します。データベースサーバは、指定されたクレデンシャルを使用して LDAP サーバに接続します。

```
VALIDATE LDAP SERVER
SEARCH DN
  URL 'ldap://iq10web:389/dc=sap,dc=com?dn?sub?uid=*'
  ACCESS ACCOUNT 'cn=Manager,dc=sap,dc=com'
  IDENTIFIED BY 'Not4YourEyes'
AUTHENTICATION URL 'ldap://iq10web:389/'
CONNECTION TIMEOUT 1000
CONNECTION RETRIES 3
TLS OFF;
```

この LDAP サーバは、ポート 389 を使用して通信します。この例では TLS 暗号化を有効にしません (TLS OFF)。次の手順に進む前に、VALIDATE LDAP SERVER 文がエラーなしで実行される必要があります。

3. CREATE LDAP SERVER 文を実行して、LDAP サーバ設定オブジェクトを作成します。

たとえば、次の文はユーザ認証に使用できる LDAP サーバ設定オブジェクトを定義します。

```
CREATE LDAP SERVER prim_ldap
SEARCH DN
  URL 'ldap://iq10web:389/dc=sap,dc=com?dn?sub?uid=*'
  ACCESS ACCOUNT 'cn=Manager,dc=sap,dc=com'
  IDENTIFIED BY 'Not4YourEyes'
AUTHENTICATION URL 'ldap://iq10web:389/'
CONNECTION TIMEOUT 1000
CONNECTION RETRIES 3
TLS OFF;
```

VALIDATE LDAP SERVER 文とは異なり、CREATE LDAP SERVER 文は LDAP サーバへの接続を試行しません。

4. CREATE LDAP SERVER 文を実行して、フェイルオーバーに使用する 2 つ目の LDAP サーバ設定オブジェクトを作成します。この手順は省略できますが、以降の手順で必要になります。

たとえば、次の文はユーザ認証のフェイルオーバーとして使用できる LDAP サーバ設定オブジェクトを定義します。

```
CREATE LDAP SERVER sec_ldap
SEARCH DN
  URL 'ldap://iq10web:390/dc=sap,dc=com?dn?sub?uid=*'
  ACCESS ACCOUNT 'cn=Manager,dc=sap,dc=com'
  IDENTIFIED BY 'Not4YourEyes'
AUTHENTICATION URL 'ldap://iq10web:390/'
CONNECTION TIMEOUT 1000
CONNECTION RETRIES 3
TLS OFF;
```

この LDAP サーバは、ポート 390 を使用して通信します。

5. SET OPTION 文を実行してログインモードを変更し、LDAP ユーザ認証を有効にします。

たとえば、次の文は標準ユーザ認証と LDAP ユーザ認証の両方を有効にします。

```
SET OPTION PUBLIC.login_mode='Standard,LDAPUA';
```

さらにセキュリティを強化するには、SET TEMPORARY OPTION によって login\_mode オプションを設定します。このように設定することで、データベースがコピーされた場合に、なりすましの LDAP サーバによる不正なアクセスを防ぐことができます。

```
SET TEMPORARY OPTION PUBLIC.login_mode='Standard,LDAPUA';
```

login\_mode オプションの現在の設定は次のようにして照会できます。

```
SELECT connection_property('login_mode');
```

6. CREATE LOGIN POLICY 文を実行して新しいログインポリシーを作成します。

たとえば、次の文はユーザの認証に使用できる新しいログインポリシーを作成します。

```
CREATE LOGIN POLICY ldap_policy_both  
LDAP_PRIMARY_SERVER=prim_ldap  
LDAP_SECONDARY_SERVER=sec_ldap  
LDAP_FAILOVER_TO_STD=ON;
```

プライマリ LDAP サーバとセカンダリ LDAP サーバの名前は、このログインポリシーで指定されます。このログインポリシーに関連付けられているユーザの認証が失敗した場合は、標準認証が試行されます (ログインモードによって許可されている場合)。ログインポリシーの現在の設定は次のようにして照会できます。

```
SELECT lpo.* FROM SYS.SYSLOGINPOLICYOPTION AS lpo,  
SYS.SYSLOGINPOLICY AS lp  
WHERE lpo.login_policy_id = lp.login_policy_id  
AND lp.login_policy_name = 'ldap_policy_both';
```

7. CREATE USER 文を実行して、前述の手順で定義された LDAP ログインポリシーを備えた新しいユーザ ID を作成します。

たとえば、次の文はプライマリ LDAP サーバまたはセカンダリ LDAP サーバに対して認証される新しいユーザ ID を作成します。

```
CREATE USER ldap_user01 LOGIN POLICY ldap_policy_both;
```

IDENTIFIED BY 句は省略します。IDENTIFIED BY 句は、標準認証に使用するパスワードを指定します。この句を使用してパスワードを指定する場合は、LDAP サーバによって認証されるパスワードと同じにする必要はありません。このパスワードは、ユーザが初めて LDAP に正しく認証されると置き換えられるため、ここでは IDENTIFIED BY 句を省略します。

8. LDAP サーバをアクティブにしてすぐに使用できるようにします。次の文はプライマリ LDAP サーバとセカンダリ LDAP サーバをアクティブにします。

```
ALTER LDAP SERVER prim_ldap WITH ACTIVATE;  
ALTER LDAP SERVER sec_ldap WITH ACTIVATE;
```

次のクエリは、すべての LDAP サーバの現在の状態を決定します。

```
CALL sa_get_ldapservers_status;
```

結果セットの ldsrv\_state カラムは、2 つの LDAP サーバのステータスが READY であることを示しています。

9. Interactive SQL の CONNECT 文を実行し、LDAP ユーザ認証を使用してデータベースに接続します。

たとえば、次の文は指定されたユーザ ID とパスワードを使用してサンプルデータベースに接続します。

```
CONNECT DATABASE demo USER ldap_user01 IDENTIFIED BY 'abcd1234';
```

LDAP サーバがユーザ認証に失敗すると、標準認証が試行されます。標準パスワードが LDAP 認証の際に更新されていないなどの理由で、標準パスワードが指定のパスワードと一致しない場合、標準認証は失敗します。

## 結果

ユーザが LDAP ユーザ認証を介してデータベースに接続されます。

## 関連情報

[login\\_mode オプション \[717 ページ\]](#)

[trusted\\_certificates\\_file オプション \[817 ページ\]](#)

### 1.1.5.5 PAM ユーザ認証

PAM (Pluggable Authentication Module) は、複数の低レベル認証スキームに共通のインタフェースを提供する認証メカニズムであり、ユーザはすべてのシステムで同じクレデンシャルを使用できます。

PAM 認証により、外部システムの各所で個別に使用されているモジュールのユーザ認証が一本化されます。認証モジュールを作成して、そのモジュールが使用されるよう PAM を設定しておけば、どの認証メカニズムでも PAM システムにプラグインすることができます。

サポート対象のすべての UNIX および Linux プラットフォームで、PAM ユーザ認証 (PAMUA) が使用可能です。ご使用のプラットフォームで PAMUA の設定を行うには、オペレーティングシステムのマニュアルを参照してください。

このセクションの内容:

[PAM ユーザ認証の有効化 \(SQL\) \[147 ページ\]](#)

PAM ユーザ認証を使用できるようデータベースを設定します。

[PAM ユーザ認証の設定 \(SQL Central\) \[148 ページ\]](#)

PAM ユーザ認証を使用できるようデータベースを設定します。

## 関連情報

[ログインポリシーオプションとデフォルト値 \[614 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

## 1.1.5.5.1 PAM ユーザ認証の有効化 (SQL)

PAM ユーザ認証を使用できるようデータベースを設定します。

### 前提条件

- PAM は UNIX システム上で設定を行う必要があります。
- 手順 2 では、SET ANY SECURITY OPTION システム権限が必要です。
- 手順 3 および 手順 4 では、MANAGE ANY LOGIN POLICY システム権限が必要です。

### コンテキスト

次の例では、PAM サービス名 `pam_rule` を設定済であるものとします。

### 手順

1. Interactive SQL で、データベースに接続します。
2. ログインポリシーを作成します。これにより、`pam_service_name` ログインオプションにサービス名 `pam_rule` を割り当て、PAMUA が認証を実行できない場合にデータベースがフェイルオーバーして通常の認証に切り替えるかどうかを指定します。次に例を示します。

```
CREATE LOGIN POLICY pam_policy
PAM_SERVICE_NAME = pam_rule
PAM_FAILOVER_TO_STD = ON;
```

3. 新しいユーザを作成し、新しいログインポリシーを割り当てます。

```
CREATE USER pam_userID LOGIN POLICY pam_policy;
```

4. `login_mode` データベースオプションに PAMUA 値を追加します。

```
SET TEMPORARY OPTION public.login_mode = "PAMUA,STANDARD";
```

5. データベース接続のテストを行う `dbping` ユーティリティを使用して、PAMUA によるログオンが可能であることを確認します。

```
dbping -c "UID=pam_userID;PWD=pam-user-password;LINKS=tcPIP(host=computer-name;PORT=port-number);DBN=database-name -d
```

## 結果

ユーザ pam\_userID は PAMUA によって UNIX の認証情報を使用してデータベースにログインできるようになります。

## 関連情報

[ログインポリシーオプションとデフォルト値 \[614 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

### 1.1.5.5.2 PAM ユーザ認証の設定 (SQL Central)

PAM ユーザ認証を使用できるようデータベースを設定します。

## 前提条件

- PAM の設定は、PAM サービスを含む UNIX システム上で行う必要があります。
- 手順 2 では、SET ANY SECURITY OPTION システム権限が必要です。
- 手順 3 および 手順 4 では、MANAGE ANY LOGIN POLICY システム権限が必要です。

## 手順

1. データベースに接続します。
2. PAM 用のログインポリシーを作成します。
  - a. 左ウィンドウ枠で、**ログインポリシー** > **新規** > **ログインポリシー** を右クリックします。
  - b. ログインポリシーの名前 (pam\_login\_policy など) を指定します。
  - c. ログインポリシーのオプションを設定するよう求められたら、**PAM サービス名**に PAM サービス名を指定します。
3. ログインポリシーをユーザに割り当てます。
  - a. 左ウィンドウ枠で、**ユーザ**をクリックします。
  - b. 右ウィンドウ枠で、ユーザを右クリックし、**プロパティ**をクリックします。
  - c. **ログインポリシー**ドロップダウンリストから、PAM ログインポリシーを選択します。
4. login\_mode データベースオプションに PAMUA 値を追加します。
  - a. 左ウィンドウ枠で、データベースを右クリックし、**オプション**をクリックします。
  - b. **login\_mode** オプションを選択し、値のリストに **PAMUA** を追加します。
  - c. **恒久的な設定を行う**をクリックします。

5. データベース接続のテストを行う dbping ユーティリティを使用して、PAMUA によるログオンが可能であることを確認します。

```
dbping -c "UID=pam_userID;PWD=pam-user-password;LINKS=tcPIP(host=computer-name;PORT=port-number);DBN=database-name -d
```

## 結果

ユーザ pam\_userID は PAMUA によって UNIX の認証情報を使用してデータベースにログインできるようになります。

## 関連情報

[ログインポリシーオプションとデフォルト値 \[614 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

### 1.1.5.6 Kerberos ユーザ認証

Kerberos ログイン機能を使用すると、データベース接続、オペレーティングシステム、ネットワークのログインを、単一のユーザ ID とパスワードで管理できます。

Kerberos ログインの使用は、ユーザにとって便利であると同時に、1 つのセキュリティシステムでデータベースとネットワークのセキュリティを維持できます。次のような利点があります。

- ユーザはデータベースに接続するときにユーザ ID やパスワードを入力しなくてよい
- 複数のユーザを 1 つのデータベースユーザ ID にマッピングできる
- Kerberos へのログインに使用する名前とパスワードはデータベースユーザの ID とパスワードと一致している必要がない

Kerberos は、シークレットキー暗号法を使用し、強力な認証と暗号化を可能にするネットワーク認証プロトコルです。Kerberos にログインしているユーザは、ユーザ ID やパスワードを指定しなくてもデータベースに接続できます。

認証に Kerberos を使用できます。認証を Kerberos に委任するには、次の処理を行います。

- Kerberos ログインを使用するようにサーバとデータベースを設定します。
- コンピュータまたはネットワークにログインするユーザ ID と、データベースユーザの間にマッピングを作成します。

#### 警告

Kerberos ログインを単一のセキュリティソリューションとして使用する場合は、コピーしたデータベースに関連するセキュリティ上の懸念について必ず把握しておきます。

Kerberos ソフトウェアは SQL Anywhere に含まれません。このソフトウェアは別途入手してください。Kerberos ソフトウェアには次のコンポーネントが含まれます。

#### Kerberos ライブラリ

Kerberos クライアントまたは GSS (Generic Security Services)-API ランタイムライブラリと呼ばれています。Kerberos ライブラリは、明確に定義されている GSS-API を実装したもので、Kerberos を使用するクライアントコンピュータおよびサーバコンピュータでそれぞれ必要です。KDC として Active Directory を使用する場合は、サードパーティ製 Kerberos クライアントライブラリの代わりに、組み込みの Windows SSPI インタフェースを使用できます。

SSPI は、クライアントのみが Kerberos 接続パラメータで使用できます。データベースサーバは SSPI を使用できません。SSPI 以外でサポートされている Kerberos クライアントを使用する必要があります。

#### **Kerberos キー配布センター (KDC) サーバ**

KDC はユーザおよびサーバの保管場所として機能します。また、ユーザやサーバの ID を検証します。KDC は、通常、アプリケーションやユーザログインに使用しないサーバコンピュータにインストールされます。

DBLib クライアント、ODBC クライアント、OLE DB クライアント、ADO.NET クライアント、SAP Open Client、jConnect クライアントからの Kerberos 認証がサポートされています。Kerberos 認証は SQL Anywhere トランスポートレイヤセキュリティ暗号化と組み合わせて使用できますが、ネットワーク通信での Kerberos 暗号化はサポートされていません。

Windows では、Kerberos を Windows ドメインとドメインアカウントに使用します。Active Directory Windows Domain Controllers は Kerberos KDC を実装します。この環境で認証を行うには、データベースサーバコンピュータにサードパーティ製 Kerberos クライアントまたはランタイムが必要ですが、Windows クライアントコンピュータは、サードパーティ製 Kerberos クライアントまたはランタイムの代わりに、組み込み Windows SSPI インタフェースを使用できます。

このセクションの内容:

#### [Kerberos クライアント \[151 ページ\]](#)

Kerberos 認証がサーバプラットフォームで利用できます。

#### [Kerberos システムの設定 \[152 ページ\]](#)

SQL Anywhere で使用するよう Kerberos 認証を設定します。

#### [Kerberos を使用するための SQL Anywhere データベースの設定 \(SQL\) \[153 ページ\]](#)

Kerberos ログインを使用するためにデータベースを設定します。

#### [SAP Open Client または jConnect アプリケーションからの接続 \[155 ページ\]](#)

データベースサーバは、SAP Open Client または jConnect アプリケーションから接続を受け付けます。

#### [Kerberos ログインマッピングの作成 \(SQL Central の場合\) \[156 ページ\]](#)

SQL Central を使用して、Kerberos 認証の個々のユーザをデータベースサーバに設定する Kerberos ログインマッピングを作成します。

#### [Kerberos ログインマッピングの取り消し \(SQL Central の場合\) \[157 ページ\]](#)

SQL Central を使用して、Kerberos ログインマッピングを取り消します。

#### [Windows で Kerberos ログインに SSPI を使用した接続 \[158 ページ\]](#)

クライアントコンピュータにインストールされた Kerberos クライアントなしで SSPI を使用して接続します。

#### [トラブルシューティング: Kerberos 接続 \[159 ページ\]](#)

Kerberos 認証を有効にしようとしたり、使おうとしたときに予期しないエラーが発生した場合は、データベースサーバとクライアントで追加の診断メッセージを有効にすることをお奨めします。

## 関連情報

[セキュリティ: ログインモードを使用したデータベースのセキュリティ保護 \[162 ページ\]](#)

## 1.1.5.6.1 Kerberos クライアント

Kerberos 認証がサーバプラットフォームで利用できます。

サポートされている Kerberos クライアントが使用する keytab ファイルと GSS-API ファイルのデフォルトの名前とロケーションのリストを次の表に示します。

### i 注記

SSPI は、クライアントのみが Kerberos 接続パラメータで使用できます。データベースサーバは SSPI を使用できません。SSPI 以外でサポートされている Kerberos クライアントを使用する必要があります。

Kerberos クライアント	デフォルト keytab ファイル	GSS-API ライブラリファイル名	注記
Windows MIT Kerberos クライアント	C:¥WINDOWS¥krb5kt	gssapi32.dll または gssapi64.dll	KRB5_KTNAME 環境変数を設定してからデータベースサーバを起動することで、別の keytab ファイルを指定できます。
Windows CyberSafe Kerberos クライアント	C:¥Program Files ¥CyberSafe¥v5srvtab	gssapi32.dll または gssapi64.dll	CSFC5KTNAME 環境変数を設定してからデータベースサーバを起動することで、別の keytab ファイルを指定できます。
UNIX MIT Kerberos クライアント	/etc/krb5.keytab	libgssapi_krb5.so <sup>1</sup>	KRB5_KTNAME 環境変数を設定してからデータベースサーバを起動することで、別の keytab ファイルを指定できます。
UNIX CyberSafe Kerberos クライアント	/krb5/v5srvtab	libgss.so <sup>1</sup>	CSFC5KTNAME 環境変数を設定してからデータベースサーバを起動することで、別の keytab ファイルを指定できます。
UNIX Heimdal Kerberos クライアント	/etc/krb5.keytab	libgssapi.so.1 <sup>1</sup>	

<sup>1</sup>ファイル名はオペレーティングシステムと Kerberos クライアントバージョンによって異なります。

## 関連情報

[SQL Anywhere がサポートする Kerberos クライアント](#)

## 1.1.5.6.2 Kerberos システムの設定

SQL Anywhere で使用するように Kerberos 認証を設定します。

### 前提条件

Kerberos 認証を使用してコンピュータにログインする必要があります。

### コンテキスト

Kerberos は、シークレットキー暗号法を使用し、強力な認証と暗号化を可能にするネットワーク認証プロトコルです。

### 手順

1. 必要に応じ、GSS-API ランタイムライブラリなどの Kerberos クライアントソフトウェアをクライアントとサーバの両方にインストールして設定を行います。

Active Directory の KDC (Key Distribution Center) を使用している Windows クライアントコンピュータでは SSPI を使用できるので、Kerberos クライアントをインストールする必要はありません。

2. 必要に応じ、Kerberos KDC に各ユーザの Kerberos プリンシパルを作成します。

Kerberos プリンシパルとは Kerberos ユーザ ID であり、形式は `user/instance@REALM` です。`/instance` はオプションです。すでに Kerberos を使用している場合にはプリンシパルが存在しているはずですので、各ユーザの Kerberos プリンシパルを作成する必要はありません。

プリンシパルの大文字と小文字は区別されるため、正しく指定する必要があります。大文字と小文字の違いしかない複数のプリンシパルのマッピングはサポートされていません (たとえば、`jjordan@MYREALM.COM` と `JJordan@MYREALM.COM` の両方にマッピングすることはできません)。

3. Kerberos プリンシパルを SQL Anywhere データベースサーバの KDC に作成します。

データベースサーバ用のデフォルトの Kerberos プリンシパルの形式は `server-name@REALM` です。`server-name` は SQL Anywhere データベースサーバ名です。別のサーバプリンシパルを使用する場合は、`-kp` サーバオプションを使用します。プリンシパルは大文字と小文字が区別されます。`server-name` には、マルチバイト文字、`/`、`¥`、または `@` を含めることはできません。

サーバでは KDC 認証に keytab ファイルが使用されるので、KDC 内にサーバサービスプリンシパルを作成する必要があります。keytab ファイルは保護および暗号化されます。

4. セキュリティに十分注意しながら、プリンシパル `server-name@REALM` 用の keytab ファイルを KDC から抽出し、SQL Anywhere データベースサーバを実行中のコンピュータにコピーします。keytab ファイルのデフォルトロケーションは、Kerberos クライアントとプラットフォームによって異なります。keytab ファイルのパーミッションは、SQL Anywhere サーバが読み取ることができ、不正なユーザに読み取りパーミッションがないよう設定する必要があります。

## 結果

Kerberos システムが認証され、SQL Anywhere で使用するように設定されます。

## 次のステップ

Kerberos を使用するように、SQL Anywhere データベースサーバとデータベースを設定します。

## 関連情報

[Kerberos を使用するための SQL Anywhere データベースの設定 \(SQL\) \[153 ページ\]](#)

[Windows で Kerberos ログインに SSPI を使用した接続 \[158 ページ\]](#)

### 1.1.5.6.3 Kerberos を使用するための SQL Anywhere データベースの設定 (SQL)

Kerberos ログインを使用するためにデータベースを設定します。

## 前提条件

SET ANY PUBLIC OPTION および MANAGE ANY USER システム権限が必要です。

Kerberos の設定について、SQL Anywhere で使用できるように設定済である必要があります。

## コンテキスト

Kerberos ログイン機能を使用すると、データベース接続、オペレーティングシステム、ネットワークのログインを、単一のユーザ ID とパスワードで管理できます。

## 手順

1. データベースサーバを -krb または -kr オプションを使用して起動し、Kerberos 認証を有効にします。また、-kl オプションを使用して GSS-API ライブラリのロケーションを指定し、Kerberos を有効にすることもできます。

- パブリックオプションまたは一時的なパブリックオプション `login_mode` の値に Kerberos を含めるよう変更します。データベースオプションは、それが指定されているデータベースにしか適用されないため、同じデータベースサーバ内にロードされて動作している場合でも、データベースが異なれば Kerberos ログインの設定も異なります。次に例を示します。

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Kerberos,Standard';
```

#### 警告

`login_mode` データベースオプションを Kerberos に設定すると、接続できるのは、Kerberos ログインマッピングを付与されているユーザだけに制限されます。SYS\_AUTH\_DBA\_ROLE 互換性ロールを持つユーザでないかぎり、ユーザ ID とパスワードを使用して接続しようとすると、エラーが発生します。

- クライアントユーザのデータベースユーザ ID を作成します。既存のデータベースユーザに適切な特権があれば、そのデータベースユーザ ID を Kerberos ログインに使用できます。次に例を示します。

```
CREATE USER "kerberos-user"  
IDENTIFIED BY abc123;
```

- GRANT KERBEROS LOGIN TO 文を実行して、クライアントの Kerberos プリンシパルから既存のデータベースユーザ ID へのマッピングを作成します。次に例を示します。

```
GRANT KERBEROS LOGIN TO "pchin@MYREALM.COM"  
AS USER "kerberos-user";
```

マッピングのない Kerberos プリンシパルを使用中に接続する場合は、Guest データベースユーザ ID が存在し、パスワードがあることを確認してください。

- クライアントユーザが Kerberos プリンシパルを使用してログオン済みである (有効な Kerberos TGT: Ticket Granting Ticket がある) こと、およびクライアントの Kerberos チケットの期限が切れていないことを確認します。ドメインアカウントにログインしている Windows ユーザは、TGT をすでに持っており、プリンシパルに必要なパーミッションがあれば、サーバに認証されます。

チケット付与権限付きチケットはユーザパスワードを使用して暗号化された Kerberos チケットで、ユーザ ID の検証に Ticket Granting Service が使用します。

- KERBEROS 接続パラメータ (通常は `KERBEROS=YES`、ただし `KERBEROS=SSPI` または `KERBEROS=GSS-API-library-file` も使用可) を指定して、クライアントから接続します。ユーザ ID またはパスワードの接続パラメータが指定された場合は、無視されます。次に例を示します。

```
dbisql -c "KERBEROS=YES;Server=my_server_princ"
```

## 結果

データベースが Kerberos 認証を使用するように設定されます。

#### 例

ユーザが、データベースサーバのデフォルトデータベースに対する Kerberos ログインマッピングと一致するユーザプロファイル名を使用してログオンした場合、次の SQL 文を使用した接続は成功します。

```
CONNECT USING 'KERBEROS=YES';
```

次のすべてに該当する場合、CONNECT 文でデータベースに接続できます。

- データベースサーバが現在実行中です。
- 現在のデータベースサーバのデフォルトデータベースは、Kerberos で認証された接続を受け入れることが可能です。
- Kerberos ログインマッピングがユーザの現在の Kerberos プリンシパルに対して作成されています。
- データベースサーバから接続についての詳細情報を要求された場合 (Interactive SQL を使用した場合など) に、情報を追加せずに OK をクリックします。

## 次のステップ

Kerberos 認証を使用してクライアントから接続できます。オプションとして、Kerberos ログインマッピングを作成できます。

## 関連情報

[Kerberos ログインマッピングの作成 \(SQL Central の場合\) \[156 ページ\]](#)

[Kerberos システムの設定 \[152 ページ\]](#)

[Windows で Kerberos ログインに SSPI を使用した接続 \[158 ページ\]](#)

[-kl データベースサーバオプション \[441 ページ\]](#)

[-kr データベースサーバオプション \(廃止予定\) \[443 ページ\]](#)

[-krb データベースサーバオプション \[444 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

### 1.1.5.6.4 SAP Open Client または jConnect アプリケーションからの接続

データベースサーバは、SAP Open Client または jConnect アプリケーションから接続を受け付けます。

- Kerberos ユーザ認証を設定します。
- Kerberos を使用するように SQL Anywhere を設定します。
- Adaptive Server Enterprise での Kerberos ユーザ認証と同様に、SAP Open Client または jConnect を設定します。サーバ名は SQL Anywhere サーバ名である必要があります。名前の大文字と小文字は区別されず、Open Client または jConnect から代替サーバ名を使用して接続することはできません。

## 関連情報

[Kerberos システムの設定 \[152 ページ\]](#)

[Kerberos を使用するための SQL Anywhere データベースの設定 \(SQL\) \[153 ページ\]](#)

Configuring Kerberos for ASE (ASE で使用するための Kerberos の設定) 

[-krb データベースサーバオプション \[444 ページ\]](#)

[-kr データベースサーバオプション \(廃止予定\) \[443 ページ\]](#)

[-kl データベースサーバオプション \[441 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

[Kerberos \(KRB\) 接続パラメータ \[87 ページ\]](#)

[トラブルシューティング: Kerberos 接続 \[159 ページ\]](#)

## 1.1.5.6.5 Kerberos ログインマッピングの作成 (SQL Central の場合)

SQL Central を使用して、Kerberos 認証の個々のユーザをデータベースサーバに設定する Kerberos ログインマッピングを作成します。

### 前提条件

Kerberos の設定について、SQL Anywhere で使用できるように設定済である必要があります。データベースサーバとデータベースについて、Kerberos を使用できるように設定済である必要があります。

データベースユーザと Kerberos プリンシパルは、常に存在している必要があります。

MANAGE ANY USER システム権限が必要です。また、SELECT ANY TABLE システム権限か、SYSLOGINMAP システムビューに対する SELECT 権限のいずれかが必要です。

ログインモードの変更を行う場合、SET ANY SECURITY OPTION システム権限も必要です。

### コンテキスト

Kerberos ログインマッピングは、コンピュータまたはネットワークにログインするユーザ ID と、データベースユーザとを結びつけるものです。

### 手順

1. *SQL Anywhere 17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、**ログインマッピング**を右クリックし、**新規 > ログインマッピング** をクリックします。
3. **Kerberos ログインマッピング**を選択し、**次へ**をクリックします。
4. データベースに接続する **Kerberos プリンシパル**を指定してくださいというフィールドに、Kerberos ログインを作成する Kerberos プリンシパル名を入力します。

5. [Kerberos プリンシパルに関連付けるデータベースユーザを指定してください](#)というリストで、この Kerberos プリンシパルをマッピングするデータベースユーザ ID を選択します。
6. [ログインマッピング作成ウィザード](#)の残りの指示に従います。

## 結果

Kerberos ログインマッピングが作成されます。

## 関連情報

[Kerberos システムの設定 \[152 ページ\]](#)

[Kerberos を使用するための SQL Anywhere データベースの設定 \(SQL\) \[153 ページ\]](#)

### 1.1.5.6.6 Kerberos ログインマッピングの取り消し (SQL Central の場合)

SQL Central を使用して、Kerberos ログインマッピングを取り消します。

## 前提条件

SQL Anywhere で使用できるようにあらかじめ Kerberos を設定しておく必要があります。データベースサーバとデータベースについて、Kerberos を使用できるように設定済である必要があります。

データベースユーザと Kerberos プリンシパルは、常に存在している必要があります。

MANAGE ANY USER システム権限が必要です。また、SELECT ANY TABLE システム権限か、SYSLOGINMAP システムビューに対する SELECT 権限のいずれかが必要です。

## コンテキスト

Kerberos ログインマッピングは、コンピュータまたはネットワークにログインするユーザ ID と、データベースユーザとを結びつけるものです。

## 手順

1. [SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[ログインマッピング](#)をクリックします。
3. 右ウィンドウ枠で、ログインマッピングを右クリックし、[削除](#)をクリックします。
4. [はい](#)をクリックします。

## 結果

Kerberos ログインマッピングが取り消されます。

## 関連情報

[Kerberos システムの設定 \[152 ページ\]](#)

[Kerberos を使用するための SQL Anywhere データベースの設定 \(SQL\) \[153 ページ\]](#)

### 1.1.5.6.7 Windows で Kerberos ログインに SSPI を使用した接続

クライアントコンピュータにインストールされた Kerberos クライアントなしで SSPI を使用して接続します。

## 前提条件

Kerberos の設定について、SQL Anywhere で使用できるように設定済である必要があります。データベースサーバとデータベースについて、Kerberos を使用できるように設定済である必要があります。

## コンテキスト

Windows ドメインでは、クライアントコンピュータに Kerberos クライアントをインストールしなくても、Windows ベースのコンピュータで SSPI を使用できます。Windows ドメインアカウントには、関連付けられた Kerberos プリンシパルがあらかじめ用意されています。

SSPI は、SQL Anywhere クライアントのみが Kerberos 接続パラメータで使用できます。SQL Anywhere データベースサーバは SSPI を使用できません。SSPI 以外でサポートされている Kerberos クライアントを使用する必要があります。

## 手順

クライアントコンピュータからデータベースに接続します。次に例を示します。

```
dbisql -c "KERBEROS=SSPI;Server=my_server_princ"
```

接続文字列に Kerberos=SSPI と指定されている場合、Kerberos ログインが試行されます。

ユーザが、データベースサーバのデフォルトデータベースに対する Kerberos ログインマッピングと一致するユーザプロファイル名を使用してログオンした場合、次の SQL 文を使用した接続は成功します。

```
CONNECT USING 'KERBEROS=SSPI';
```

## 結果

SSPI を使用して、Windows で Kerberos 認証を行うことができます。

## 関連情報

[Kerberos システムの設定 \[152 ページ\]](#)

[Kerberos を使用するための SQL Anywhere データベースの設定 \(SQL\) \[153 ページ\]](#)

### 1.1.5.6.8 トラブルシューティング: Kerberos 接続

Kerberos 認証を有効にしようとしたり、使おうとしたときに予期しないエラーが発生した場合は、データベースサーバとクライアントで追加の診断メッセージを有効にすることをお奨めします。

データベースサーバの起動時に `-z` オプションを指定します。または、すでに実行中のサーバのデータベースサーバメッセージログに追加の診断メッセージを表示するには `CALL sa_server_option( 'DebuggingInformation', 'ON' )` を使用します。LogFile 接続パラメータを使用すると、指定したファイルにクライアント診断メッセージが書き込まれます。

LogFile 接続パラメータを使用する代わりに、`-z` パラメータを使用して Ping ユーティリティ (`dbping`) を実行することができます。`-z` パラメータにより診断メッセージが表示され、接続の問題の原因を特定するのに役立ちます。

## データベースサーバの起動に関する問題

現象	一般的な解決策
メッセージ "Kerberos GSS-API ライブラリをロードできません"	<ul style="list-style-type: none"> <li>● Kerberos クライアントが、GSS-API ライブラリも含めて、データベースサーバコンピュータにインストールされていることを確認します。</li> <li>● ロードしようとしたライブラリ名が、データベースサーバの -z 出力にリストされます。ライブラリ名が正しいことを確認します。必要に応じて、-kl オプションを使用して正しいライブラリ名を指定します。</li> <li>● フォルダと、サポートしているライブラリが、ライブラリパス (Windows では %PATH%) にリストされていることを確認します。</li> <li>● GSS-API ライブラリにエントリポイントがないことがデータベースサーバの -z 出力に示されていた場合、そのライブラリはサポートされている Kerberos Version 5 GSS-API ライブラリではありません。</li> </ul>
メッセージ "サーバ名 "server-name" の Kerberos クレデンシャルを取得できません"	<ul style="list-style-type: none"> <li>● <code>server-name@REALM</code> のプリンシパルが KDC にあることを確認します。プリンシパルは大文字と小文字が区別されるので、データベースサーバ名とプリンシパル名のユーザ部分の大文字と小文字が一致していることを確認します。</li> <li>● SQL Anywhere サーバ名がプリンシパルのプライマリ/ユーザ部分になっていることを確認します。</li> <li>● サーバのプリンシパルが keytab ファイルに抽出されていること、およびその keytab ファイルが Kerberos クライアントに対して適切なロケーションにあることを確認します。</li> <li>● データベースサーバコンピュータ上の Kerberos クライアントの領域がサーバプリンシパルの領域と異なっている場合は、-kr オプションを使用してサーバプリンシパルの領域を指定します。</li> </ul>
クライアントエラー "Kerberos ログインが失敗しました"	<ul style="list-style-type: none"> <li>● データベースサーバの診断メッセージを確認します。サーバが使用する keytab ファイルに関する問題のなかには、クライアントが認証しようとするまで検出されないものがあります。</li> </ul>

## Kerberos クライアント接続のトラブルシューティング

クライアントが Kerberos 認証を使用して接続しようとしてエラーが発生した場合について、次の表に示します。

現象	一般的な解決策
エラー "Kerberos ログインはサポートされていません" が発生し、LogFile にメモリのロードに失敗しました" が出力されている	<ul style="list-style-type: none"> <li>• Kerberos クライアントが、GSS-API ライブラリも含めて、クライアントコンピュータにインストールされていることを確認します。</li> <li>• LogFile で指定されたファイルに、ロードしようとしたライブラリがリストされています。ライブラリ名が正しいことを確認し、必要に応じて Kerberos 接続パラメータを使用して正しいライブラリ名を指定します。</li> <li>• サポートしているライブラリがあるディレクトリがライブラリパス (Windows では %PATH%) にリストされていることを確認します。</li> <li>• GSS-API ライブラリにエントリポイントがないことが LogFile 出力に示されていた場合、そのライブラリはサポートされている Kerberos Version 5 GSS-API ライブラリではありません。</li> </ul>
エラー "Kerberos ログインはサポートされていません"	<ul style="list-style-type: none"> <li>• データベースサーバに対して -krb、-kl、-kr の各サーバオプションが 1 つ以上指定されており、Kerberos ログインが有効になっていることを確認します。</li> <li>• Kerberos ログインがクライアントとサーバの両プラットフォーム上でサポートされていることを確認します。</li> </ul>
エラー "Kerberos ログインが失敗しました"	<ul style="list-style-type: none"> <li>• ユーザが Kerberos にログイン済みであること、およびそのユーザに期限が切れていない有効なチケット付与権限付きチケットがあることを確認します。</li> <li>• クライアントコンピュータとサーバコンピュータとの間で、時刻のずれが 5 分未満であることを確認します。</li> </ul>
エラー "ログインモード 'Kerberos' は、login_mode 設定で許可されていません"	<ul style="list-style-type: none"> <li>• Kerberos ログインが許可されるには、login_mode オプションのパブリックまたは一時的なパブリックのデータベースオプションの設定に値 Kerberos が含まれている必要があります。</li> </ul>
"ログイン ID 'client-Kerberos-principal' はどのデータベースユーザ ID にもマップされていません"	<ul style="list-style-type: none"> <li>• Kerberos プリンシパルが GRANT KERBEROS LOGIN 文を使用してデータベースユーザ ID にマッピングされている必要があります。GRANT KERBEROS LOGIN 文には、領域を含む完全なクライアントプリンシパルが指定されている必要があります。また、インスタンスまたは領域しか変わらないプリンシパルも別のプリンシパルとして扱われます。</li> <li>• また、明示的にマッピングされていない有効な Kerberos プリンシパルを接続可能にするには、GRANT CONNECT を使用して、ゲストデータベースユーザ ID とパスワードを作成します。</li> </ul>

## 関連情報

[-z データベースサーバオプション \[509 ページ\]](#)

[Kerberos クライアント \[151 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

## 1.1.5.7 セキュリティ: ログインモードを使用したデータベースのセキュリティ保護

データベースをセキュリティ保護するための方法はいくつかあります。

標準ログイン、統合化ログイン、Kerberos ログイン、LDAPUA ログイン、PAMUA ログインのうち、データベースに使用を許可する組み合わせを、SET OPTION 文を使用して login\_mode オプションの値に設定すると、データベースは指定したタイプのログインが永続的に可能となります。データベースに対して統合化ログイン、Kerberos ログイン、LDAPUA ログイン、または PAMUA ログインを有効にする場合、オペレーティングシステムまたはネットワークのセキュリティモデルに依存します。たとえば、次の文は標準ログインと統合化ログインを永続的に許可します。

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

データベースを停止して再起動した場合でも、このオプションの値は変わらず、統合化ログインは有効のままです。

SET TEMPORARY OPTION を使用して login\_mode オプションを設定した場合、統合化ログインによるユーザアクセスは可能ですが、データベースがシャットダウンされるまでの間に限られます。次の文は、オプションの値を一時的に変更します。

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

永久オプション値が Standard の場合、データベースは停止時にその値に戻ります。

一時的なパブリックオプションを設定すると、データベースのセキュリティを強化できます。データベースファイルが別のコンピュータにコピーされた場合、統合化ログイン、Kerberos ログイン、LDAPUA ログイン、および PAMUA ログインはデフォルトでは有効になりません。

データベースに機密情報が含まれる場合、データベースファイルが保存されているコンピュータを不正アクセスから保護する必要があります。保護しなかった場合、データベースファイルがコピーされ、別のコンピュータからデータに不正にアクセスされる可能性があります。

データベースのセキュリティを強化するには、次の処理を行います。

- パスワードは複雑にして、推測しにくくします。
- SQL Anywhere の AES 暗号化機能を使用して、データベースファイルを強力的に暗号化します。暗号化キーは複雑にして、推測しにくくします。
- 永続的な PUBLIC.login\_mode データベースオプションを Standard に設定します。統合化ログインまたは Kerberos ログインを有効にする場合は、一時的なパブリックオプションのみがサーバの起動ごとに変更されるようにします。これにより、データベースがコピーされたとしても、可能になるのは標準ログインだけになります。

## 1.1.5.8 セキュリティ: 統合化ログインを使用すると、無制限データベースアクセスが行われる可能性があります

統合化ログイン機能は、ネイティブな Windows 認証を利用してデータベースへのアクセスを制御する機能です。

標準的な SQL Anywhere ユーザ ID とパスワードによるデータベースセキュリティは省略されます。Windows にログインしているユーザのクレデンシャルが、Windows ドメインコントローラへの認証に使用されます。認証ユーザとデータベースのユーザ ID とのマッピングが存在すれば、そのユーザはデータベースへのアクセスを許可されます。

統合化ログインを使用する場合、データベース管理者は、データベースへのアクセスを制限するために、Windows によって使用されるログインセキュリティに特に注意する必要があります。

### 警告

ドメインを指定せずに、ユーザ名に統合化ログインを付与することはお奨めしません。

次のように、データベースへの統合化ログインを有効化し、Windows ユーザとデータベースユーザ ID とのマッピングを作成したとします。

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Standard,Integrated';
GRANT INTEGRATED LOGIN TO DSmith AS USER BROWSER;
```

ユーザは、Windows ドメインに関係なく DSmith として Windows コンピュータへのログインに成功すれば、それ以上の資格情報を提示しなくても、その Windows コンピュータと同じネットワーク上で実行中のデータベースによって認証を受けることができます。

今度は、次のようにデータベースへの統合化ログインを有効化したとします。

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Standard,Integrated';
GRANT INTEGRATED LOGIN TO "MyDomain¥DSmith" AS USER BROWSER;
```

データベースに認証されるためには、ユーザはまず MyDomain ドメインで DSmith として Windows コンピュータへのログインに成功しなければなりません (MyDomain¥DSmith)。そうすれば、それ以上の資格情報を提示しなくても、その Windows コンピュータと同じネットワーク上で実行中のデータベースに接続できます。ドメイン名を含めることで、想定通りの適正なユーザクレデンシャルが指定されることを保証できます。

### 警告

パスワード付きのユーザ ID「Guest」を作成すると、Windows ドメインへの認証を実行できるユーザであれば、誰でもデータベースに統合化ログインアクセスできます。

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Standard,Integrated';
CREATE USER Guest IDENTIFIED BY 'secret';
```

パスワード付きのユーザ ID「Guest」を作成し、login\_mode 設定に統合化を含めておけば、ネットワーク上の Windows ドメインへの認証を実行できるユーザであれば、誰でもデータベースに統合化ログインできます。Windows ユーザ名に対応する統合化ログインマッピングが存在しない場合、Guest ユーザが (存在すれば) 使用されます。Guest ユーザは統合化ログインマッピングを必要としません (Guest に対しては GRANT INTEGRATED LOGIN 文を実行する必要がありません)。

Guest ユーザが存在しない場合、Windows ユーザへのマッピングがなければ統合化ログインは失敗します。

Guest ユーザは割り当てられたロールまたは権限の制約を受けませんが、Guest ユーザ ID の作成は推奨されません。たとえば、データベースのシステムテーブルを調べられる可能性があります。

## 1.1.6 通信プロトコル

データベースサーバはさまざまなプロトコルを使用して通信を行います。サポートしているプロトコルの動作をカスタマイズするためのオプションも多数用意されています。

このセクションの内容:

[通信プロトコルオプションの考慮事項 \[164 ページ\]](#)

クライアントアプリケーションとデータベースサーバ間の通信では、通信プロトコルが必要です。

[TCP/IP プロトコル \[166 ページ\]](#)

TCP/IP は、別のコンピュータ上で実行中のデータベースにクライアントを接続するために使用します。

### 1.1.6.1 通信プロトコルオプションの考慮事項

クライアントアプリケーションとデータベースサーバ間の通信では、通信プロトコルが必要です。

ネットワーク上での通信のための通信プロトコルと、同じコンピュータ上での通信のための通信プロトコルがサポートされています。

#### ネットワークデータベースサーバ (**dbsrv17**) で使用できるプロトコル

デフォルトでは、ネットワークデータベースサーバは共有メモリプロトコルと TCP/IP プロトコルを起動します。TCP/IP プロトコルオプションを提供することも、-x オプションを使用して TCP/IP プロトコルを無効にすることもできます。

共有メモリ

このプロトコルは、同一コンピュータ通信で使用され、常にすべてのプラットフォームで使用可能です。

TCP/IP

このプロトコルは、主に異なるコンピュータ間の通信に使用されます。TDS クライアント (SAP Open Client または jConnect JDBC ドライバ) を使用している場合は、TCP/IP を使用する必要があります。すべてのプラットフォームで使用できます。

#### パーソナルデータベースサーバ (**dbeng17**) で使用できるプロトコル

デフォルトでは、パーソナルデータベースサーバは共有メモリ接続のみを許可します。TCP/IP プロトコルも起動するには、-x オプションを使用します。

共有メモリ

このプロトコルは、同一コンピュータ通信で使用され、常にすべてのプラットフォームで使用可能です。

同一コンピュータ通信では、共有メモリの方が TCP/IP よりもパフォーマンスが高くなります。

TCP/IP

このプロトコルは、パーソナルサーバ上の同一コンピュータ通信に対してのみ使用可能です。また、TDS クライアント (SAP Open Client または jConnect JDBC ドライバ) を使用している場合は、TCP/IP を使用する必要があります。すべてのプラットフォームで使用できます。

## SQL Anywhere クライアントからの接続

### 共有メモリ

共有メモリは、デフォルトの接続プロトコルです。共有メモリ経由で接続する場合は、クライアントの接続文字列には Host と CommLinks (LINKS) のどちらの接続パラメータも指定しないでください。

### TCP/IP

TCP/IP を経由して接続する場合は、Host または CommLinks (LINKS) 接続パラメータのいずれかを指定します。Host 接続パラメータを指定することをお奨めします。

このセクションの内容:

[共有メモリのセッションとターミナルについての考慮事項 \[165 ページ\]](#)

共有メモリのセッションとターミナルについてはいくつかの考慮事項があります。

## 関連情報

[TCP/IP プロトコル \[166 ページ\]](#)

[ネットワークプロトコルオプション \[177 ページ\]](#)

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[-x データベースサーバオプション \[500 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

### 1.1.6.1.1 共有メモリのセッションとターミナルについての考慮事項

共有メモリのセッションとターミナルについてはいくつかの考慮事項があります。

#### 共有メモリ、サービス、ターミナルサービス (Windows 7 以降)

- クライアントは、共有メモリ経由でサービスとして動作する任意のデータベースサーバに接続できます。
- クライアントは、クライアントと同じセッションで実行中であるデータベースサーバに共有メモリ経由で接続できます。共有メモリ接続を行うときに、クライアントは最初にクライアントと同じセッションで指定された名前のデータベースサーバを探します。何も見つからない場合、クライアントは指定された名前のサービスを探します。

- 1つのターミナルセッションで実行中のクライアントは、別のターミナルセッションで実行中のデータベースサーバに対して共有メモリ経由で接続することはできません。代わりに、TCP/IP を使用します。
- サービスとして実行するクライアントは、1つのターミナルセッションで実行中のデータベースサーバに対して共有メモリ経由で接続することはできません。代わりに、TCP/IP を使用します。
- 現在のセッションで同じ名前のデータベースサーバがすでに実行中の場合、または同じ名前で行中のサービスがある場合、データベースサーバは、起動できません。
- 別のセッションで同じ名前で行中のデータベースサーバがある場合は、同じ名前で行中のサービスがなく、データベースサーバがどちらも TCP/IP を使用しない限り、データベースサーバは起動できます。

## 共有メモリ、サービス、ターミナルサービス (Windows 2003)

データベースサーバとクライアントをサービスとしてまたはターミナルサービス内で実行する場合:

- クライアントは、クライアントと同じセッションで実行中であるか、またはプライマリセッションで実行中であるデータベースサーバに共有メモリ経由で接続できます。共有メモリ接続を行うときに、クライアントは最初にクライアントと同じセッション内、次にプライマリセッション内で、指定された名前のデータベースサーバを探します。
- プライマリセッション内で実行中のクライアントは、プライマリ以外のセッションで実行中のデータベースサーバに対して共有メモリ経由で接続することはできません。プライマリ以外のセッション内で実行中のクライアントは、プライマリ以外の別のセッションで実行中のデータベースサーバに対して共有メモリ経由で接続することはできません。代わりに、TCP/IP を使用します。
- 現在のセッション内またはプライマリセッション内で同じ名前データベースサーバがすでに実行中である場合、データベースサーバは起動できません。(プライマリセッションの) デスクトップ上で実行中のデータベースサーバは、同じ名前のサービスが起動するのを妨げる可能性があります。ただし、通常、サービスは、システムが起動するときに起動するので、この問題が発生するのはまれです。
- プライマリ以外の別のセッションで共有メモリ経由で実行中の同じ名前のデータベースサーバがある場合であっても、プライマリセッションで同じ名前で行中のデータベースサーバが存在しない限り、データベースサーバは共有メモリを使って起動できます。
- 他の任意のセッション (プライマリまたはプライマリ以外) で TCP/IP 経由で実行中の同じ名前のデータベースサーバが存在する場合、データベースサーバは TCP/IP を使っては起動できません。

### 1.1.6.2 TCP/IP プロトコル

TCP/IP は、別のコンピュータ上で実行中のデータベースにクライアントを接続するために使用します。

TCP/IP プロトコルを使用するときは、トランスポートレイヤセキュリティと RSA 暗号化テクノロジーを使用して、クライアント/サーバ通信を安全化できます。

## UDP

UDP は、トランスポートレイヤのプロトコルです。これは IP の最上部のプロトコルです。UDP は最初にサーバ名解決を行うときに使用される場合があります。その後の接続と通信には TCP が使用されます。

クライアントのブロードキャストへの応答としてデータベースサーバから送信される UDP パケットには機密情報が含まれません。これらのパケットに含まれるデータは次のとおりです。

- データベースサーバ名
- ポート番号
- データベースサーバのバージョン
- データベースサーバで実行されているデータベースの名前

-dh データベースオプションを使用すると、UDP ブロードキャスト応答に対してデータベース名を難読化することができます。

-sb サーバオプションを指定すると、UDP リスナを無効にできます。

## Windows での TCP/IP の使用

Windows 上のデータベースサーバの TCP/IP 実装では、Winsock 2.2 を使用します。

このセクションの内容:

### [TCP/IP を使用したクライアント/サーバ通信の暗号化 \[168 ページ\]](#)

トランスポートレイヤセキュリティ (TLS) を使用して、TCP/IP を介したクライアントアプリケーションとデータベースサーバ間の通信を安全化できます。

### [IPv6 のサポート \[168 ページ\]](#)

IPv6 対応のコンピュータでは、ネットワークデータベースサーバは、デフォルトで IPv4 と IPv6 のすべてのアドレスを受信します。

### [ファイアウォール接続 \[169 ページ\]](#)

クライアントアプリケーションがファイアウォールによってデータベースサーバとは区切られている場合は、接続が制限されます。

### [ダイヤルアップネットワーク接続 \(CommLinks 接続パラメータ\) \[171 ページ\]](#)

ダイヤルアップネットワーク接続経由で接続するには、CommLinks 接続パラメータを使用する必要があります。

### [ネームサーバとして LDAP を使用した接続 \[171 ページ\]](#)

Windows または UNIX プラットフォーム上で動作している場合、中央 LDAP サーバを指定して企業内の全データベースサーバを追跡できます。

### [通信の圧縮設定 \[175 ページ\]](#)

1 つまたはすべての接続について圧縮機能を有効にして、パケット圧縮時の最小サイズを設定すると、パフォーマンスを向上できる場合があります。

## 関連情報

[-dh データベースオプション \[526 ページ\]](#)

[-sb データベースサーバオプション \[468 ページ\]](#)

[-x データベースサーバオプション \[500 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[Broadcast \(BCAST\) プロトコルオプション \[183 ページ\]](#)

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

[MyIP \(ME\) プロトコルオプション \[219 ページ\]](#)

## 1.1.6.2.1 TCP/IP を使用したクライアント/サーバ通信の暗号化

トランスポートレイヤセキュリティ (TLS) を使用して、TCP/IP を介したクライアントアプリケーションとデータベースサーバ間の通信を安全化できます。

デフォルトでは通信パケットが暗号化されないため、セキュリティに関して潜在的な危険があります。トランスポートレイヤセキュリティ (TLS) により、サーバの検証、RSA 暗号化テクノロジーを使用した強力な暗号化、およびデータ整合性を保護するその他の機能が提供されます。

### 関連情報

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

## 1.1.6.2.2 IPv6 のサポート

IPv6 対応のコンピュータでは、ネットワークデータベースサーバは、デフォルトで IPv4 と IPv6 のすべてのアドレスを受信します。

IPv6 は、Windows CE ではサポートされません。

通常、IPv6 を使用するようデータベースサーバの開始行を変更する必要はありません。IP アドレスの指定が必要な場合は、データベースサーバライブラリとクライアントライブラリは、どちらも IPv4 アドレスと IPv6 アドレスを受け付けます。たとえば、コンピュータで複数のネットワークカードを使用できる場合、2 つの IPv4 アドレスと 2 つの IPv6 アドレスが存在することがあります。IPv6 アドレスの場合は、アドレスを角括弧または丸括弧で囲む必要があります。データベースサーバが受信する IPv6 アドレスを 1 つに制限する場合は、アドレスを次のフォーマットで指定できます。

```
dbsrv17 -x tcpip(MyIP=fd77:55f:5a64:52a:202:5445:5245:444f) ...
```

同様に、クライアントアプリケーションでサーバの IP アドレスを指定する必要がある場合は、接続文字列または ODBC データソースに次のフォーマットでアドレスを指定できます。

```
...HOST=fd77:55f:5a64::444f;...
```

各インタフェースにインタフェース識別子が与えられており、IPv6 アドレスの末尾に示されます。たとえば、`ipconfig.exe` にアドレス `fd77:55f:5a64::444f%7` がリストされた場合、インタフェース識別子は 7 です。Windows プラットフォームで IPv6 アドレスを指定する場合、インタフェース識別子を使用する必要があります。UNIX では、インタフェース識別子とインタフェース名のどちらでも指定できます (インタフェース名は、`ifconfig` によってレポートされるインタフェースの名前です)。たとえば、IPv6 アドレス `fd77:55f:5a64::444f%eth1` では、インタフェース名 `eth1` です。Linux (カーネル 2.6.13 以降) で IPv6 アドレスを指定する場合は、インタフェース識別子が必要です。この要件は次によって指定された値に影響します。

- Host 接続パラメータ
- Broadcast (BCAST) プロトコルオプション
- Host (IP) プロトコルオプション (クライアント側のみ)
- MyIP (ME) プロトコルオプション

#### 例

ipconfig.exe に 2 つのインターフェースがリストされており、片方の識別子が 1 でもう片方の識別子が 2 だとします。インターフェース番号 2 によって使用されているネットワーク上のデータベースサーバを検索している場合、クライアントライブラリに対してそのインターフェースにだけブロードキャストするよう指示できます。

```
LINKS=tcpip (BROADCAST=ff02::1%2)
```

IPv6 リンクローカルのマルチキャストアドレスは ff02::1 です。

## 関連情報

[Host 接続パラメータ \[82 ページ\]](#)

[Broadcast \(BCAST\) プロトコルオプション \[183 ページ\]](#)

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

[MyIP \(ME\) プロトコルオプション \[219 ページ\]](#)

### 1.1.6.2.3 ファイアウォール接続

クライアントアプリケーションがファイアウォールによってデータベースサーバとは区切られている場合は、接続が制限されません。

ファイアウォールソフトウェアは、ネットワークポートに従って、ネットワークパケットをフィルタリングする可能性があります。また、一般に、UDP パケットはファイアウォールを通過できません。

通常、接続は、Host 接続パラメータを使用して適切に設定されたファイアウォールを経由し、データベースサーバのアドレスとポートを提供することで実行できます。データベースサーバがデフォルトポート 2638 を使用している場合、ポートは必要ありません。

次の接続文字列は、ポート 2020 を使用するアドレス serverhost のコンピュータ上で実行中の myserver という名前のデータベースサーバに接続します。Host 接続パラメータが TCP/IP アドレスとポートを指定しているため、UDP パケットは使用されません。

```
Server=myserver;Host=serverhost:2020
```

## 特定のクライアントポートのみを許可するファイアウォール

特定のクライアントポートのみを許可するようにファイアウォールを設定する必要がある場合、Host 接続パラメータではなく、CommLinks(LINKS) 接続パラメータを使用する必要があります。次の TCP/IP プロトコルオプションは、CommLinks 接続パラメータを使用するときに必要です。

### Host

このプロトコルオプションには、データベースサーバを実行しているホスト名を設定します。省略形の IP を使用できます。

### ServerPort

データベースサーバがデフォルトポート 2638 を使用していない場合、使用しているポートを指定します。Port と短縮してもかまいません。

### ClientPort

このプロトコルオプションには、使用するクライアントアプリケーションで有効な範囲の値を設定します。CPORT と短縮してもかまいません。

### DoBroadcast=NONE

サーバに接続するときに UDP が使用されないようにするには、このプロトコルオプションを設定します。DOBROAD と短縮してもかまいません。

ファイアウォールは、データベースサーバのアドレスとすべてのクライアントのアドレスとの間の TCP/IP トラフィックを許可するように設定します。データベースサーバのアドレスは、サーバを実行中のコンピュータの IP アドレス (HOST プロトコルオプション) とデータベースサーバの IP ポート番号 (ServerPort プロトコルオプション、デフォルトは 2638) です。

クライアントポートが再び利用されるまでに数分のタイムアウトがあるため、各クライアントコンピュータからの同時接続の最大数よりも多いクライアントポート数の範囲を使用します。ClientPort プロトコルオプションで指定されるクライアントの範囲は、ファイアウォールで許可される範囲と一致する必要があります。

### 例

次の接続文字列のフラグメントは、クライアントアプリケーションをポート 5050 ~ 5060 に制限します。また、サーバポート 2020 を使用するアドレス myhost のコンピュータで実行されているデータベースサーバ myeng に接続します。DoBroadcast プロトコルオプションを NONE に設定しているため、UDP ブロードキャストは実行されません。

```
Server=myeng;LINKS=tcPIP (ClientPort=5050-5060;HOST=myhost;PORT=2020;DoBroadcast=NONE)
```

## 関連情報

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

[ClientPort \(CPORT\) プロトコルオプション \(クライアント側のみ\) \[190 ページ\]](#)

[ServerPort \(PORT\) プロトコルオプション \[225 ページ\]](#)

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

[DoBroadcast \(DOBROAD\) プロトコルオプション \[192 ページ\]](#)

## 1.1.6.2.4 ダイアルアップネットワーク接続 (CommLinks 接続パラメータ)

ダイアルアップネットワーク接続経由で接続するには、CommLinks 接続パラメータを使用する必要があります。

クライアント側では、以下のプロトコルオプションを指定してください。

### Host

Host (IP) プロトコルオプションを使用して、データベースサーバのホスト名または IP アドレスを指定します。

### DoBroadcast

Host (IP) プロトコルオプションを指定した場合は、データベースサーバでブロードキャスト検索を行う必要はありません。このため、ダイレクトブロードキャストを使用してください。

### MyIP

クライアント側では、**MyIP=NONE** に設定してください。

### TIMEOUT

サーバを検索する間のクライアントの待機時間を長くするには、TIMEOUT (TO) プロトコルオプションを設定します。

### 例

次に示すのは、ダイアルアップネットワーク接続経由で接続するために使用する一般的な CommLinks (LINKS) 接続パラメータの例です。

```
LINKS=tcPIP (MyIP=NONE; DoBroadcast=DIRECT; HOST=server-ip); TIMEOUT=15)
```

## 関連情報

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

[DoBroadcast \(DOBROAD\) プロトコルオプション \[192 ページ\]](#)

[MyIP \(ME\) プロトコルオプション \[219 ページ\]](#)

[Timeout \(TO\) プロトコルオプション \[232 ページ\]](#)

## 1.1.6.2.5 ネームサーバとして LDAP を使用した接続

Windows または UNIX プラットフォーム上で動作している場合、中央 LDAP サーバを指定して企業内の全データベースサーバを追跡できます。

データベースサーバ自体を LDAP サーバに登録する場合、クライアントは LDAP サーバに問い合わせでデータベースサーバを検索できます。この場合、サーバが WAN 上、LAN 上、またはファイアウォールの外側にあっても検索できます。LDAP ネームサーバは、Host 接続パラメータまたは HOST プロトコルオプションでデータベースサーバアドレスを指定するクライアントからは使用されません。

このセクションの内容:

#### [saldap.ini ファイル設定 \[172 ページ\]](#)

LDAP をネームサーバとして使用してデータベースを検索するには、LDAP サーバの検索方法と接続方法に関する情報を含むファイルをデータベースサーバコンピュータと各クライアントコンピュータに作成する必要があります。デフォルトで、このファイル名は `saldap.ini` ですが、この名前は変更できます。

#### [IBM AIX の /usr/lib ディレクトリにリンクを作成する \[174 ページ\]](#)

`/usr/lib` ディレクトリにリンクを作成し、LDAP ライブラリが含まれるディレクトリが `LIBPATH` に含まれていることを確認します。

### 1.1.6.2.5.1 `saldap.ini` ファイル設定

LDAP をネームサーバとして使用してデータベースを検索するには、LDAP サーバの検索方法と接続方法に関する情報を含むファイルをデータベースサーバコンピュータと各クライアントコンピュータに作成する必要があります。デフォルトで、このファイル名は `saldap.ini` ですが、この名前は変更できます。

`saldap.ini` ファイルの名前を変更する場合、データベースサーバ接続時には、ファイル名を指定するために LDAP プロトコルオプションを使用する必要があります。このファイルがない場合、LDAP ネームサーバのサポートは無効になっています。

ファイル非表示ユーティリティ (`dbfhide`) を使用して、`saldap.ini` ファイルの内容を暗号化できます。その場合、ファイルの非暗号化コピーを保持し、暗号化ファイルを更新できるようにします。

このファイルは、LDAP プロトコルオプションに完全なパスを指定していない場合は SQL Anywhere 実行プログラム (たとえば、Windows の場合は `%SQLANY17%\bin32`) と同じディレクトリ置く必要があります。このファイルの内容は、次のフォーマットになります。

```
[LDAP]
server=computer-running-LDAP-server
port=port-number-of-LDAP-server
basedn=Base-DN
authdn=Authentication-DN
password=password-for-authdn
search_timeout=age-of-timestamps-to-be-ignored
update_timeout=frequency-of-timestamp-updates
read_authdn=read-only-authentication-domain-name
read_password=password-for-authdn
```

#### **server**

LDAP サーバを実行中のコンピュータの名前または IP アドレス。この値は UNIX で必要です。このエントリが Windows がない場合、Windows がローカルのドメインコントローラーで動作中の LDAP サーバを検索します。

#### **port**

LDAP サーバで使用されるポート番号。デフォルトは 389 です。

#### **basedn**

SQL Anywhere エントリが格納されているサブツリーのドメイン名。この値はデフォルトでツリーのルートになります。

#### **authdn**

認証ドメイン名。ドメイン名は、LDAP ディレクトリにある既存のユーザオブジェクトである必要があります。このディレクトリには `basedn` への書き込みアクセスがあります。このパラメータはデータベースサーバで必要ですが、クライアントでは無視されます。

## password

authdn のパスワード。このパラメータはデータベースサーバで必要ですが、クライアントでは無視されます。

## search\_timeout

タイムスタンプがクライアントまたはサーバ列挙ユーティリティ (dblocate) で無視されるタイムスタンプの経過時間。値 0 はこのオプションを無効にして、すべてのエントリが現在のものと見なされます。デフォルト値は 600 秒 (10 分) です。

## update\_timeout

LDAP ディレクトリ内のタイムスタンプの更新頻度。値として 0 を指定するとこのオプションは無効になり、データベースサーバはタイムスタンプを更新しません。デフォルト値は 120 秒 (2 分) です。

## read\_authdn

読み取り専用の認証ドメイン名。ドメイン名は、LDAP ディレクトリにある既存のユーザオブジェクトを指定します。このディレクトリには basedn への読み取りアクセスがあります。このパラメータが必要なのは、検索の実行前に LDAP サーバで非匿名バインドが必要な場合だけです。たとえば、Active Directory が LDAP サーバとして使用されている場合、通常はこのフィールドが必要です。このパラメータが指定されていない場合、バインドは匿名になります。

## read\_password

authdn のパスワード。クライアントにこのパラメータが必要なのは、read\_authdn パラメータが指定されている場合だけです。

## 接続の確立方法

データベースサーバは、起動時に、LDAP 内で同じ名前を持つ既存のエントリがないかどうかを確認します。一致するエントリが見つかった場合、LDAP のロケーションエントリと起動しようとしているデータベースサーバとが一致した場合、または LDAP エントリのタイムスタンプフィールドが search\_timeout パラメータで指定されている時間よりも長い場合に、エントリは置き換えられます。

あるデータベースサーバが起動しようとするときに同じ名前で行中の別のデータベースサーバが存在する場合、データベースは起動できません。

LDAP のエントリを確実に最新ののものにするために、データベースサーバは LDAP エントリ内のタイムスタンプフィールドを 2 分ごとに更新します。エントリのタイムスタンプが 10 分以上古い場合、クライアントは LDAP エントリを無視します。これらの設定はいずれも変更可能です。

クライアントでは、クライアントが UDP ブロードキャストを実行する前に LDAP ディレクトリが検索されるので、データベースサーバが見つかるまで UDP ブロードキャストは送信されません。LDAP 検索は非常に高速なので、失敗しても認識できるほどの遅延は発生しません。

## LDAP サーバとサーバ列挙ユーティリティ (dblocate)

サーバ列挙ユーティリティ (dblocate) では LDAP も使用されます。LDAP にリストされているすべてのデータベースサーバが、返されたデータベースサーバのリストに追加されます。これにより、サーバ列挙ユーティリティ (dblocate) が、たとえばブロードキャストが到達しなかったデータベースサーバなど、通常どおりに返されなかったサーバをリストします。10 分以上古いタイムスタンプを持つエントリは含まれません。

## 例

次に、`saldap.ini` ファイルの例を示します。

```
[LDAP]
server=ldapserver
basedn=dc=sap,dc=com
authdn=cn=SAserver,ou=iAnywhereASA,dc=sap,dc=com
password=secret
```

エントリが `iAnywhereASA` と呼ばれる `basedn` のサブツリーに保管されます。このエントリは SQL Anywhere が LDAP を使用できるようになる前に作成します。サブツリーを作成するには、LDAPADD ユーティリティを使用します。次のような情報を提供します。

```
dn: ou=iAnywhereASA,basedn
objectClass: organizationalUnit
objectClass: top
ou: iAnywhereASA
```

## 関連情報

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

[サーバ列挙ユーティリティ \(dblocate\) \[1145 ページ\]](#)

## 1.1.6.2.5.2 IBM AIX の `/usr/lib` ディレクトリにリンクを作成する

`/usr/lib` ディレクトリにリンクを作成し、LDAP ライブラリが含まれるディレクトリが `LIBPATH` に含まれていることを確認します。

## 前提条件

LDAP は TCP/IP とともに使用し、ネットワークデータベースサーバでのみ使用されます。

## コンテキスト

サーバ列挙ユーティリティ (dblocate) は LDAP サーバを使用してそのようなサーバを検索できます。

## 手順

1. ルートユーザで次のコマンドを実行して /usr/lib ディレクトリにリンクを作成します。

```
cd /usr/lib
ln -s /opt/IBM/ldap/V6.1/lib64/libibmldap.a libibmldap64.a
ln -s /opt/IBM/ldap/V6.1/lib/libibmldap.a
```

2. (省略可) LDAP ライブラリのあるディレクトリが LIBPATH に含まれていることを確認します。

32ビットライブラリの場合:

```
export LIBPATH=/opt/IBM/ldap/V6.1/lib:$LIBPATH
```

64ビットライブラリの場合:

```
export LIBPATH=/opt/IBM/ldap/V6.1/lib64:$LIBPATH
```

## 結果

/usr/lib にリンクが作成されます。

## 関連情報

[LDAP プロトコルオプション \[205 ページ\]](#)

[サーバ列挙ユーティリティ \(dblocate\) \[1145 ページ\]](#)

### 1.1.6.2.6 通信の圧縮設定

1つまたはすべての接続について圧縮機能を有効にして、パケット圧縮時の最小サイズを設定すると、パフォーマンスを向上できる場合があります。

そのため、圧縮スレッシュホールドを変更することが有益かどうかを判断するために、ネットワーク上でアプリケーションとのパフォーマンス分析を実行する必要があります。

圧縮機能を有効にすると、データパケットに格納される情報量が増大し、特定のデータセットの送信に必要なパケット数が減少します。パケット数を減らすと、データを高速で送信できます。

圧縮の有効化が特定の状況で有益であるかどうかを判断するには、通信の圧縮を運用環境で使用する前に、アプリケーションを使用しているネットワークでパフォーマンス分析を行います。

次の項目によって圧縮のパフォーマンスをチューニングできます。

#### 圧縮の有効化

次のような状況では、1つ (またはすべて) の接続で圧縮を有効化すると、パフォーマンスが大幅に向上する可能性があります。

- 一部の無線ネットワーク、モデム、シリアルリンク、WAN などの低速ネットワークで使用するとき。
- 圧縮機能が組み込まれている低速ネットワークで SQL Anywhere 暗号化で使用するとき。これは、パケットが圧縮されてから暗号化されるためです。

ただし、圧縮の有効化は、パフォーマンス低下の原因になる可能性もあります。次に例を示します。

- 通信の圧縮には、より多くのメモリと CPU が使用されます。このため、特に LAN やその他の高速ネットワークを使用する場合に、パフォーマンスが低下することがあります。
- ほとんどのモデムと一部の低速ネットワークに、すでに圧縮機能が組み込まれます。この場合、通信圧縮を行っても、データの暗号化を同時に実行しないかぎり、パフォーマンスはほとんど向上しません。

圧縮のスレッシュホールドの変更

### i 注記

ほとんどのネットワークでは、圧縮のスレッシュホールドを変更する必要はありません。まれではありますが、パフォーマンスは、圧縮のスレッシュホールドを調整することでも向上できる可能性があります。

圧縮スレッシュホールドの変更が特定の状況で有益であるかどうかを判断するには、運用環境で調整を行う前に、アプリケーションを使用しているネットワークでパフォーマンス分析を行います。

圧縮が有効な場合、パケットは、各々のサイズに応じて圧縮するかどうかを決定します。圧縮のスレッシュホールドよりも小さなパケットは、通信圧縮が有効になっている場合でも圧縮されません。

パケットの圧縮には CPU 時間が必要なため、小さなパケットを圧縮しようとするだけで、実際にはパフォーマンスが低下することもあります。

一般に、圧縮のスレッシュホールド値を小さくすると、非常に低速なネットワークではパフォーマンスが向上し、値を大きくすると CPU 使用率の減少によってパフォーマンスが向上する場合があります。ただし、圧縮のスレッシュホールド値を小さくするとクライアントとサーバの両方で CPU 使用率が増加するので、パフォーマンス分析を行って、圧縮のスレッシュホールドを変更することでパフォーマンスが向上するかどうかを判断してください。

SQL Anywhere 圧縮設定の調整

- 通信の圧縮を有効にします。  
高度に圧縮可能なデータを大きなパケットサイズで大量にデータ転送することで、最高の圧縮率を得ることができます。
- CompressionThreshold 設定を調整します。  
圧縮スレッシュホールドの値を小さくすると、非常に低速なネットワークではパフォーマンスが向上し、値を大きくすると CPU 使用率の減少によってパフォーマンスが向上する場合があります。

## 関連情報

[Compress \(COMP\) 接続パラメータ \[59 ページ\]](#)

[-pc データベースサーバオプション \[457 ページ\]](#)

[Compress \(COMP\) 接続パラメータ \[59 ページ\]](#)

[-pc データベースサーバオプション \[457 ページ\]](#)

[CompressionThreshold \(COMPTH\) 接続パラメータ \[60 ページ\]](#)

[-pt データベースサーバオプション \[461 ページ\]](#)

## 1.1.7 ネットワークプロトコルオプション

ネットワークプロトコルオプションを使用すると、異なるネットワークプロトコルの実装の問題を回避できます。

サーバ側	クライアント側
<p>データベースサーバの場合、<code>-x tcpip</code> オプションを使用してネットワーク通信プロトコルオプションを指定します。次に例を示します。</p> <pre>dbsrv17 -x tcpip (PARAM1=value1;PARAM2=value2;...)</pre>	<p>クライアントアプリケーションの場合、CommLinks (LINKS) 接続パラメータを使用してネットワークプロトコルオプションを指定します。</p> <pre>CommLinks=tcpip (PARAM1=value1;PARAM2=value2;...)</pre>
<p>データベースサーバの場合、<code>-ec tls</code> オプションを使用してネットワークトランスポートレイヤセキュリティ暗号化プロトコルを指定します。次に例を示します。</p> <pre>dbsrv17 -ec tls (PARAM1=value1;PARAM2=value2;...)</pre>	<p>クライアントアプリケーションの場合、Encryption (ENC) 接続パラメータを使用してトランスポートレイヤセキュリティ暗号化オプションを指定します。</p> <pre>Encryption=tls (PARAM1=value1;PARAM2=value2;...)</pre>
<p>データベースサーバの場合、<code>-xs http</code>、<code>-xs https</code>、<code>-xs odata</code> オプションを使用して web サーバプロトコルオプションを指定します。次に例を示します。</p> <pre>dbsrv17 -xs http (PARAM1=value1;PARAM2=value2;...)</pre>	

オプション値にスペースが含まれる場合、ネットワークプロトコルオプションを二重引用符で囲み、システムコマンドインタプリタによって正しく解析されるようにします。次に例を示します。

```
dbisql -c "host=localhost;server=demo;encryption=tls (fips=yes;certificate_name=RSA
Server;certificate_company=SAP;certificate_unit=SQL
Anywhere;trusted_certificates=rsaroot.crt)"
```

UNIX では、セミコロンがコマンドの区切り文字として解釈されるため、複数の接続パラメータやプロトコルオプションを指定する場合にも二重引用符が必要です。

ブールオプションは、YES、Y、ON、TRUE、T、1 のいずれかによってオンになり、NO、N、OFF、FALSE、F、0 のいずれかによってオフになります。プロトコルオプション名では、大文字と小文字が区別されません。次に例を示します。

```
dbisql -c "Host=localhost;Server=demo;Encryption=TLS (FIPS=1;Certificate_Name=RSA
Server;Certificate_Company=SAP;Certificate_Unit=SQL
Anywhere;Trusted_Certificates=rsaroot.crt)"
```

プロトコルオプションを設定ファイルに記述し、サーバオプションの `@data` を使用して設定ファイルを呼び出すこともできます。

### i 注記

CommLinks (LINKS) 接続パラメータではなく Host 接続パラメータを使用します。CommLinks (LINKS) 接続パラメータは、Host または ServerPort (PORT) 以外の TCP/IP オプションを指定する必要がある場合にのみ使用してください。

接続文字列に CommLinks と Host の両方は指定できません。

TCP/IP 接続に CommLinks (LINKS) 接続パラメータを使用する場合は、ServerName (Server) 接続パラメータを使用してサーバ名を指定します。

クライアントととして同じコンピュータ上のデータベースサーバに接続する場合は、共有メモリをおすすめします。

TCPIP	HTTP	HTTPS	TLS	OData
Broadcast (BCAST) protocol option	DatabaseName (DBN) protocol option	DatabaseName (DBN) protocol option	certificate_company protocol option (client side only)	ExitOnError (EXIT) protocol option
BroadcastListener (BLISTENER) protocol option (server side only)	KeepaliveTimeout (KTO) protocol option	FIPS protocol option	certificate_name protocol option (client side only)	LogFile (LOG) protocol option
ClientPort (CPORT) protocol option (client side only)	LocalOnly (LO) protocol option	Identity protocol option	certificate_unit protocol option (client side only)	LogVerbosity protocol option
DoBroadcast (DOBROAD) protocol option	LogFile (LOG) protocol option	Identity_Password protocol option	FIPS protocol option	MyIP protocol option
Host (IP) protocol option (client side only)	LogFormat (LF) protocol option	KeepaliveTimeout (KTO) protocol option	Identity protocol option (server side only)	QuietConsole (QUIET) protocol option
LDAP protocol option	LogMaxSize (LSIZE) protocol option	LocalOnly (LO) protocol option	Identity_Password protocol option (server side only)	SecureMyIP protocol option
LocalOnly (LO) protocol option	LogOptions (LOPT) protocol option	LogFile (LOG) protocol option	trusted_certificates protocol option (client side only)	SecureServerPort protocol option
MyIP (ME) protocol option	LogRename (LRENAME) protocol option	LogFormat (LF) protocol option	skip_certificate_name_check option (client side only)	ServerPort (PORT) protocol option
ReceiveBufferSize (RCVBUFSZ) protocol option	MaxConnections (MAXCONN) protocol option	LogMaxSize (LSIZE) protocol option	allow_expired_certs option	SSLKeyStore (KEYSTORE) protocol option
SendBufferSize (SNDBUFSZ) protocol option	MaxRequestSize (MAXSIZE) protocol option	LogOptions (LOPT) protocol option		SSLKeyStorePassword (KEYSTOREPASSWORD) protocol option
ServerPort (PORT) protocol option	MaxRequestVars (MAXVARS) protocol option	MaxConnections (MAXCONN) protocol option		
TDS protocol option (server side only)	MyIP (ME) protocol option	MaxRequestSize (MAXSIZE) protocol option		
Timeout (TO) protocol option	ServerPort (PORT) protocol option	MaxRequestVars (MAXVARS) protocol option		
VerifyServerName (VERIFY) protocol option (client side only)	Timeout (TO) protocol option	MyIP (ME) protocol option		
		ServerPort (PORT) protocol option		

TCPIP	HTTP	HTTPS	TLS	OData
		Timeout (TO) protocol option		
		allow_expired_certs option		

このセクションの内容:

[allow\\_expired\\_certs プロトコルオプション \[181 ページ\]](#)

期限切れになったかまだ有効になっていない ID ファイルまたはサーバ証明書を受け入れます。

[Broadcast \(BCAST\) プロトコルオプション \[183 ページ\]](#)

サーバの検索に使用する IP ブロードキャストアドレスを指定します。

[BroadcastListener \(BLISTENER\) プロトコルオプション \(サーバ側のみ\) \[184 ページ\]](#)

指定したポートのブロードキャスト受信を制御します。

[certificate\\_company プロトコルオプション \(クライアント側のみ\) \[186 ページ\]](#)

証明書に記されている組織フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

[certificate\\_name プロトコルオプション \(クライアント側のみ\) \[187 ページ\]](#)

証明書に記されている通称フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

[certificate\\_unit プロトコルオプション \(クライアント側のみ\) \[188 ページ\]](#)

証明書に記されている Organization Unit フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

[ClientPort \(CPORT\) プロトコルオプション \(クライアント側のみ\) \[190 ページ\]](#)

TCP/IP を使用してクライアントアプリケーションが通信するポート番号を指定します。

[DatabaseName \(DBN\) プロトコルオプション \[191 ページ\]](#)

Web 要求を処理するときに使用するデータベースの名前を指定します。また、REQUIRED や AUTO キーワードを使用して URL の一部としてデータベース名が必要かどうかを指定します。

[DoBroadcast \(DOBROAD\) プロトコルオプション \[192 ページ\]](#)

クライアントがデータベースサーバを検索する方法と、データベースサーバが起動時にブロードキャストを実行するかどうかを制御します。

[ExitOnError \(EXIT\) プロトコルオプション \[194 ページ\]](#)

起動時にエラーが発生した場合に OData サーバを停止するかどうかを制御します。

[FIPS プロトコルオプション \[195 ページ\]](#)

データベースクライアント/サーバ (TLS) や web サーバ (HTTPS) の通信の暗号化に、FIPS 認定 (FIPS PUB 140-2) のセキュリティアルゴリズムを使用できます。

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

データベースサーバのアドレスを指定します。

[SecureMyIP プロトコルオプション \[200 ページ\]](#)

データベースサーバが保護された接続を受信するネットワークインタフェースを指定します。

[identity プロトコルオプション \[201 ページ\]](#)

ID ファイルの名前を指定します。

#### [identity\\_password プロトコルオプション \[202 ページ\]](#)

暗号化証明書のパスワードを指定します。

#### [KeepaliveTimeout \(KTO\) プロトコルオプション \[204 ページ\]](#)

HTTP 要求間の最大アイドル時間を秒単位で指定します。

#### [LDAP プロトコルオプション \[205 ページ\]](#)

クライアントは、IP アドレスを指定せずにデータベースサーバを検索できます。

#### [LocalOnly \(LO\) プロトコルオプション \[206 ページ\]](#)

データベースサーバが接続をローカルコンピュータのみに制限できるようにします。

#### [LogFile \(LOG\) プロトコルオプション \[208 ページ\]](#)

データベースサーバが Web 要求に関する情報を書き込むファイル名を指定します。

#### [LogFormat \(LF\) プロトコルオプション \[209 ページ\]](#)

データベースサーバがメッセージログファイルに書き込む Web 要求に関する情報メッセージのフォーマットを制御し、メッセージに表示されるフィールドを指定します。

#### [LogMaxSize \(LSIZE\) プロトコルオプション \[211 ページ\]](#)

データベースサーバが Web 要求に関する情報を書き込むメッセージログファイルの最大サイズを制御します。

#### [LogOptions \(LOPT\) プロトコルオプション \[212 ページ\]](#)

データベースサーバが Web 要求に関する情報を書き込むログに記録されるメッセージタイプを指定します。

#### [LogRename \(LRENAME\) プロトコルオプション \[214 ページ\]](#)

ユーザが以前の HTTP デバッグログファイルの名称を変更できます。

#### [LogVerbosity プロトコルオプション \[215 ページ\]](#)

ログの冗長性レベルを指定します。

#### [MaxConnections \(MAXCONN\) プロトコルオプション \[216 ページ\]](#)

データベースサーバで許可される HTTP/HTTPS 同時接続の最大数を指定します。

#### [MaxRequestSize \(MAXSIZE\) プロトコルオプション \[217 ページ\]](#)

データベースサーバで許可できる最大要求サイズを指定します。

#### [MaxRequestVars \(MAXVARS\) プロトコルオプション \[218 ページ\]](#)

データベースサーバに送信される要求内で許可する HTTP 入力変数の最大数を指定します。

#### [MyIP \(ME\) プロトコルオプション \[219 ページ\]](#)

データベースサーバまたはクライアントコンピュータにネットワークインタフェースコントローラ (NIC) またはアダプタが複数存在する場合にどれを使用するかを細かく制御するには、このオプションを使用します。

#### [QuietConsole \(QUIET\) プロトコルオプション \[221 ページ\]](#)

サーバメッセージウィンドウが最小化された状態で起動するかどうかを OData サーバに指示します。

#### [ReceiveBufferSize \(RCVBUFSZ\) プロトコルオプション \[222 ページ\]](#)

TCP/IP プロトコルスタックが使用する受信バッファのサイズを設定します。

#### [SecureServerPort プロトコルオプション \[223 ページ\]](#)

OData サーバが保護された (HTTPS) 接続を受信するポートを指定します。

#### [SendBufferSize \(SNDBUFSZ\) プロトコルオプション \[224 ページ\]](#)

TCP/IP プロトコルスタックが使用する送信バッファのサイズを設定します。

#### [ServerPort \(PORT\) プロトコルオプション \[225 ページ\]](#)

データベースサーバが受信しているポートを指定します。

#### [skip\\_certificate\\_name\\_check](#) プロトコルオプション (クライアント側のみ) [228 ページ]

クライアントライブラリがサーバホスト名とデータベースサーバの証明書ホスト名との照合を省略するかどうかを制御します。

#### [SSLKeyStore \(KEYSTORE\)](#) プロトコルオプション [229 ページ]

OData サーバが通信を暗号化するために使用する SSL 証明書が含まれる Java Key Store のファイルパスを指定します。

#### [SSLKeyStorePassword \(KEYSTOREPASSWORD\)](#) プロトコルオプション [230 ページ]

SSLKeyStore (KEYSTORE) プロトコルオプションによって識別された Java キーストアに対して認証するために OData サーバが使用するパスワードを指定します。

#### [TDS](#) プロトコルオプション (サーバ側のみ) [231 ページ]

データベースサーバへの TDS 接続が許可されるかどうかを制御します。

#### [Timeout \(TO\)](#) プロトコルオプション [232 ページ]

通信 (TCP/IP) 確立時の応答の待ち時間を秒単位で指定します。また、要求 (HTTP または HTTPS) の待ち時間を秒単位で指定します。

#### [trusted\\_certificates](#) プロトコルオプション [234 ページ]

信頼できる証明書を 1 つ以上含むファイルのパスとファイル名を指定します。

#### [VerifyServerName \(VERIFY\)](#) プロトコルオプション (クライアント側のみ) [236 ページ]

クライアントが接続前にデータベースサーバ名を確認する必要があるかどうかを制御します。

## 関連情報

[通信プロトコルオプションの考慮事項](#) [164 ページ]

[CommLinks \(LINKS\) 接続パラメータ](#) [57 ページ]

[Encryption \(ENC\) 接続パラメータ](#) [77 ページ]

[Host 接続パラメータ](#) [82 ページ]

[ServerName \(Server\) 接続パラメータ](#) [105 ページ]

[-xs データベースサーバオプション](#) [507 ページ]

### 1.1.7.1 allow\_expired\_certs プロトコルオプション

期限切れになったかまだ有効になっていない ID ファイルまたはサーバ証明書を受け入れます。

#### 構文

```
allow_expired_certs={ YES | NO }
```

## 使用可能なプロトコル

HTTPS、TLS

## デフォルト

NO

## 備考

データベースサーバにこのオプションが設定されると、データベースサーバは期限切れになったかまだ有効になっていない ID ファイルを受け入れます。

クライアントにこのオプションが設定されると、クライアントライブラリはルート証明書を信頼し、期限切れになったかまだ有効になっていないデータベースサーバ証明書を受け入れます。

### 例

次の例は、FIPS 認定 TLS 暗号アルゴリズムと ID 証明書 `rsaserver.id` を使用するデータベースサーバを起動します。`rsaserver.id` が期限切れの場合でも、データベースサーバにより受け入れられます。

```
dbsrv17 -ec TLS (FIPS=ON; IDENTITY=rsaserver.id; IDENTITY_PASSWORD=test;
ALLOW_EXPIRED_CERTS=ON) -x tcpip mydemo.db
```

次の接続文字列フラグメントは、TLS (RSA 暗号) と、期限切れの場合でも受け入れられる指定した証明書ファイルを使用して、データベースサーバに接続します。

```
"HOST=myhost; SERVER=myserver; ENCRYPTION=TLS (FIPS=OFF; TRUSTED_CERTIFICATE=rsaroot.c
rt; ALLOW_EXPIRED_CERTS=ON) "
```

## 関連情報

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

## 1.1.7.2 Broadcast (BCAST) プロトコルオプション

サーバの検索に使用する IP ブロードキャストアドレスを指定します。

### 構文

```
{ Broadcast | BCAST }=ip-address
```

### 使用可能なプロトコル

TCP/IP

### 指定可能な値

**ip-address**

この文字列は IP アドレス形式で指定する必要があります。

### デフォルト

同一サブネット上のすべてのアドレスにブロードキャストします。

### 備考

#### データベースサーバ

データベースサーバは、起動時に UDP ブロードキャストパケットを送信し、同じサーバ名を持つサーバが他に存在しないかどうかを確認します。このオプションは、その際に使用されるブロードキャストアドレスを指定します。

#### クライアント

クライアントアプリケーションがサーバを探るとき、このオプションは、サーバの検出に使われる IP ブロードキャストアドレスを指定します。

デフォルトのブロードキャストアドレスは、ローカル IP アドレスとサブネットマスクを使用して作成されます。サブネットマスクは、IP アドレスのどの部分がネットワークを指定し、どの部分がホストを指定するかを示します。

たとえば、マスクが 255.255.255.0 のサブネット 10.24.98.x の場合、デフォルトのブロードキャストアドレスは 10.24.98.255 になります。

Windows プラットフォームで IPv6 アドレスを指定する場合、インタフェース識別子を使用する必要があります。UNIX プラットフォームでは、IPv6 アドレスのインタフェース識別子とインタフェース名の両方がサポートされます。Linux (カーネル 2.6.13 以降) では、インタフェース識別子が必要です。

#### 例

次の接続文字列の例は、IPv6 を使用する場合にインタフェース番号 2 だけでブロードキャストするようクライアントに指示します。

```
LINKS=tcpip (BROADCAST=ff02::1%2)
```

次の接続文字列の例は、クライアントに対して、IPv4 を使用するとき 10.24.98.0 から 10.24.98.254 までの範囲ですべての IP アドレスにブロードキャストするように通知します。

```
LINKS=tcpip (BROADCAST=10.24.98.255)
```

## 関連情報

[トラブルシューティング: Broadcast Repeater ユーティリティによるデータベースサーバの検索方法 \[254 ページ\]](#)

[IPv6 のサポート \[168 ページ\]](#)

[BroadcastListener \(BLISTENER\) プロトコルオプション \(サーバ側のみ\) \[184 ページ\]](#)

[DoBroadcast \(DOBROAD\) プロトコルオプション \[192 ページ\]](#)

### 1.1.7.3 BroadcastListener (BLISTENER) プロトコルオプション (サーバ側のみ)

指定したポートのブロードキャスト受信を制御します。

#### 構文

```
{ BroadcastListener | BLISTENER }={ YES | NO }
```

## 使用可能なプロトコル

TCP/IP (サーバ側のみ)

## デフォルト

YES

## 備考

このオプションを使用して、このポートのブロードキャスト受信をオフに設定できます。

`-sb 0`を指定することは、TCP/IP で `BroadcastListener=NO` に設定するのと同じことです。

ブロードキャスト受信が OFF の場合、データベースサーバは UDP ブロードキャストに応答しません。クライアントは、Host 接続パラメータか HOST プロトコルオプションを使用して、データベースサーバのホスト名または IP アドレスを指定するか、データベースサーバを LDAP に登録し、クライアントで LDAP を使用してデータベースサーバを検索する必要があります。また、`dblocate` ユーティリティの出力にはデータベースサーバが含まれません。

### 例

次の例は、TCP/IP 接続に対応するサーバのうち、Host 接続パラメータまたは HOST プロトコルオプションを使用した TCP/IP 接続を必要とするデータベースサーバを起動します (サーバの検索に LDAP を使用しない場合)。

```
dbsrv17 -x tcpip(BroadcastListener=no) -n myserver ...
```

次の例には、データベースサーバに接続するためのホスト名とサーバ名を指定するクライアント接続文字列が含まれています。

```
dbisql -c "Host=myhost;Server=myserver;UID=DBA;PWD=passwd"
```

データベースサーバのコマンドラインに `BroadcastListener=yes` が指定されていれば、サーバ名、ユーザ ID、パスワードを指定するだけで接続できます。

## 関連情報

[-sb データベースサーバオプション \[468 ページ\]](#)

[Broadcast \(BCAST\) プロトコルオプション \[183 ページ\]](#)

[DoBroadcast \(DOBROAD\) プロトコルオプション \[192 ページ\]](#)

[サーバ列挙ユーティリティ \(dblocate\) \[1145 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

## 1.1.7.4 certificate\_company プロトコルオプション (クライアント側のみ)

証明書に記されている組織フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

### 構文

```
certificate_company = organization
```

### 使用可能なプロトコル

HTTPS、TLS

### デフォルト

なし

### 備考

このオプションを使用すると、証明書の識別情報部分にある組織フィールドが、指定した値と一致するかどうかを確認できます。

TLS または HTTPS 接続を開始する際、クライアントライブラリは、RFC 2818 に記述された手順を使用して、データベースサーバのホスト名をサーバが提供する証明書と照らし合わせます。この照合が行われるのは、certificate\_name、certificate\_company、certificate\_unit のいずれのオプションも指定されていない場合、あるいは、skip\_certificate\_name\_check オプションが有効でない場合です。certificate\_name、certificate\_company、certificate\_unit のいずれかが指定されていれば、それらのオプションだけが確認されます。skip\_certificate\_name\_check オプションを有効にすると、ホスト名のチェックが無効になります。

ホスト名や IP アドレスは、subjectAltName (サブジェクトの別名、SAN) 拡張、および Common Name (CN) フィールドから取得します。SAN には、ワイルドカードによって複数のホスト名が指定されている場合があります。たとえば、Google 社の証明書には、\*.google.com、\*.google.ca、\*.android.com が指定されているかもしれません。その場合、www.google.ca は有効なホスト名ということになります。

HTTPS は、Web サービスのクライアントプロシージャだけでサポートされています。

## 例

次のコマンドは、トランスポートレイヤセキュリティを使用して、Interactive SQL とデータベースサーバ demo とを接続します。

```
dbisql -c "UID=DBA;PWD=sql;Server=demo;Host=localhost;Encryption=
  TLS(fips=yes;trusted_certificate=rsaroot.crt;certificate_name=RSA
  Server;certificate_company=SAP;certificate_unit=SQL Anywhere)"
```

## 関連情報

[SQL Anywhere クライアント/サーバ通信の暗号化 \[1653 ページ\]](#)

[certificate\\_name プロトコルオプション \(クライアント側のみ\) \[187 ページ\]](#)

[certificate\\_unit プロトコルオプション \(クライアント側のみ\) \[188 ページ\]](#)

[trusted\\_certificates プロトコルオプション \[234 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

## 1.1.7.5 certificate\_name プロトコルオプション (クライアント側のみ)

証明書に記されている通称フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

### 構文

```
certificate_name=common-name
```

## 使用可能なプロトコル

HTTPS、TLS

## デフォルト

なし

## 備考

このオプションを使用すると、証明書の識別情報部分にある通称フィールドが、指定した値と一致するかどうかを確認できます。

TLS または HTTPS 接続を開始する際、クライアントライブラリは、RFC 2818 に記述された手順を使用して、データベースサーバのホスト名をサーバが提供する証明書と照らし合わせます。この照合が行われるのは、certificate\_name、certificate\_company、certificate\_unit のいずれのオプションも指定されていない場合、あるいは、skip\_certificate\_name\_check オプションが有効でない場合です。certificate\_name、certificate\_company、certificate\_unit のいずれかが指定されていれば、それらのオプションだけが確認されます。skip\_certificate\_name\_check オプションを有効にすると、ホスト名のチェックが無効になります。

ホスト名や IP アドレスは、subjectAltName (サブジェクトの別名、SAN) 拡張、および Common Name (CN) フィールドから取得します。SAN には、ワイルドカードによって複数のホスト名が指定されている場合があります。たとえば、Google 社の証明書には、\*.google.com、\*.google.ca、\*.android.com が指定されているかもしれません。その場合、www.google.ca は有効なホスト名ということになります。

HTTPS は、Web サービスのクライアントプロシージャだけでサポートされています。

### 例

次のコマンドは、トランスポートレイヤセキュリティを使用して、Interactive SQL とデータベースサーバ demo とを接続します。

```
dbisql -c "UID=DBA;PWD=sql;Server=demo;Host=localhost;Encryption=
  TLS(fips=yes;trusted_certificate=rsaroot.crt;certificate_name=RSA
  Server;certificate_company=SAP;certificate_unit=SQL Anywhere) "
```

## 関連情報

[SQL Anywhere クライアント/サーバ通信の暗号化 \[1653 ページ\]](#)

[certificate\\_company プロトコルオプション \(クライアント側のみ\) \[186 ページ\]](#)

[certificate\\_unit プロトコルオプション \(クライアント側のみ\) \[188 ページ\]](#)

[trusted\\_certificates プロトコルオプション \[234 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

### 1.1.7.6 certificate\_unit プロトコルオプション (クライアント側のみ)

証明書に記されている Organization Unit フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

#### 構文

```
certificate_unit=organization-unit
```

## 使用可能なプロトコル

HTTPS、TLS

## デフォルト

なし

## 備考

このオプションを使用すると、証明書の識別情報部分にある Organization Unit フィールドが、指定した値と一致するかどうかを確認できます。

TLS または HTTPS 接続を開始する際、クライアントライブラリは、RFC 2818 に記述された手順を使用して、データベースサーバのホスト名をサーバが提供する証明書と照らし合わせます。この照合が行われるのは、certificate\_name、certificate\_company、certificate\_unit のいずれのオプションも指定されていない場合、あるいは、skip\_certificate\_name\_check オプションが有効でない場合です。certificate\_name、certificate\_company、certificate\_unit のいずれかが指定されていれば、それらのオプションだけが確認されます。skip\_certificate\_name\_check オプションを有効にすると、ホスト名のチェックが無効になります。

ホスト名や IP アドレスは、subjectAltName (サブジェクトの別名、SAN) 拡張、および通称 (CN) フィールドから取得します。SAN には、ワイルドカードによって複数のホスト名が指定されている場合があります。たとえば、Google 社の証明書には、\*.google.com、\*.google.ca、\*.android.com が指定されているかもしれません。その場合、www.google.ca は有効なホスト名ということになります。

HTTPS は、Web サービスのクライアントプロシージャだけでサポートされています。

### 例

次のコマンドは、トランスポートレイヤセキュリティを使用して、Interactive SQL とサーバ demo とを接続します。

```
dbisql -c "UID=DBA;PWD=sql;Server=demo;Host=localhost;Encryption=
  TLS(fips=yes;trusted_certificate=rsaroot.crt;certificate_name=RSA
  Server;certificate_company=test;certificate_unit=test) "
```

## 関連情報

[SQL Anywhere クライアント/サーバ通信の暗号化 \[1653 ページ\]](#)

[certificate\\_company プロトコルオプション \(クライアント側のみ\) \[186 ページ\]](#)

[certificate\\_name プロトコルオプション \(クライアント側のみ\) \[187 ページ\]](#)

[trusted\\_certificates プロトコルオプション \[234 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

## 1.1.7.7 ClientPort (CPORT) プロトコルオプション (クライアント側のみ)

TCP/IP を使用してクライアントアプリケーションが通信するポート番号を指定します。

### 構文

```
{ ClientPort | CPORT }=port-number
```

### 使用可能なプロトコル

TCP/IP (クライアント側のみ)

### デフォルト

ネットワークの実装によって、接続ごとに動的に割り当てられます。ファイアウォールの制限がない場合は、このプロトコルオプションを使用しないようお奨めします。

### 備考

このオプションは、ファイアウォールを介した接続のために提供されています。ファイアウォールソフトウェアは、TCP/UDP ポートに従ってフィルタします。ファイアウォールの理由によって必要な場合以外は、このプロトコルオプションを使用しないようお奨めします。

ClientPort オプションは、クライアントアプリケーションが TCP/IP を使って通信するポート番号を指定します。単一のポート番号、または個々のポート番号の組み合わせやポート番号の範囲を指定できます。次に例を示します。

- (CPORT=1234)
- (CPORT=1234,1235,1239)
- (CPORT=1234-1238)
- (CPORT=1234-1237,1239,1242)

指定されたデータソースや接続文字列を使用して複数の接続を確立するには、ポート番号のリストや範囲を指定することをお奨めします。ポート番号を 1 つだけ指定すると、アプリケーションが維持できるのは、一度に 1 つの接続のみとなります。また、1 つの接続を閉じた後は、数分のタイムアウト時間が生じます。その間、指定されたポートを使って新しい接続は作成できません。ポート番号のリストや範囲を指定すると、アプリケーションは、いずれかのポート番号との接続が確立するまで、試行を続けます。

## 例

次の接続文字列フラグメントは、ポート 6000 を使用するアプリケーションから、ポート 5000 を使用する my-server という名前のサーバへの接続を確立します。

```
CommLinks=tcPIP(ClientPort=6000;ServerPort=5000);ServerName=my-server
```

次の接続文字列フラグメントは、指定されたポートを使用できるアプリケーションから、デフォルトのサーバポートを使用する my-server という名前のサーバに通信する接続を確立します。

```
CommLinks=tcPIP(ClientPort=5040,5050-5060,5070);  
ServerName=my-server
```

## 関連情報

[通信プロトコル \[164 ページ\]](#)

[ファイアウォール接続 \[169 ページ\]](#)

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

[DoBroadcast \(DOBROAD\) プロトコルオプション \[192 ページ\]](#)

[ServerPort \(PORT\) プロトコルオプション \[225 ページ\]](#)

## 1.1.7.8 DatabaseName (DBN) プロトコルオプション

Web 要求を処理するときに使用するデータベースの名前を指定します。また、REQUIRED や AUTO キーワードを使用して URL の一部としてデータベース名が必要かどうかを指定します。

### 構文

```
{ DatabaseName | DBN }={ AUTO | REQUIRED | database-name }
```

## 使用可能なプロトコル

HTTP、HTTPS

## デフォルト

AUTO

## 備考

このプロトコルオプションが REQUIRED に設定されている場合は、URL がデータベース名を指定します。

このプロトコルオプションが AUTO に設定されている場合は、URL がデータベース名を指定できますが、必須ではありません。URL にデータベース名が含まれていない場合は、サーバでのデフォルトのデータベースを Web 要求の処理に使用します。AUTO に設定されている場合、サーバは URL にデータベース名が含まれているかどうかを判別しなければならないため、あいまいにならないように Web サイトを設計してください。

このプロトコルオプションにデータベースが設定されている場合は、このデータベースを使用してすべての Web 要求を処理します。URL にはデータベース名を含めないでください。

### 例

次のコマンドは 2 つのデータベースを起動しますが、HTTP 経由でのアクセスを許可されているのはそのうちの 1 つだけです。

```
dbsrv17 -xs http (DBN=web) "%SQLANYSAMP17%¥demo.db" web.db
```

次のコマンドは、2 つの HTTP Web サービスを起動します。1 つは `your-first-database.db` 用、もう 1 つは `your-second-database.db` 用です。

```
dbsrv17 -xs http (PORT=80;DBN=your-first-database),http (PORT=8800;DBN=your-second-database) your-first-database.db your-second-database.db
```

## 1.1.7.9 DoBroadcast (DOBROAD) プロトコルオプション

クライアントがデータベースサーバを検索する方法と、データベースサーバが起動時にブロードキャストを実行するかどうかを制御します。

### 構文

```
(database server)
{ DoBroadcast | DOBROAD }={ YES | NO }
(client)
{ DoBroadcast | DOBROAD }={ ALL | NONE | DIRECT }
```

## 使用可能なプロトコル

TCP/IP

## 指定可能な値 (データベースサーバ)

### YES

DoBroadcast=YES に設定すると、データベースサーバは起動時にブロードキャストを実行して、同じ名前の他のデータベースサーバを検索できます。

### NO

DoBroadcast=NO に設定すると、起動時にデータベースサーバがブロードキャストを実行して、同じ名前の他のデータベースサーバを検索しないようにできます。この設定が役に立つ場合もまれにありますが、通常は必要ありません。

## 指定可能な値 (クライアント)

### ALL

DoBroadcast=ALL に設定した場合、ブロードキャストを実行してデータベースサーバを検索します。最初は、ローカルサブネットにブロードキャストされます。HOST が指定されている場合、ブロードキャストパケットも各ホストに送信されます。ブロードキャストパケットはすべて UDP パケットです。

### NONE

DoBroadcast=NONE を指定すると、UDP ブロードキャストは使用されず、データベースサーバアドレスキャッシュ (sasrv.ini ファイルに存在) は無視されます。指定した HOST/PORT との TCP/IP 接続が直接行われ、サーバ名が検証されます。DoBroadcast=NONE を指定する場合は、HOST (IP) プロトコルオプションが必要です。

### DIRECT

DoBroadcast=DIRECT を設定した場合、データベースサーバを検索するときに、ローカルサブネットへのブロードキャストは実行されません。ブロードキャストパケットは、HOST (IP) プロトコルオプションリストにあるホストにのみ送信されます。DoBroadcast=DIRECT を指定する場合は、HOST (IP) プロトコルオプションが必要です。

## デフォルト

YES (データベースサーバ)

ALL (クライアント) HOST (IP) プロトコルオプションが指定された場合、デフォルトは DIRECT です。

## 備考

### クライアントでの使用法

TCP/IP では、VerifyServerName (VERIFY) プロトコルオプションを NO に設定することで、データベースサーバ名の検証を回避できます。HOST (IP) プロトコルオプションは、LDAP が使用されている場合以外は必須のパラメータです。ServerPort (PORT) プロトコルオプションは任意です。

DIRECT と NONE の場合は、HOST オプションでデータベースサーバホストを指定します。

## 例

次の接続文字列は、silver という名前のコンピュータ上で実行している gold という名前のデータベースサーバに接続します。

```
Host=silver;ServerName=gold
```

次の接続文字列は、CommLinks プロトコルオプションを使用して同じデータベースサーバに接続します。

```
CommLinks=tcpip(DOUBROADCAST=DIRECT;HOST=silver);ServerName=gold
```

## 関連情報

[Broadcast \(BCAST\) プロトコルオプション \[183 ページ\]](#)

[BroadcastListener \(BLISTENER\) プロトコルオプション \(サーバ側のみ\) \[184 ページ\]](#)

## 1.1.7.10 ExitOnError (EXIT) プロトコルオプション

起動時にエラーが発生した場合に OData サーバを停止するかどうかを制御します。

### 構文

```
{ ExitOnError | EXIT }={ YES | NO }
```

## 使用可能なプロトコル

OData

## 指定可能な値 (データベースサーバ)

### YES

ExitOnError=YES と設定すると、起動時にエラーが発生した場合に OData サーバが停止します。

### NO

ExitOnError=NO と設定すると、起動時にエラーが発生しても OData サーバは実行されたままです。

デフォルト

YES

備考

起動時にエラーが発生した場合に OData サーバを停止するかどうかを指定します。

 例

次の例では、ポート 8080 で OData 要求を受信し、起動時にエラーが発生すると停止するデータベースサーバを起動します。

```
dbsrv17 -xs odata(ServerPort=8080;EXIT=YES) c:¥mydatabase.db
```

## 1.1.7.11 FIPS プロトコルオプション

データベースクライアント/サーバ (TLS) や web サーバ (HTTPS) の通信の暗号化に、FIPS 認定 (FIPS PUB 140-2) のセキュリティアルゴリズムを使用できます。

 構文

```
FIPS={ YES | NO }
```

使用可能なプロトコル

HTTPS、TLS

指定可能な値

YES、NO

デフォルト

NO

## 備考

FIPS 認定の暗号は、RSA 暗号化でのみでサポートされています。RSA 暗号化の実装に使用されたコードが認定されたものであるかどうかにかかわらず、同じ暗号化が実行されます。FIPS 認定でない暗号化アルゴリズムを使用しているクライアントと、FIPS 認定の暗号化アルゴリズムを使用しているサーバとの通信も可能であり、その逆もまた可能です。

FIPS 認定の暗号化には別途ライセンスが必要です。

### 例

次のコマンドラインは、Web サーバの ServerPort、FIPS、Identity、Identity\_Password ネットワークプロトコルオプションを設定します。

```
dbsrv17 -xs
https(serverport=544;fips=yes;identity=rsaserver.id;identity_password=test)
demo.db
```

次のコマンドラインは、データベースサーバの、FIPS、Identity、Identity\_Password ネットワークプロトコルオプションを設定します。

```
dbsrv17 -ec tls(fips=yes;identity=rsaserver.id;identity_password=test) demo.db
```

次のコマンドは、トランスポートレイヤセキュリティを使用して、Interactive SQL と、前の例で起動したデータベースサーバ demo とを接続します。

```
dbisql -c "UID=DBA;PWD=sql;Server=demo;Host=localhost;Encryption=
TLS(fips=yes;trusted_certificate=rsaroot.crt;certificate_name=RSA
Server;certificate_company=SAP;certificate_unit=SQL Anywhere)"
```

certificate\_name オプション、certificate\_company オプション、certificate\_unit オプションによって、rsaserver.id 認定ファイルを使用したサーバへの接続を制限しています。

## 関連情報

- [SQL Anywhere クライアント/サーバ通信の暗号化 \[1653 ページ\]](#)
- [certificate\\_company プロトコルオプション \(クライアント側のみ\) \[186 ページ\]](#)
- [certificate\\_name プロトコルオプション \(クライアント側のみ\) \[187 ページ\]](#)
- [certificate\\_unit プロトコルオプション \(クライアント側のみ\) \[188 ページ\]](#)
- [trusted\\_certificates プロトコルオプション \[234 ページ\]](#)
- [Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)
- [証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)
- [-ec データベースサーバオプション \[402 ページ\]](#)
- [-fips データベースサーバオプション \[411 ページ\]](#)
- [-xs データベースサーバオプション \[507 ページ\]](#)

## 1.1.7.12 Host (IP) プロトコルオプション (クライアント側のみ)

データベースサーバのアドレスを指定します。

### 構文

```
{ Host | IP }=ip-address
```

### 使用可能なプロトコル

TCP/IP (クライアント側のみ)

### 指定可能な値

#### ip-address

この文字列は、IP アドレス形式で指定する必要があり、任意でポート番号 (コロンで区切る) を含めることができます。ホスト値のリストは、カンマで区切って指定します。IPv6 アドレスの場合は、アドレスを角カッコまたはカッコで囲む必要があります。**localhost** を使用すれば現在のコンピュータを識別できます。次に例を示します。

```
links=tcPIP (HOST=myhost)
links=tcPIP (HOST=myhost:1234)
links=tcPIP (HOST=10.25.13.5,myotherhost)
links=tcPIP (HOST=myhost:1234,10.25.65.112)
links=tcPIP (HOST=myhost:1234,myotherhost:5678)
```

### デフォルト

現在の TCP/IP サブセットを検索します。

### 備考

Host は、データベースサーバのアドレスを指定します。IP と Host は同義語です。カンマで区切ったアドレスのリストを使って、複数のコンピュータを検索できます。また、コロンを区切り文字として使用してポート番号を IP アドレスに追加できます。別の方法として、HOST=myhost;PORT=5000 のように、ServerPort プロトコルオプションを明示的に使用してホストとサーバのポートを指定することもできます。IPv6 アドレスの場合は、アドレスを角カッコまたはカッコで囲む必要があります。例：

[fe80::5445:5245:444f]:2638。ポートを指定した場合は、そのポート番号のみが TCP/IP 接続と UDP ブロードキャストに使用されます。ポート番号を指定しない場合は、ポート 2638 が使用されます。

Windows プラットフォームで IPv6 アドレスを指定する場合、インタフェース識別子を使用する必要があります。UNIX プラットフォームでは、IPv6 アドレスのインタフェース識別子とインタフェース名の両方がサポートされます。Linux (カーネル 2.6.13 以降) では、インタフェース識別子が必要です。

TCP/IP を使用してパーソナルデータベースサーバに接続する場合は、ホスト名 `localhost` を使用する必要があります。-x オプションを使用して、パーソナルデータベースサーバで TCP/IP 通信プロトコルを有効にする必要があります。

-z オプションを使用すると、データベースサーバに、起動時にデータベースサーバメッセージウィンドウでアドレス情報を表示させることができます。

LogFile 接続パラメータを指定すると、クライアントアプリケーションに、アドレス指定情報をログファイルに書き込ませることができます。

Host プロトコルオプションまたは Host 接続パラメータと共に LocalOnly プロトコルオプションを指定することはできません。

Host プロトコルオプションは、Host 接続パラメータとは別のものです。Host プロトコルオプションは、CommLinks 接続パラメータで使用されます。CommLinks (LINKS) 接続パラメータは、Host または ServerPort (PORT) 以外の TCP/IP オプションを指定する必要がある場合のみ使用してください。接続文字列に CommLinks と Host の両方は指定できません。

CommLinks 接続パラメータにはホスト名の指定が必要ですが、ポート番号および ServerName 接続パラメータはオプションです。したがって、可能な組み合わせは 4 つになります。

Host 接続パラメータ	相当する CommLinks (LINKS) 接続パラメータの文字列
Host=serverhost:1234 serverhost のポート 1234 上で実行しているデータベースサーバに接続試行が行われます。	LINKS=TCPIP (HOST=serverhost:1234;DoBroadcast=None;VerifyServerName=No)
Host=serverhost:1234; ServerName=myserver serverhost のポート 1234 上で実行している myserver というデータベースサーバに接続試行が行われます。	LINKS=TCPIP (HOST=serverhost:1234;DoBroadcast=None;VerifyServerName=Yes); ServerName=myserver
Host=serverhost serverhost のデフォルトポート 2638 上で実行しているデータベースサーバに接続試行が行われます。	LINKS=TCPIP (HOST=serverhost:2638;DoBroadcast=None;VerifyServerName=No)
Host=serverhost; ServerName=myserver serverhost の任意のポート上で実行している myserver というデータベースサーバに接続試行が行われます。	LINKS=TCPIP (HOST=serverhost;DoBroadcast=Direct;VerifyServerName=Yes); ServerName=myserver

## 例

次の接続文字列フラグメントは、kangaroo と 197.75.209.222 (ポート 2369) というコンピュータを検索し、データベースサーバを見つけるようにクライアントに指示します。

```
LINKS=tcPIP (IP=kangaroo,197.75.209.222:2369)
```

次の接続文字列フラグメントは、my-server と kangaroo というコンピュータを検索し、データベースサーバを見つけるようにクライアントに指示します。

```
LINKS=tcpip (HOST=my-server, kangaroo; PORT=2639)
```

次の接続文字列フラグメントは、host1 のポート 1234 で稼働するサーバと、host2 のポート 4567 で稼働するサーバを検索するようにクライアントに指示します。クライアントは、host1 のポート 4567 や host2 のポート 1234 は検索しません。

```
LINKS=tcpip (HOST=host1:1234, host2:4567)
```

次の接続文字列フラグメントは、IPv6 アドレスのサーバを探すようクライアントに指示します。

```
LINKS=tcpip (HOST=fe80::5445:5245:444f)
```

次の接続文字列フラグメントは、ポート 2639 の IPv6 アドレス上でデータベースサーバを探すようクライアントに指示します。

```
LINKS=tcpip (HOST=[fe80::5445:5245:444f]:2639)
```

Host プロトコルオプションを指定した IPv6 アドレスの使用例を次に示します。

```
// Global scope address, unique everywhere, so no interface index is required
// no index required
-c "links=tcpip(host=fd77:55d:59d9:56a:202:55ff:fe76:df19) "
// all communication is done through interface 2
-c "links=tcpip(host=fd77:55d:59d9:56a:202:55ff:fe76:df19%2) "
// all communication is done through eth0
-c "links=tcpip(host=fd77:55d:59d9:56a:202:55ff:fe76:df19%eth0) "
// Link scope address, addresses are unique on each interface
// possibly ambiguous (this host may exist through both eth0 and eth1)
-c "links=tcpip(host=fe80::202:55ff:fe76:df19) "
// not ambiguous because it must use interface 2
-c "links=tcpip(host=fe80::202:55ff:fe76:df19%2) "
// not ambiguous because it must use eth0
-c "links=tcpip(host=fe80::202:55ff:fe76:df19%eth0) "
```

## 関連情報

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[IPv6 のサポート \[168 ページ\]](#)

[-z データベースサーバオプション \[509 ページ\]](#)

[LogFile \(LOG\) 接続パラメータ \[92 ページ\]](#)

[ClientPort \(CPORT\) プロトコルオプション \(クライアント側のみ\) \[190 ページ\]](#)

[ServerPort \(PORT\) プロトコルオプション \[225 ページ\]](#)

## 1.1.7.13 SecureMyIP プロトコルオプション

データベースサーバが保護された接続を受信するネットワークインタフェースを指定します。

### 構文

```
SecureMyIP=IP-address
```

### 使用可能なプロトコル

OData

### 指定可能な値

#### IP-address

この文字列は IP アドレス形式で指定する必要があります。

### 備考

SecureMyIP プロトコルオプションは、複数のネットワークアダプタを持つコンピュータに対して指定します。

各アダプタには IP アドレスがあります。デフォルトでは、データベースサーバは検出したすべてのネットワークインタフェースを使用します。データベースサーバがすべてのネットワークインタフェースで受信しないようにする場合、使用するインタフェースのアドレスを SecureMyIP プロトコルオプションで指定します。

Windows プラットフォームで IPv6 アドレスを指定する場合は、インタフェース識別子を使用します。UNIX プラットフォームでは、IPv6 アドレスのインタフェース識別子とインタフェース名の両方がサポートされます。Linux (カーネル 2.6.13 以降) では、インタフェース識別子が必要です。

### 例

次のコマンドは、指定した保護されたネットワーク上の保護された OData 要求およびすべてのインタフェース上の保護されていない要求を受信するようデータベースサーバに指示します。

```
dbsrv17 -xs odata (SecureMyIP=192.75.209.12;SSLKeyStore=c:\mykeys.jks;SSLKeyStorePassword=password)
```

## 1.1.7.14 identity プロトコルオプション

ID ファイルの名前を指定します。

### 構文

```
identity=identity-file
```

### 使用可能なプロトコル

HTTPS、TLS

### 指定可能な値

#### identity-file

この文字列は、ID ファイルの名前を指定します。

### デフォルト

デフォルトの ID ファイル名はありません。

### 備考

トランスポートレイヤセキュリティを使用する場合、ID ファイルにはパブリック証明書とそのプライベートキーが含まれています。自己署名されていない証明書の場合は、その証明書に署名を行うすべての証明書も ID ファイルに格納されています。identity\_password オプションを使用することで、プライベートキーのパスワードを指定します。

### 例

TLS 暗号を使用するために、クライアント接続が必要なデータベースサーバを起動します。

```
dbsrv17 -ec tls(fips=no;identity=rsaserver.id;identity_password=test) demo.db
```

次のコマンドは、トランスポートレイヤセキュリティを使用して、Interactive SQL と、前の例で起動したデータベースサーバ demo とを接続します。

```
dbisql -c "UID=DBA;PWD=sql;Server=demo;Host=localhost;Encryption=
  TLS(FIPS=yes;trusted_certificate=rsaroot.crt;certificate_name=RSA
  Server;certificate_company=SAP;certificate_unit=SQL Anywhere)"
```

TLS 暗号を使用するために、web 接続が必要なデータベースサーバを起動します。

```
dbsrv17 -xs https(identity=rsaserver.id;identity_password=test) demo.db
```

### 警告

サンプル ID ファイルは、テスト作業と開発作業にのみ使用します。この証明書は SQL Anywhere の標準部分であるため、保護機能は備えていません。アプリケーションを配備する前に独自の証明書で置換してください。

データベースサーバに TLS 暗号化の使用を要求するクライアントアプリケーションを起動します。

```
dbping -d -c  
"UID=DBA;PWD=sql;HOST=myserverhost;SERVER=myservername;DBN=mydatabase;ENC=TLS(trusted_certificate=myroot.crt; identity=myclient.id;identity_password=client-password)"
```

## 関連情報

- [クライアント側 TLS 証明書の使用 \[1652 ページ\]](#)
- [トランスポートレイヤセキュリティの設定方法 \[1643 ページ\]](#)
- [identity\\_password プロトコルオプション \[202 ページ\]](#)
- [trusted\\_certificates プロトコルオプション \[234 ページ\]](#)
- [Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)
- [-ec データベースサーバオプション \[402 ページ\]](#)

### 1.1.7.15 identity\_password プロトコルオプション

暗号化証明書のパスワードを指定します。

#### 構文

```
identity_password=password
```

## 使用可能なプロトコル

HTTPS、TLS

## 指定可能な値

### password

この文字列は、暗号化証明書のパスワードを指定します。

## デフォルト

デフォルトの ID ファイルのパスワードはありません。

## 備考

トランスポートレイヤセキュリティを使用している場合、このオプションにより、identity プロトコルオプションで指定した暗号化証明書に対応するパスワードを指定します。

### 例

特定の証明書を使用するために、Web 接続が必要なデータベースサーバを起動します。

```
dbsrv17 -xs https(identity=rsaserver.id;identity_password=test) demo.db
```

特定の証明書を使用するために、クライアント接続が必要なデータベースサーバを起動します。

```
dbsrv17 -ec tls(identity=rsaserver.id;identity_password=test) demo.db
```

検証可能な証明書をデータベースサーバに提供し、特定の証明書の使用をデータベースに要求するクライアントアプリケーションを起動します。

```
dbping -d -c  
"UID=DBA;PWD=sql;HOST=myserverhost;SERVER=myservername;DBN=mydatabase;ENC=TLS (trusted_certificate=myroot.crt; identity=myclient.id;identity_password=client-password) "
```

## 関連情報

[クライアント側 TLS 証明書の使用 \[1652 ページ\]](#)

[トランスポートレイヤセキュリティの設定方法 \[1643 ページ\]](#)

[identity プロトコルオプション \[201 ページ\]](#)

[trusted\\_certificates プロトコルオプション \[234 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

## 1.1.7.16 KeepaliveTimeout (KTO) プロトコルオプション

HTTP 要求間の最大アイドル時間を秒単位で指定します。

### 構文

```
{ KeepaliveTimeout | KTO }=timeout-value
```

### 使用可能なプロトコル

HTTP

HTTPS

### 指定可能な値

#### timeout-value

この整数は、HTTP 要求間に待機する時間数を秒単位で指定します。接続がタイムアウトされないようにする場合は、KTO=0 を指定します。運用環境では、悪質なクライアントがデータベースサーバのリソースを消費して、他のクライアントが接続できなくなる可能性があるため、KTO=0 に設定することはおすすめしません。

### デフォルト

60

### 備考

このプロトコルオプションは、keep-alive HTTP 要求にのみ適用されます。これは、最新の HTTP 要求が受信された時間から、データが次の HTTP 要求用に最初に受信されるときまでに許可される最大アイドル時間を指定します。新しい要求用のデータが受信されると、Timeout プロトコルオプションで定義されたタイムアウト期間の計測が開始されます。

### 例

```
dbsrv17 -n demo1 -xs https (port=443;  
identity=C:\Users\Public\Documents\SQL Anywhere 17\Samples\Certificates  
\rsaserver.id;  
identity_password=test;kto=50) demo.db
```

## 関連情報

[Timeout \(TO\) プロトコルオプション \[232 ページ\]](#)

### 1.1.7.17 LDAP プロトコルオプション

クライアントは、IP アドレスを指定せずにデータベースサーバを検索できます。

#### 構文

```
LDAP={ YES | NO | filename }
```

## 使用法

TCP/IP

## 指定可能な値

### YES

LDAP=YES と指定すると、LDAP ネームサーバのサポートが有効になり、`saldap.ini` (デフォルトファイル名) が設定ファイルとして使用されます。

### NO

NO を指定すると、LDAP ネームサーバのサポートが無効になります。

### filename

LDAP=filename を指定すると、LDAP ネームサーバのサポートが有効になり、指定したファイルが設定ファイルとして使用されます。

## デフォルト

YES

## 備考

データベースサーバ自体を LDAP サーバに登録することにより、クライアントが LDAP サーバにクエリを実行できます。これにより、WAN 上で動作するクライアントやファイアウォールを経由するクライアントが IP アドレスを指定せずにサーバを検索できます。また、検出ユーティリティ (dblocate) もそのようなサーバを検索できるようになります。

LDAP プロトコルオプションに完全なパスを指定していない限り、saldap.ini ファイルは、SQL Anywhere 実行プログラムと同じディレクトリ (たとえば、Windows の場合は %SQLANY17%\bin32) に置く必要があります。

ファイル非表示ユーティリティ (dbfhide) を使用して、saldap.ini ファイルの内容を暗号化できます。その場合、ファイルの非暗号化コピーを保持し、暗号化ファイルを更新できるようにします。

LDAP をネームサーバとして使用することは、TCP/IP サーバ通信でのみサポートされます。

### 例

次の例は、LDAP でサーバを起動します

```
dbsrv17 -n demo17 -x tcpip(ldap=saldap.ini) demo.db
```

次の例は、LDAP サーバに接続します

```
dbisql -c links=tcpip(ldap=y);servername=demo17;uid=dba;pwd=sql
```

## 関連情報

[ネームサーバとして LDAP を使用した接続 \[171 ページ\]](#)

[saldap.ini ファイル設定 \[172 ページ\]](#)

[.ini ファイルの内容を非表示にする \[566 ページ\]](#)

### 1.1.7.18 LocalOnly (LO) プロトコルオプション

データベースサーバが接続をローカルコンピュータのみに制限できるようにします。

クライアントがローカルコンピュータ上のサーバ (存在する場合) のみに接続できるようにします。

### 構文

```
{ LocalOnly | LO }={ YES | NO }
```

## 使用可能なプロトコル

TCP/IP、HTTP、HTTPS

## デフォルト

NO

## 備考

### データベースサーバ

LocalOnly (LO) プロトコルオプションをサーバで使用して、ローカルコンピュータへの接続に限定させることができます。リモートコンピュータからの接続試行ではこのサーバは検索されず、検出 (dblocate) ユーティリティからはこのサーバは見えません。

デフォルト値は NO で、データベースサーバはどの場所にあるクライアントの要求も受け入れます。

このオプションを YES に設定すると、ネットワークデータベースサーバは異なるコンピュータで実行中のクライアントからの通信をすべて拒否します。

YES に設定すると、ネットワークサーバは接続や CPU の制限がないパーソナルサーバとして実行されます。このオプションは、他のコンピュータからの要求を受け入れることのないパーソナルデータベースサーバには影響しません。

### クライアント

LocalOnly (LO) プロトコルオプションを使用すると、ローカルコンピュータ上で実行しているデータベースサーバへの接続を制限できます。

デフォルト値は NO であり、クライアントはサーバがどこにありようと接続を試行します。

YES に設定すると、通常のブロードキャストメカニズムは使用されますが、他のコンピュータ上のデータベースサーバからのブロードキャスト応答は無視されます。

HOST プロトコルオプションまたは Host 接続パラメータと共に LocalOnly プロトコルオプションを指定することはできません。

LocalOnly (LO) プロトコルオプションは、DoBroadcast=ALL (デフォルト) が指定されている場合にのみ意味があり、この場合のみ受け入れられます。

## 関連情報

[Broadcast \(BCAST\) プロトコルオプション \[183 ページ\]](#)

[DoBroadcast \(DOBROAD\) プロトコルオプション \[192 ページ\]](#)

## 1.1.7.19 LogFile (LOG) プロトコルオプション

データベースサーバが Web 要求に関する情報を書き込むファイル名を指定します。

### 構文

```
{ LogFile | LOG }=filename
```

### 使用可能なプロトコル

HTTP、HTTPS、OData

### デフォルト

なし

### 備考

このプロトコルオプションは、データベースサーバが Web 要求に関する情報を書き込むファイル名を指定します。

### 例

次の例では、ポート 8080 で OData 要求を受信するデータベースサーバを起動し、ログファイル myodata.log を使用します。

```
dbsrv17 -xs odata (ServerPort=8080;LOG=myodata.log) c:¥mydatabase.db
```

### 関連情報

[LogFormat \(LF\) プロトコルオプション \[209 ページ\]](#)

[LogMaxSize \(LSIZE\) プロトコルオプション \[211 ページ\]](#)

[LogOptions \(LOPT\) プロトコルオプション \[212 ページ\]](#)

## 1.1.7.20 LogFormat (LF) プロトコルオプション

データベースサーバがメッセージログファイルに書き込む Web 要求に関する情報メッセージのフォーマットを制御し、メッセージに表示されるフィールドを指定します。

### 構文

```
{ LogFormat | LF }=format-string
```

### 使用可能なプロトコル

HTTP、HTTPS

### 指定可能な値

#### format-string

次のコードがサポートされます。

@@

@ 文字

@B

要求の処理が開始された日付/時刻 (エラーにより要求をキューイングできない場合を除く)

@C

クライアントが接続した日付/時刻

@D

要求に関連するデータベース名

@E

エラーが発生した場合の、エラーメッセージテキスト

@F

要求の処理が終了した日付/時刻

@I

クライアントの IP アドレス

@J

@I オプションで指定されたクライアントポートのログを取ります。

@L

ヘッダと本文を含んだ応答の長さ (バイト)

@M

HTTP 要求方式

@P

要求に関連するリスナポート

@Q

要求の処理がキューイングされた日付/時刻 (エラーにより要求をキューイングできない場合を除く)

@R

HTTP 応答のステータスコードおよび説明

@S

HTTP ステータスコード

@T

現在のログエントリが書き込まれた日付/時刻

@U

要求 URI

@V

要求 HTTP バージョン

@W

要求を処理した時間 (@F - @B)、またはエラーにより要求が処理されなかった場合は 0.000

## デフォルト

@T - @W - @I:@J - @P - "@M @U @V" - @R - @L - @E

## 備考

このプロトコルオプションは、Web 要求に関する情報を格納するメッセージログファイルに書き込まれるメッセージのフォーマットと、表示されるフィールドを制御します。文字列に表示される場合、各メッセージが書き込まれると現在の値がコードに置き換えられます。

HTTP 要求メソッドがサポートされていない、URI のフォーマットが不正である、URI に必要なデータベース名が含まれていないなどの理由で Web 要求が失敗した場合、HTTP メソッド (@M) と HTTP バージョン (@V) は文字列 ??? を返し、URI (@U) は >>> に続いて指定された要求を返します。

たとえば、ログフォーマットが「@M @U @V」に設定されているとき、URL request/sample/test HTTP/1.0 に対して未知の HTTP メソッドによる要求が行われると、次の文字列が返されます。

```
"???">>>request/sample/test HTTP/1.0???"
```

## 関連情報

[LogFile \(LOG\) プロトコルオプション \[208 ページ\]](#)

[LogMaxSize \(LSIZE\) プロトコルオプション \[211 ページ\]](#)

[LogOptions \(LOPT\) プロトコルオプション \[212 ページ\]](#)

### 1.1.7.21 LogMaxSize (LSIZE) プロトコルオプション

データベースサーバが Web 要求に関する情報を書き込むメッセージログファイルの最大サイズを制御します。

#### 構文

```
{ LogMaxSize | LSIZE }=size[ k | m | g ]
```

#### 使用可能なプロトコル

HTTP、HTTPS

#### 指定可能な値

##### size

この整数は、Web 要求情報が書き込まれるファイルの最大サイズを指定します。デフォルト値はバイト単位ですが、k、m、または g を使用して、それぞれキロバイト単位、メガバイト単位、ギガバイト単位を指定できます。LogMaxSize が 0 の場合、メッセージログファイルのサイズは無制限です。

#### デフォルト

0

#### 備考

メッセージログファイルが指定したサイズに達すると、名前が変更されて、別のログファイルが作成されます。

## 関連情報

[LogFile \(LOG\) プロトコルオプション \[208 ページ\]](#)

[LogFormat \(LF\) プロトコルオプション \[209 ページ\]](#)

[LogOptions \(LOPT\) プロトコルオプション \[212 ページ\]](#)

### 1.1.7.22 LogOptions (LOPT) プロトコルオプション

データベースサーバが Web 要求に関する情報を書き込むログに記録されるメッセージタイプを指定します。

#### 構文

```
{ LogOptions | LOPT } = [ NONE ] [, OK ] [, INFO ] [, ERRORS ] [, ALL ] [, status-codes ] [, REQHDRS ] [, RESHDRS ] [, HEADERS ]
```

#### 使用可能なプロトコル

HTTP、HTTPS

#### 指定可能な値

次のキーワードは、ログ出力するメッセージのカテゴリを制御します。

##### NONE

何もログ出力しない

##### OK

ログ要求が正しく完了 (20x HTTP ステータスコード)

##### INFO

終了または未変更ステータスコードを返すログ要求 (3xx HTTP ステータスコード)

##### ERRORS

すべてのエラーのログを取る (4xx と 5xx HTTP ステータスコード)

##### USER

すべてのエラーのログを取る (6xx、7xx、8xx、9xx HTTP ステータスコード)

##### ALL

すべての要求をログ出力する

次の共通 HTTP ステータスコードも使用可能です。特定のステータスコードを返す要求をログ出力するために使用できます。

C200

OK

C400

不正な要求

C401

無認可

C403

禁止

C404

見つからない

C408

要求タイムアウト

C501

未実装

C503

サービス利用不可

次のキーワードを使用してログ出力されたメッセージの詳細情報を取得できます。

**REQHDRS**

要求のロギング時に、メッセージログファイルに要求ヘッダも書き込みます。

**RESHDRS**

要求のロギング時に、メッセージログファイルに応答ヘッダも書き込みます。

**HEADERS**

要求のロギング時に、メッセージログファイルに要求ヘッダと応答ヘッダの両方を書き込みます (REQHDRS,RESHDRSと同様)。

## デフォルト

ALL

## 備考

使用可能な値には、特定のメッセージタイプと HTTP ステータスコードを選択するキーワードがあります。カンマで区切って複数の値を指定できます。

## 関連情報

[LogFile \(LOG\) プロトコルオプション \[208 ページ\]](#)

[LogFormat \(LF\) プロトコルオプション \[209 ページ\]](#)

[LogMaxSize \(LSIZE\) プロトコルオプション \[211 ページ\]](#)

### 1.1.7.23 LogRename (LRENAME) プロトコルオプション

ユーザが以前の HTTP デバッグログファイルの名称を変更できます。

#### 構文

```
{ LogRename | LRENAME } = { OLD | DATE | number }
```

#### 使用可能なプロトコル

HTTP

#### 指定可能な値

**OLD** 既存の HTTP デバッグログファイルの名称を `logfile.old` に変更し、既存の `logfile.old` ファイルを削除します。これはデフォルトです。

#### DATE

既存の HTTP デバッグログファイルの名称を `logfile-date-num.ext` に変更します。ここで、`logfile.ext` は元の HTTP デバッグログファイルで、`date` は YYYYMMDD 形式の現在の日付で、`num` は 00 から始まる 2 桁の番号です。

#### i 注記

LogRename=DATE と指定すると、HTTP デバッグログファイルを無制限に作成できます。必要に応じて、未使用のファイルをクリーンアップします。

**number** 以前の HTTP デバッグログファイルの指定した番号を保持し、それらに `logfile.1`、`logfile.2`、`logfile.3` といった名称を付けます。`number` には、必ず正の整数を指定してください。

#### デフォルト

OLD

## 備考

LogRename は、Log および LogMaxSize プロトコルオプションとともに使用すると、以前の HTTP デバッグログファイルの名称の付け方をデータベースサーバに指示します。

### 1.1.7.24 LogVerbosity プロトコルオプション

ログの冗長性レベルを指定します。

#### 構文

```
LogVerbosity = { 1 | 2 | 3 | 4 }
```

#### 使用可能なプロトコル

OData

#### 指定可能な値

1

予期しないエラーに関する情報を出力します。

2

一般的な情報と設定メッセージを出力します。

3

HTTP 要求に関する詳細な情報を出力します。

4

デバッグメッセージを出力します。

#### デフォルト

2

## 備考

冗長性レベルが高い場合、追加情報がロギングされ、すべてのレベルで提供される情報が含まれます。

### 例

次の例では、ログの冗長性レベルを 3 に設定して、ポート 8080 で OData 要求を受信するデータベースサーバを起動します。

```
dbsrv17 -xs odata (ServerPort=8080;LogVerbosity=3) c:¥mydatabase.db
```

## 1.1.7.25 MaxConnections (MAXCONN) プロトコルオプション

データベースサーバで許可される HTTP/HTTPS 同時接続の最大数を指定します。

### 構文

```
{ MaxConnections | MAXCONN }=size
```

## 使用可能なプロトコル

HTTP、HTTPS

## 指定可能な値

**size**

この整数は、データベースサーバで許可される同時接続の数を指定します。値 0 は、無制限であることを示します。

## デフォルト

7 (パーソナルサーバ)

ライセンスされている接続数 (ネットワークサーバ)

## 備考

パーソナルサーバは最大 10 個の同時接続を受け入れることができます。そのうち、7 個のみを HTTP/HTTPS 接続にできます。

データベースサーバの HTTP/HTTPS 接続の制限に達すると、データベースサーバは HTTP/HTTPS 接続試行をキューに入れます。HTTP/HTTPS キューに接続が入っている間、HTTP/HTTPS 接続の代わりに標準接続を行うことはできません。HTTP/HTTPS 接続の数が制限されている場合、または標準接続用にデータベース接続の一部が予約されている場合を除き、標準接続を行うとしているユーザは無限に待機することとなります。ネットワークデータベースサーバは、標準接続用に接続を予約しません。標準接続を予約するには、reserved\_connections データベースオプションを使用します。

## 関連情報

[MaxRequestSize \(MAXSIZE\) プロトコルオプション \[217 ページ\]](#)

[MaxRequestVars \(MAXVARS\) プロトコルオプション \[218 ページ\]](#)

[サーバライセンス取得ユーティリティ \(dblic\) \[1148 ページ\]](#)

[reserved\\_connections オプション \[773 ページ\]](#)

### 1.1.7.26 MaxRequestSize (MAXSIZE) プロトコルオプション

データベースサーバで許可できる最大要求サイズを指定します。

#### 構文

```
{ MaxRequestSize | MAXSIZE }=size[ k | m | g ]
```

## 使用可能なプロトコル

HTTP、HTTPS

## 指定可能な値

### size

この整数は、データベースサーバで許可できる最大要求サイズを指定します。デフォルト値はバイト単位ですが、k、m、または g を使用して、それぞれキロバイト単位、メガバイト単位、ギガバイト単位を指定できます。値 0 はこの制限を無効にしますが、細心の注意を払って使用してください。この制限がないと、悪質なクライアントがデータベースサーバに過負荷をかけたり、メモリ不足を引き起こしたりする可能性があります。

## デフォルト

100k

## 備考

要求のサイズがこの制限を超えると、接続は閉じられ、クライアントは要求が大きすぎることを示す応答を返します。この値は要求サイズのみを制限するもので、応答サイズは制限しません。

### 例

次のコマンドは、150000 バイトまでのサイズの要求を受け入れるようサーバに指示します。

```
dbsrv17 -xs http(MaxRequestSize=150000)
```

## 関連情報

[MaxConnections \(MAXCONN\) プロトコルオプション \[216 ページ\]](#)

[MaxRequestVars \(MAXVARS\) プロトコルオプション \[218 ページ\]](#)

## 1.1.7.27 MaxRequestVars (MAXVARS) プロトコルオプション

データベースサーバに送信される要求内で許可する HTTP 入力変数の最大数を指定します。

### 構文

```
{ MaxRequestVars | MAXVARS }=number
```

## 使用可能なプロトコル

HTTP、HTTPS

## 指定可能な値

**number**

この整数は、データベースサーバに送信される要求内で許可する入力変数の最大数を指定します。値 0 は上限なしを示しますが、許可される入力変数の数は、MaxRequestSize プロトコルオプションで決定される、要求内で許可されるバイト数により引き続き制限されます。

## デフォルト

10000

## 備考

入力変数の数がこの制限を超えると、接続は閉じられ、クライアントは要求が不正な要求であることを示す応答を返します。バージョン 12 以前のデータベースサーバには、HTTP または HTTPS 入力要求変数の数に関する制限がありませんでした。MaxRequestVars プロトコルオプションを 0 に設定すると以前の動作にリストアされますが、この方法はお奨めできません。

### 例

このコマンドにより、最大 100 の入力変数を持つ要求を受け入れるデータベースサーバが起動されます。

```
dbsrv17 -xs http( MaxRequestVars=100 )
```

## 関連情報

[MaxConnections \(MAXCONN\) プロトコルオプション \[216 ページ\]](#)

[MaxRequestSize \(MAXSIZE\) プロトコルオプション \[217 ページ\]](#)

## 1.1.7.28 MyIP (ME) プロトコルオプション

データベースサーバまたはクライアントコンピュータにネットワークインタフェースコントローラ (NIC) またはアダプタが複数存在する場合にどれを使用するかを細かく制御するには、このオプションを使用します。

### 構文

```
{ MyIP | ME }={ ip-address [, ip-address ... ] | NONE }
```

## 使用可能なプロトコル

TCP/IP、HTTP、HTTPS、OData

## 指定可能な値

### ip-address

この文字列は IP アドレス形式で指定する必要があります。複数の IP アドレスを指定する場合はカンマで区切ります。

OData の場合、IP アドレスは 1 つしか指定できません。

### NONE

IP アドレスとしてキーワード NONE が指定されている場合は、ローカル IP アドレスを決定しようとしません。NONE キーワードは、複数のネットワークカードを持つコンピュータや、リモートアクセスサービス (RAS) とネットワークアダプタを持つコンピュータなど、この操作により大きな負荷がかかるコンピュータシステム上のクライアントで使用するものです。データベースサーバでは使用しないでください。

## 備考

データベースサーバの場合、このオプションはクライアント接続をリッスンするために使用するサーバ IP アドレスを示します。

クライアントの場合、このオプションはデータベースサーバと通信しようとするときに試行するクライアント IP アドレスを示します。

各 NIC またはネットワークアダプタは、IP アドレスにより識別されます。デフォルトでは、データベースサーバは検出したすべてのネットワークインタフェースを使用します。データベースサーバがすべてのネットワークインタフェースで受信しないようにする場合、使用する各インタフェースのアドレスを MyIP (ME) プロトコルオプションで指定します。

Windows プラットフォームで IPv6 アドレスを指定する場合、インタフェース識別子を使用する必要があります。UNIX プラットフォームでは、IPv6 アドレスのインタフェース識別子とインタフェース名の両方がサポートされます。Linux (カーネル 2.6.13 以降) では、インタフェース識別子が必要です。

### 例

次のコマンドライン (すべて 1 行に入力) は、データベースサーバに 2 つのネットワークアダプタを使用することを指示します。

```
dbsrv17 -x tcpip(MyIP=192.75.209.12,192.75.209.32) "%SQLANYSAMF17%¥demo.db"
```

次のコマンドライン (すべて 1 行に入力) は、IPv6 ネットワークアダプタを使用することをデータベースサーバに指示します。

```
dbsrv17 -x tcpip(MyIP=fe80::5445:5245:444f) "%SQLANYSAMF17%¥demo.db"
```

次の接続文字列フラグメントは、クライアントがアドレス情報を決定しないように指定します。

```
LINKS=tcpip(MyIP=NONE)
```

次のコマンドは、指定したネットワークアダプタ上で OData 要求を受信するようデータベースサーバに指示します。

```
dbsrv17 -xs odata (MyIP=192.75.209.12)
```

## 関連情報

[TCP/IP プロトコル \[166 ページ\]](#)

[IPv6 のサポート \[168 ページ\]](#)

### 1.1.7.29 QuietConsole (QUIET) プロトコルオプション

サーバメッセージウィンドウが最小化された状態で起動するかどうかを OData サーバに指示します。

#### 構文

```
{ QuietConsole | QUIET }={ YES | NO }
```

## 使用可能なプロトコル

OData

## 指定可能な値

**YES** QuietConsole=YES と指定すると、OData サーバはサーバメッセージウィンドウが最小化された状態で起動します。これはデフォルトです。

**NO** QuietConsole=NO と指定すると、OData サーバはサーバメッセージウィンドウが表示された状態で起動します。

## デフォルト

YES

## 備考

OData サーバがサーバメッセージウィンドウが最小化された状態で起動するかどうかを指定します。

### 例

次の例では、ポート 8080 で OData 要求を受信し、サーバメッセージウィンドウが表示された状態で起動するデータベースサーバを起動します。

```
dbsrv17 -xs odata (ServerPort=8080;QUIET=NO) c:¥mydatabase.db
```

## 1.1.7.30 ReceiveBufferSize (RCVBUFSZ) プロトコルオプション

TCP/IP プロトコルスタックが使用する受信バッファのサイズを設定します。

### 構文

```
{ ReceiveBufferSize | RCVBUFSZ }=size[ k | m | g ]
```

## 使用可能なプロトコル

TCP/IP

## 指定可能な値

### size

TCP/IP スタックが使用する受信バッファのサイズを指定します。指定可能な最大サイズは 1 MB です。これは、その影響を熟知しているユーザだけが使用するような高度なオプションです。

## デフォルト

コンピュータによって異なります。

## 備考

ネットワーク経由の BLOB パフォーマンスが重要である場合は、この値を増やせます。

## 関連情報

[TCP/IP プロトコル \[166 ページ\]](#)

### 1.1.7.31 SecureServerPort プロトコルオプション

OData サーバが保護された (HTTPS) 接続を受信するポートを指定します。

#### 構文

```
SecureServerPort=port-number
```

## 使用可能なプロトコル

OData

## 指定可能な値

### port-number

OData サーバの場合、SecureServerPort プロトコルオプションは HTTPS を使用する通信のポート番号を指定します。

たとえば (port=1234) のように、単一のポート番号を指定します。

クライアントの場合、SecureServerPort プロトコルオプションは、データベースサーバが HTTPS 通信を受信している 1 つまたは複数のポートをそのクライアントに知らせます。クライアントは、SecureServerPort プロトコルオプションで指定されたすべてのポートにブロードキャストを実行して、OData サーバを検索します。

## デフォルト

443

## 備考

HTTPS 接続を許可しない場合は、"SecureServerPort=" と指定するか、SSLKeyStore (KEYSTORE) プロトコルオプションを指定しないでください。

### 例

次のコマンドを実行して、ポート 8443 で保護された OData 要求を受信するデータベースサーバを起動します。キーストア `c:¥mykeys.jks` とキーストアパスワード `password` を使用します。また、保護されていない OData 要求がデータベースサーバで受信されないように指定します。

```
dbsrv17 -xs odata (SecureServerPort=8443;SSLKeyStore=c:¥mykeys.jks;SSLKeyStorePassword=password;ServerPort=)
```

## 1.1.7.32 SendBufferSize (SNDBUFSZ) プロトコルオプション

TCP/IP プロトコルスタックが使用する送信バッファのサイズを設定します。

### 構文

```
{ SendBufferSize | SNDBUFSZ }=size[ k | m | g ]
```

## 使用可能なプロトコル

TCP/IP

## 指定可能な値

### size

TCP/IP スタックが使用する送信バッファのサイズを指定します。指定可能な最大サイズは 1 MB です。これは、その影響を熟知しているユーザだけが使用するような高度なオプションです。

## デフォルト

コンピュータによって異なります。

## 備考

ネットワーク経由の BLOB パフォーマンスが重要である場合は、この値を増やせます。

## 関連情報

[TCP/IP プロトコル \[166 ページ\]](#)

### 1.1.7.33 ServerPort (PORT) プロトコルオプション

データベースサーバが受信しているポートを指定します。

#### 構文

```
{ ServerPort | PORT }=port-number
```

## 使用可能なプロトコル

TCP/IP、HTTP、HTTPS、OData

## 指定可能な値

### port-number

データベースサーバの場合、ServerPort プロトコルオプションは TCP/IP を使用する通信のポート番号を指定します。

単一のポート番号、または個々のポート番号の組み合わせやポート番号の範囲を指定できます。ポート番号のリストや範囲を指定すると、データベースサーバは指定されたすべてのポート番号をバインドしようとします。次に例を示します。

- (port=1234)
- (port=1234,1235,1239)
- (port=1234-1238)
- (port=1234-1237,1239,1242)

OData の場合、受け入れるポート番号は 1 つだけです。OData HTTP による接続が許可されないようにするには、ServerPort= と指定します。HTTP と OData の両方、あるいは HTTPS と OData の両方を使用する場合には、どちらかのポート番号を変える必要があります。

クライアントの場合、ServerPort (PORT) プロトコルオプションは、データベースサーバが TCP/IP 通信を受信する 1 つまたは複数のポートをそのクライアントに知らせます。クライアントは、ServerPort (PORT) プロトコルオプションで指定されたすべてのポートにブロードキャストを実行して、データベースサーバを検索します。

## デフォルト

### TCP/IP

2638

### HTTP

80

### HTTPS

443

OData 80

## TCP/IP に関する注記

あるコンピュータ上で TCP/IP を使用するすべてのアプリケーションは、それぞれ別の TCP/IP ポートを使用するので、ネットワークパケットは正しいアプリケーションに到達します。Internet Assigned Numbers Authority は、SQL Anywhere データベースサーバの TCP/IP 通信に対してポート番号 2638 を割り当てています。ただし、その他のアプリケーションはこの予約ポートの使用を許可されていないわけではないため、データベースサーバと別のアプリケーション間でアドレスが衝突する可能性があります。

### サーバ側

サーバ側では、ServerPort (PORT) プロトコルオプションを指定した場合は、指定したポート番号のみが TCP/IP 接続に使用されます。UDP リスナは、指定されたポート番号とデフォルトポート番号 (2638) の両方を使用します。ServerPort プロトコルオプションを指定しない場合は、デフォルトポート (2638) が使用されます。

### クライアント側

ServerPort (PORT) プロトコルオプションを指定した場合は、指定したポート番号のみが TCP/IP 接続と UDP ブロードキャストに使用されます。ServerPort プロトコルオプションを指定しない場合は、ポート 2638 が使用されます。

サーバが 2638 以外のポートで実行中である場合、アプリケーションは、通常はサーバ名を指定し、ポート番号は指定しないでサーバに接続できます。-sb サーバオプションを使用する場合は、ポート番号を指定する必要があります。

## HTTP に関する注記

データベースサーバを Web サーバとして使用している場合、80 および 443 の標準 HTTP および HTTPS ポートで受信を行います。

### 例

次の例は、ServerPort プロトコルオプションを使用する方法を示しています。

1. ネットワークデータベースサーバを起動します。

```
dbsrv17 -x tcpip -n server1
```

ポート番号 2638 が取得されました。

- 別のデータベースサーバを起動しようとします。

```
dbsrv17 -x tcpip -n server2
```

デフォルトのポートは現在使用されているので、サーバは別のポートを起動します。ポート番号は、サーバメッセージウィンドウに表示されます。

- 次のようにポート 2629 でデータベースサーバを起動します。

```
dbsrv17 -x tcpip(ServerPort=2629) -n server3 c:¥mydata.db
```

- 次のように、dbisql から、server3 が Host serverhost3 で起動され、ホスト接続パラメータを使って server3 に接続すると仮定します。

```
dbisql -c "UID=DBA;PWD=sql;ServerName=server3;Host=serverhost3:2629"
```

- 次のように、dbisql から、server3 がホスト serverhost3 で起動され、CommLinks 接続パラメータを使って server3 に接続すると仮定します。

```
dbisql -c  
"UID=DBA;PWD=sql;ServerName=server3;CommLinks=tcpip (HOST=serverhost3;PORT=2629)  
"
```

コンピュータ上の別の Web サーバがすでにポート 80 を使用しているとします。次の例は、代わりにポート 8080 で受信するデータベースサーバを起動します。

```
dbsrv17 -xs http(PORT=8080) -n server3 web.db
```

ポート 8080 で OData 要求を受信するデータベースサーバを起動します。

```
dbsrv17 -xs odata(ServerPort=8080) c:¥mydatabase.db
```

## 関連情報

[-x データベースサーバオプション \[500 ページ\]](#)

[-xs データベースサーバオプション \[507 ページ\]](#)

[-sb データベースサーバオプション \[468 ページ\]](#)

## 1.1.7.34 skip\_certificate\_name\_check プロトコルオプション (クライアント側のみ)

クライアントライブラリがサーバホスト名とデータベースサーバの証明書ホスト名との照合を省略するかどうかを制御します。

### 使用可能なプロトコル

HTTPS、TLS

### デフォルト

なし

### 備考

データベースサーバのホスト名がルート証明書のホスト名のいずれかと一致するかどうか制御するには、このブールオプションを ON または OFF に設定します。このオプションを有効にすると、クライアントはサーバの認証を完全に行うことができないので、攻撃に対して脆弱なままとなります。

TLS または HTTPS 接続を開始する際、クライアントライブラリは、データベースサーバのホスト名をサーバが提供する証明書と照らし合わせます。この照合が行われるのは、certificate\_name、certificate\_company、certificate\_unit のいずれのオプションも指定されていない場合、あるいは、skip\_certificate\_name\_check オプションが有効でない場合です。certificate\_name、certificate\_company、certificate\_unit のいずれかが指定されていれば、それらのオプションだけが確認されます。skip\_certificate\_name\_check オプションを有効にすると、ホスト名のチェックが無効になります。

ホスト名や IP アドレスは、subjectAltName (サブジェクトの別名、SAN) 拡張、および通称 (CN) フィールドから取得します。SAN には、ワイルドカードによって複数のホスト名が指定されている場合があります。たとえば、Google 社の証明書には、\*.google.com、\*.google.ca、\*.android.com が指定されているかもしれません。その場合、www.google.ca は有効なホスト名ということになります。

HTTPS は、Web サービスのクライアントプロシージャだけでサポートされています。

#### 例

次の接続文字列フラグメントは、myrootcert.crt がデータベースサーバの証明書署名チェーンのルートであることを確認しますが、myhost.com が証明書で識別されることは確認しません。

```
HOST=myhost.com; SERVER=myserver; ENCRYPTION=TLS (trusted_certificate=myrootcert.crt;  
skip_certificate_name_check=ON)
```

## 関連情報

[trusted\\_certificates プロトコルオプション \[234 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

### 1.1.7.35 SSLKeyStore (KEYSTORE) プロトコルオプション

OData サーバが通信を暗号化するために使用する SSL 証明書が含まれる Java Key Store のファイルパスを指定します。

#### 構文

```
{ SSLKeyStore | KEYSTORE }=filepath
```

#### 使用可能なプロトコル

OData

#### 指定可能な値

**filepath** Java Key Store のファイルパス。

#### デフォルト

なし

#### 備考

保護された OData 要求を有効にするには、このオプションを指定する必要があります。

#### 例

次のコマンドを実行して、ポート 8443 で保護された OData 要求を受信するデータベースサーバを起動します。キーストア `c:\mykeys.jks` とキーストアパスワード `password` を使用します。また、保護されていない OData 要求がデータベースサーバで受信されないように指定します。

```
dbsrv17 -xs odata (SecureServerPort=8443;SSLKeyStore=c:\mykeys.jks;SSLKeyStorePassword=password;ServerPort=)
```

## 関連情報

[SSLKeyStorePassword \(KEYSTOREPASSWORD\) プロトコルオプション \[230 ページ\]](#)

### 1.1.7.36 SSLKeyStorePassword (KEYSTOREPASSWORD) プロトコルオプション

SSLKeyStore (KEYSTORE) プロトコルオプションによって識別された Java キーストアに対して認証するために OData サーバが使用するパスワードを指定します。

#### 構文

```
{ SSLKeyStorePassword | KEYSTOREPASSWORD }=password
```

#### 使用可能なプロトコル

OData

#### 指定可能な値

**password** Java キーストアに対して認証するためのパスワード

#### デフォルト

なし

#### 備考

この OData オプションを使用して、SSLKeyStore (KEYSTORE) プロトコルオプションの使用時に識別された Java キーストアに対して認証するために OData サーバが使用するパスワードを指定します。

## 例

次のコマンドを実行して、ポート 8443 で保護された OData 要求を受信するデータベースサーバを起動します。キーストア `c:¥mykeys.jks` とキーストアパスワード `password` を使用します。また、保護されていない OData 要求がデータベースサーバで受信されないように指定します。

```
dbsrv17 -xs odata (SecureServerPort=8443;SSLKeyStore=c:¥mykeys.jks;SSLKeyStorePassword=password;ServerPort=)
```

## 関連情報

[SSLKeyStore \(KEYSTORE\) プロトコルオプション \[229 ページ\]](#)

### 1.1.7.37 TDS プロトコルオプション (サーバ側のみ)

データベースサーバへの TDS 接続が許可されるかどうかを制御します。

## 構文

```
TDS={ YES | NO }
```

## 使用可能なプロトコル

TCP/IP (サーバ側のみ)

## デフォルト

YES

## 備考

データベースサーバへの TDS 接続を許可しない場合には、TDS を NO に設定します。データベースサーバに暗号化された接続だけを許可する場合、TDS 接続を許可しないようにする方法はこのプロトコルオプションしかありません。

## 例

次のコマンドは、SAP Open Client または jConnect アプリケーションからの接続は許可しないで、TCP/IP プロトコルを使ってデータベースサーバを起動します。

```
dbsrv17 -x tcpip( TDS=NO ) ...
```

## 関連情報

[-ec データベースサーバオプション \[402 ページ\]](#)

## 1.1.7.38 Timeout (TO) プロトコルオプション

通信 (TCP/IP) 確立時の応答の待ち時間を秒単位で指定します。また、要求 (HTTP または HTTPS) の待ち時間を秒単位で指定します。

## 構文

```
{ Timeout | TO }=timeout-value
```

## 使用可能なプロトコル

TCP/IP、HTTP、HTTPS

## 指定可能な値

### timeout-value

- TCP/IP を使用するとき、この整数は、通信確立時に、応答を待つ時間を秒単位で設定します。指定可能な最長時間は、3600 秒です。
- HTTP または HTTPS を使用するとき、この整数は、完全な要求を受信するまで待つ時間の長さを秒単位で指定します。

## デフォルト

### TCP/IP

5  
HTTP  
30  
HTTPS  
30

## 備考

クライアントでは、このプロトコルオプションはどの程度長く待機するのかを指定します。

- サーバ検索時の UDP ブロードキャスト応答。
- データベースサーバへの接続を行うための TCP/IP 接続呼び出し。
- 切断時のデータベースサーバからの応答。

サーバで TCP/IP を使用しているときに、このプロトコルオプションは、どの程度の時間待機するかを指定します。

- ミラーリングまたは診断サーバが接続を行うために検索するときの UDP ブロードキャスト応答。
- 他のデータベースサーバへのミラーリングまたは診断接続を行うときの TCP/IP 接続呼び出し。
- ミラーリングまたは診断サーバから切断するときの応答。

データベースサーバに対して HTTP または HTTPS を使用しているときに、このプロトコルは次を指定します。

- 要求を受け取るときに許可されている最大アイドル時間。Timeout 期間は、要求が開始するときに開始し、要求が完了するときに終了します。この制限に達した場合は、接続が閉じられて要求タイムアウトメッセージがクライアントに返されます。

### 例

次の接続文字列は、クライアントライブラリに対して、20 秒のタイムアウトで TCP/IP を使用してデータベースサーバに接続するように指示します。

```
CommLinks=tcPIP(TO=20)
```

## 関連情報

[KeepaliveTimeout \(KTO\) プロトコルオプション \[204 ページ\]](#)

## 1.1.7.39 trusted\_certificates プロトコルオプション

信頼できる証明書を1つ以上含むファイルのパスとファイル名を指定します。

### 構文

```
trusted_certificates=public-certificate
```

### 使用可能なプロトコル

TLS、HTTPS

### デフォルト

なし

### 備考

キーワード `trusted_certificate` も受け入れられます。証明書のファイルパスとファイル名を指定します。オペレーティングシステムの証明書ストアにある証明書を使用する場合は `*` を指定します。

`trusted_certificates` 暗号化プロトコルオプションを使用して、信頼できる証明書を指定します。データベースサーバは、`trusted_certificates` 暗号化プロトコルオプションを使用して、データベースサーバが信頼している機関によって署名されている証明書を使用している場合のみクライアントが接続できるようにします。同様に、クライアントは `trusted_certificates` 暗号化プロトコルオプションを使用して、適切なデータベースサーバに接続するようにします。信頼できる証明書は、サーバの自己署名証明書または民間認証局に属する証明書のいずれかです。FIPS 認定の RSA 暗号化を使用している場合は、RSA を使用して証明書を生成する必要があります。Encryption 接続パラメータで TLS を指定する場合、このプロトコルオプションは必須です。

TLS または HTTPS 接続を開始する際、データベースサーバライブラリは、RFC 2818 に記述された手順を使用して、データベースサーバのホスト名をサーバが提供する証明書と照らし合わせます。この照合が行われるのは、`certificate_name`、`certificate_company`、`certificate_unit` のいずれのオプションも指定されていない場合、あるいは、`skip_certificate_name_check` オプションが有効でない場合です。`certificate_name`、`certificate_company`、`certificate_unit` のいずれかが指定されていれば、それらのオプションだけが確認されます。`skip_certificate_name_check` オプションを有効にすると、ホスト名のチェックが無効になります。このオプションを有効にすると、クライアントはサーバの認証を完全に行うことができないので、攻撃に対して脆弱なままとなります。

ホスト名や IP アドレスは、`subjectAltName` (サブジェクトの別名、SAN) 拡張、および `Common Name (CN)` フィールドから取得します。SAN には、ワイルドカードによって複数のホスト名が指定されている場合があります。たとえば、Google 社の証明書には、`*.google.com`、`*.google.ca`、`*.android.com` が指定されているかもしれません。その場合、`www.google.ca` は有効なホスト名ということになります。

HTTPS は、Web サービスのクライアントプロシージャだけでサポートされています。

## 例

次の例は、証明書ファイルと証明書名を指定しています。

```
ENCRYPTION=TLS (trusted_certificate=myrootcert.crt;certificate_name=MyCertificateName)
```

この例では、クライアントライブラリで次のような処理が行われています。

- サーバの証明書署名チェーンのルートに myrootcert.crt が存在することを確認します。
- 証明書の通称フィールドに MyCertificateName が設定されていることを確認します。

次の例は、証明書ファイルのみを指定しています。

```
ENCRYPTION=TLS (trusted_certificate=myrootcert.crt)
```

この例では、クライアントライブラリで次のような処理が行われています。

- サーバの証明書署名チェーンのルートに myrootcert.crt が存在することを確認します。
- サーバの証明書にあるホスト名または IP アドレスのうちの 1 つが、接続しているホストのホスト名または IP アドレスに一致することを確認します。

次の例は、証明書ファイルを指定し、証明書名のチェックを省略するよう要求しています。

```
ENCRYPTION=TLS (trusted_certificate=myrootcert.crt;skip_certificate_name_check=true)
```

この例では、クライアントライブラリで次のような処理が行われています。

- サーバの証明書署名チェーンのルートに myrootcert.crt が存在することを確認します。
- これ以外の確認は行われません。

次の例は、証明書がオペレーティングシステムの証明書ストアにあり、証明書の通称が SAPNetCA であると指定しています。

```
ENCRYPTION=TLS (trusted_certificate=*;certificate_name=SAPNetCA)
```

次の例は、トランスポートレイヤセキュリティを使用して、Interactive SQL とデータベースサーバ demo とを接続します。

```
dbisql -c "UID=DBA;PWD=sql;Server=demo;Host=localhost;  
ENCRYPTION=TLS (fips=no;  
trusted_certificate=rsaroot.crt;  
certificate_name=RSA Server;  
certificate_company=SAP;  
certificate_unit=SQL Anywhere) "
```

## 関連情報

[SQL Anywhere クライアント/サーバ通信の暗号化 \[1653 ページ\]](#)

[クライアント側 TLS 証明書の使用 \[1652 ページ\]](#)

[certificate\\_company プロトコルオプション \(クライアント側のみ\) \[186 ページ\]](#)

[certificate\\_name](#) プロトコルオプション (クライアント側のみ) [187 ページ]  
[certificate\\_unit](#) プロトコルオプション (クライアント側のみ) [188 ページ]  
[Encryption \(ENC\)](#) 接続パラメータ [77 ページ]  
[証明書作成ユーティリティ \(createcert\)](#) [1079 ページ]  
[skip\\_certificate\\_name\\_check](#) プロトコルオプション (クライアント側のみ) [228 ページ]  
[identity](#) プロトコルオプション [201 ページ]  
[identity\\_password](#) プロトコルオプション [202 ページ]

## 1.1.7.40 VerifyServerName (VERIFY) プロトコルオプション (クライアント側のみ)

クライアントが接続前にデータベースサーバ名を確認する必要があるかどうかを制御します。

### 構文

```
{ VerifyServerName | VERIFY }={ YES | NO }
```

### 使用可能なプロトコル

TCP/IP (クライアント側のみ)

### デフォルト

YES

### 備考

クライアントが TCP/IP 接続に CommLinks (LINKS) 接続パラメータを使用し、正常に接続が確立できた場合、見つかったサーバ名と探しているサーバ名が同じであることが確認されます。ServerName (Server) 接続パラメータを使用してサーバ名を指定していないか、サーバ名が一致しない場合、接続は失敗します。VerifyServerName=NO を指定すると、サーバ名は検証されません。これにより、SQL Anywhere クライアントが SQL Anywhere サーバの IP アドレス/ポートしか知らない場合でも、そのポートに接続できます。

### i 注記

CommLinks (LINKS) 接続パラメータではなく Host 接続パラメータを使用することをお奨めします。Host 接続パラメータが使用され、ホスト名とポートの両方が指定されているとき、サーバ名を指定する必要はありません。ただし、正しいサーバへの接続が確実に確立されるように、常にサーバ名を指定することをお奨めします。

## 関連情報

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

## 1.1.8 データベースおよびデータベースサーバに接続できる限界数

データベースに接続するときは、データベースサーバ経由で接続します。データベースへの同時接続の最大数は、いくつかの要素によって異なります。

要素は次のとおりです。

- ライセンス。
- 実行を選択したデータベースサーバのタイプ。
- データベースサーバで実行されるデータベースの数。
- データベースとデータベースサーバが受け入れる接続のタイプ。
- アプリケーションの最大接続するを減らすために指定したデータベースおよびデータベースサーバオプション。

### データベースサーバに接続できる限界数

パーソナルデータベースサーバ (dbeng17) に接続できる限界数は 10 です。デフォルトでは、HTTP 接続となるのはこのうち 7 つの接続だけです。少なくとも 3 つの接続が標準接続用に予約されています。標準接続とは、HTTP または HTTPS 接続以外のあらゆる接続のことです。

ネットワークデータベースサーバに接続できる限界数は、パーシートライセンスにより許容される最大接続数と同じです。またはパーコアライセンスの場合は、無制限 (32766) です (実際には、オペレーティングシステムの制限のため、単一のデータベースサーバへの接続を 32766 作成することはできません)。さらに、データベースサーバも処理中に内部テンポラリ接続を作成するため、ユーザが使用可能な接続数が減少します)。

データベースサーバが受け入れ可能な同時接続の最大数を減らすには、-gm データベースサーバオプションを指定します。

データベースサーバが HTTP/HTTPS 接続を受け入れる場合、MaxConnections プロトコルオプションを指定することで、データベースサーバに作成できる HTTP/HTTPS 接続の最大数を制限することができます。reserved\_connections データベースオプションを設定することで、標準接続にのみデータベース接続を予約することもできます。

### データベース接続

データベースサーバ上で動作するデータベース数が多い場合には、データベースサーバへの接続数を制限するよりも、特定のデータベースへの接続数を制限するほうが効果的です。たとえば、接続が 100 まで許可されているデータベースサーバで、1 つのデータベースへの接続数が 90 に達すると、他のデータベースで利用できる接続は 10 だけしか残りません。

データベースが受け入れることができる接続数を制限するには、そのデータベースの `maximum_connections` データベースオプションを設定します。

データベースが HTTP/HTTPS 接続を受け入れるが、標準接続がいつでも受け入れられるようにする必要がある場合、標準接続用に接続を予約します。 `reserved_connections` データベースオプションを指定して、標準接続を予約します。

ログインポリシーを使用して、特定のログインポリシーを持つユーザに対する接続数を制限することもできます。ログインポリシーを作成する場合、 `max_connections` ログインポリシーオプションまたは `max_non_dba_connections` ログインポリシーオプションを設定します。または、 `login_procedure` データベースオプションを使用することで、カスタムログインプロシージャを作成して接続数を制限することができます。

## HTTP/HTTPS 接続キューと接続制限

データベースサーバは、HTTP/HTTPS 接続をキューに入れます。データベースサーバがその接続限界数に到達し、新しい HTTP/HTTPS 接続が試みられると、接続試行はキューに入れられます。接続試行は、HTTP/HTTPS 接続が使用可能になるか、150 秒後に試行がタイムアウトするまでキューに残ります。キューには、最大 1000 の接続試行を保持できます。

標準接続はキューに入れられません。したがって、データベースサーバの接続数が限界に達すると、標準接続の新しい試みは失敗します。

HTTP/HTTPS キューに接続試行が入っている間、HTTP/HTTPS 接続の代わりに標準接続を行うことはできません。標準接続用に接続が予約されている場合以外は、標準接続を行うとしているユーザは無限に待機することになります。データベースの接続の一部を標準接続に予約することで、ユーザがデータベースに接続し、接続の削除、バックアップの実行などの管理タスクを実行できるようにすることができます。

`reserved_connections` データベースオプションを使用することで、標準接続を予約します。ネットワークデータベースサーバは、標準接続のための接続の予約は行われません。パーソナルデータベースサーバでは、標準接続のために最低でも 3 つの接続が予約されています。

`MaxConnections` プロトコルオプションを指定することで、データベースサーバに行うことができる HTTP/HTTPS 接続の数を制限することもできます。

## 接続の限界数を超えた接続

接続の限界数に到達すると、DBA ユーザは必要に応じて限界数を超えてデータベースへの標準接続を 1 つ作成し、接続を強制的に削除することができます。この新しい接続は、次の条件がすべて満たされる場合にのみ行うことができます。

- ユーザが `DROP CONNECTION` または `SERVER OPERATOR` システム権限を持っている。
- データベースサーバの総接続数 (新しい接続を除く) が `MaxConnections` データベースサーバプロパティ制限以下になっている。
- データベースの総接続数 (新しい接続を除く) が `MaxConnections` データベースプロパティ制限以下になっている。
- 接続が標準接続である。

このセクションの内容:

### [データベース接続の制限 \[239 ページ\]](#)

`max_connections` データベースオプションを使用して、データベースへの同時接続数を制限します。

[標準接続の予約 \[240 ページ\]](#)

標準接続を受け入れ可能な接続の最小数を指定します。

## 関連情報

[接続リсна \[241 ページ\]](#)

[max\\_connections オプション \[725 ページ\]](#)

[reserved\\_connections オプション \[773 ページ\]](#)

[MaxConnections \(MAXCONN\) プロトコルオプション \[216 ページ\]](#)

[-gm データベースサーバオプション \[420 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[データベースプロパティの一覧 \[874 ページ\]](#)

[login\\_procedure オプション \[720 ページ\]](#)

[ログインポリシーオプションとデフォルト値 \[614 ページ\]](#)

### 1.1.8.1 データベース接続の制限

max\_connections データベースオプションを使用して、データベースへの同時接続数を制限します。

#### 前提条件

SET ANY SYSTEM OPTION システム権限が必要です。

#### コンテキスト

データベースサーバ上で動作するデータベース数が多い場合には、データベースサーバへの接続数を制限するよりも、特定のデータベースへの接続数を制限するほうが効果的です。たとえば、接続が 100 まで許可されているデータベースサーバで、1 つのデータベースへの接続数が 90 に達すると、他のデータベースで利用できる接続は 10 だけしか残りません。

max\_connections データベースオプションが設定されていない場合、データベースサーバの接続の限界数がデータベースに適用されます。

#### 手順

次のオプションのうちの 1 つを選択してください。

オプション	アクション
<b>max_connections</b> データベースオプションを永続的に設定します。	SET OPTION 文を実行し、データベースに対して許可する同時接続数を指定します。例: <pre>SET OPTION PUBLIC.max_connections = 25;</pre>
<b>max_connections</b> データベースオプションを一時的に設定します。	SET TEMPORARY OPTION 文を実行し、データベースに対して許可する同時接続数を一時的に制限します。指定された制限は、データベースが動作している限り有効です。例: <pre>SET TEMPORARY OPTION PUBLIC.max_connections = 25;</pre>

## 結果

データベースまたはデータベースサーバへの同時接続は、指定された数に制限されます。

## 関連情報

[接続、データベース、データベースサーバのプロパティ \[838 ページ\]](#)

[max\\_connections オプション \[725 ページ\]](#)

### 1.1.8.2 標準接続の予約

標準接続を受け入れ可能な接続の最小数を指定します。

## 前提条件

SET ANY SYSTEM OPTION システム権限が必要です。

## コンテキスト

HTTP/HTTPS の接続制限に到達すると、接続が使用可能になるまで新しい HTTP/HTTPS 接続がキューに入れられます。HTTP/HTTPS 接続がキューに入っている間、データベース接続の一部が標準接続用に予約されている場合以外、標準接続は成功しません。

データベースの接続の一部を標準接続に予約することで、ユーザがデータベースに接続し、接続の削除、バックアップの実行などの管理タスクを実行できるようにすることができます。

デフォルトでは、パーソナルデータベースサーバでは、標準接続のために最低でも 3 つの接続が予約されています。

ネットワークデータベースサーバでは、接続の予約は行われません。

## 手順

次のオプションのうちの 1 つを選択してください。

オプション	アクション
<b>reserved_connections</b> データベースオプションを永続的に設定します。	SET OPTION 文を実行し、標準接続用に予約する必要がある同時接続数を指定します。このオプションは、データベースで HTTP/HTTPS 接続がサポートされる場合のみ適用されます。例: <pre>SET OPTION PUBLIC.reserved_connections = 25;</pre>
<b>reserved_connections</b> データベースオプションを一時的に設定します。	SET TEMPORARY OPTION 文を実行し、標準接続用に予約する必要がある同時接続数を一時的に指定します。このオプションは、データベースで HTTP/HTTPS 接続がサポートされる場合のみ適用されます。指定された数は、データベースが動作している限り有効です。例: <pre>SET TEMPORARY OPTION PUBLIC.reserved_connections = 25;</pre>

## 関連情報

[接続、データベース、データベースサーバのプロパティ \[838 ページ\]](#)

[reserved\\_connections オプション \[773 ページ\]](#)

### 1.1.9 接続リスナ

接続リスナを使用すると、特定のネットワークプロトコルやアドレス、ポートを使用してデータベースサーバに接続できます。

接続リスナは、特定のアドレスで、指定したネットワークプロトコルからの接続の試行を受信します。データベースサーバでは、共有メモリ、HTTP、HTTPS、TCP/IP 用のリスナを使用できます。共有メモリ以外の接続リスナは、IP アドレスおよび特定のポートで受信します。データベースサーバの起動時に、-x または -xs データベースサーバオプションによって、どの種類の接続要求や Web 要求を受信するかをデータベースサーバに指定します。

データベースサーバを実行しているコンピュータの IP アドレスが変更された場合、データベースサーバはこれに対応するため必要に応じて接続リスナを自動的に起動または停止します。たとえば、データベースサーバを実行中のラップトップが新しい無線ネットワークに接続したとすると、データベースサーバは変更を検知して、新しい IP アドレスで接続リスナを起動します。また、ラップトップがネットワークとの接続を切断したら、データベースサーバは切断を検知して、該当する接続リスナを停止します。

システムプロシージャ `sp_start_listener` や `sp_stop_listener` を実行するか、あるいは SQL Central でデータベースサーバのリスナタブを使用すれば、データベースサーバを停止して再起動しなくても、接続リスナの起動や停止を手動で行うことができます。

このセクションの内容:

#### [接続リスナの作成 \(SQL Central\) \[242 ページ\]](#)

接続リスナを作成すると、データベースサーバ起動時の指定と異なるネットワークプロトコル、アドレス、ポート番号を使用して、データベースサーバに接続することができます。共有メモリ、TCP/IP、HTTP、HTTPS に対応する接続リスナを作成できます。

#### [接続リスナの削除 \(SQL Central\) \[243 ページ\]](#)

接続リスナを削除すると、指定したネットワークプロトコル、アドレス、ポート番号による新たな接続の受け入れができなくなります。既存の接続は影響を受けません。

## 1.1.9.1 接続リスナの作成 (SQL Central)

接続リスナを作成すると、データベースサーバ起動時の指定と異なるネットワークプロトコル、アドレス、ポート番号を使用して、データベースサーバに接続することができます。共有メモリ、TCP/IP、HTTP、HTTPS に対応する接続リスナを作成できます。

### 前提条件

sp\_start\_listener システムプロシージャの EXECUTE 権限、および、MANAGE LISTENERS システム権限が必要です。

### コンテキスト

TCP/IP 接続リスナは、データベースサーバ起動時に -ec データベースオプションで指定した暗号化設定を使用します。

共有メモリ接続リスナは、データベースサーバ起動時に指定した -ec データベースサーバオプションと関係なく作成できます。sp\_start\_listner を使用して共有メモリ接続リスナを起動すると、データベースサーバとの間で暗号化されない接続が許可されます。

### 手順

1. SQL Central では、SQL Anywhere 17 プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠でサーバを選択します。
3. 右ウィンドウ枠で、リスナタブをクリックします。
4. 右クリックして、**新規** **リスナ** をクリックし、ウィザードの指示に従います。

## 結果

接続リスナが作成されます。

### 1.1.9.2 接続リスナの削除 (SQL Central)

接続リスナを削除すると、指定したネットワークプロトコル、アドレス、ポート番号による新たな接続の受け入れができなくなります。既存の接続は影響を受けません。

## 前提条件

sp\_stop\_listener システムプロシージャの EXECUTE 権限、および、MANAGE LISTENERS システム権限が必要です。

## コンテキスト

## 手順

1. SQL Central では、SQL Anywhere 17 プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠でサーバを選択します。
3. 右側のウィンドウ枠で、停止するリスナをクリックします。
4. 右クリックして、**削除**をクリックします。

## 結果

接続リスナが削除されます。

### 1.1.10 接続 ID

ユーザがデータベースに接続すると、データベースサーバはその接続にユニークな接続 ID を割り当てます。

データベースサーバ接続は、チャンネルを形成します。クライアントアプリケーションからのアクティビティは、すべてそのチャンネルを介して行われます。接続が確立されるまで、クライアントアプリケーションはデータベースサーバと対話できません。データ

ベースサーバの接続が確立したら、ユーザの権限によって、ユーザがデータベースサーバに対して実行できる操作が決まります。

テンポラリ接続に対する接続 ID は、10 億番台から始まります。

データベースサーバに対して新しく接続するたびに、サーバは接続 ID 値を 1 つずつ増やします。これらの接続 ID は、データベースサーバメッセージログに書き込まれます。このログには、情報メッセージ、エラー、警告、MESSAGE 文からのメッセージが記録されます。接続 ID を使用して、次のタスクを実行できます。

- 要求ログ情報のフィルタリング
- データベースにロックがある接続の識別
- データベースサーバ起動後の合計接続数やその接続の順番の追跡

CONNECTION\_PROPERTY 関数を使用して Number 接続プロパティを問い合わせることによって、ユーザの接続 ID を取得できます。

## 関連情報

[詳細情報: テンポラリ接続 \[246 ページ\]](#)

[データベースサーバのロギング \[322 ページ\]](#)

[要求ロギング \[1430 ページ\]](#)

[-z データベースサーバオプション \[509 ページ\]](#)

### 1.1.11 接続プールによるアプリケーションパフォーマンスの向上

接続プールによって、データベースサーバに対して複数の短時間接続を行うアプリケーションのパフォーマンスが向上する可能性があります。

接続プールが接続に対して有効になっていると、接続は切断されたときに自動的にキャッシュされ、アプリケーションが再接続された時点で再利用できます。サーバ側の接続プールの場合、ConnectionPool (CPOOL) 接続パラメータを使用してプールを有効にします。同じ接続文字列を使用してアプリケーションで指定回数の接続を行うと、接続がプールされます。

.NET アプリケーションでの接続プールには、.NET の *Pooling* 接続パラメータを使用します。ConnectionPool (CPOOL) 接続パラメータは無視されます。.NET データプロバイダはプールされた接続をクリーンアップし、再初期化します。

接続をキャッシュするには、同じ接続文字列を使用してアプリケーションで最低でも 5 回の接続を行う必要があります。キャッシュされた接続を再利用するには、接続名はそのつど違っていてもかまいませんが、他のすべての接続パラメータは同じにする必要があります。

アプリケーションで接続を再度処理するときに、同じ接続文字列に対して使用できる接続がキャッシュされている場合は、キャッシュされた接続が再利用されます。接続は、ConnectionPool (CPOOL) 接続パラメータで指定された期間中、キャッシュされた状態のままになります (デフォルトでは 60 秒)。

接続プーリングは、統合ログインや Kerberos ログインなどの非標準のデータベース認証では行われません。ユーザ ID とパスワードの認証のみがサポートされます。

アプリケーションの動作に変更があると、キャッシュされた接続は再利用されません。たとえば、キャッシュされた接続は、次の場合に再利用されません。

- 接続が存在しないときに自動的に停止するデータベースの場合。
- 接続が無効になっている場合。
- データベースサーバが接続制限値に達した場合。
- パスワードが変更された場合。
- login\_mode オプションが設定されている場合。

アプリケーションに対して接続プーリングを透過的にするため、接続のキャッシュ時に障害が発生した場合には接続が切断されます。キャッシュされた接続を再利用しようとするときに障害が発生すると、データベースサーバで通常の接続が行われず。

接続が切断され、CPOOL 接続パラメータで指定された最大接続数に達していない場合は、接続がキャッシュされます。接続は再初期化されます。アプリケーションによって接続が切断されていても、キャッシュされた接続はデータベースサーバに接続したままになります。接続のクリーンアップと再初期化には、次の処理が含まれます。

- 未処理のすべてのトランザクションのロールバック。
- テンポラリテーブル、テンポラリ関数、変数の削除。
- 接続オプションと接続カウンターのリセット。
- データベースサーバの接続カウントの増分と減分。キャッシュされた接続だけしかないデータベースサーバが停止した場合、アクティブな接続があることは通知されません。
- 定義済みのすべての切断イベントと接続イベントの実行。
- login\_procedure データベースオプションの実行とログインポリシーの検証。
- 接続 ID のリセット。

## 他の接続プール製品での SQL Anywhere 接続プールの使用

接続プールをサポートしている製品または API を使用している場合、その製品または API の接続プールは SQL Anywhere の接続プールよりも優先されます。どちらのタイプの接続プールも、同時にアクティブにすることができます。

使用中の製品または API の接続プールの動作は、SQL Anywhere の接続プールと大幅に異なる場合があります。使用中の製品または API の接続プールの動作がアプリケーションに適していない場合は、SQL Anywhere の接続プールを使用して、アプリケーションのパフォーマンスを向上させることができます。

## 接続プールと読み込み専用スケールアウト

接続に対して NodeType (NODE) 接続パラメータも指定されている場合、通常、アプリケーションはプライマリサーバに接続し、プライマリサーバは負荷が最も低いコピーノードを決定します。その時点で、そのノードに接続がリダイレクトされます。このような接続と切断をアプリケーションで短時間に何度も行うと、接続がプールされ、どのコピーノードを使用するかプライマリサーバに問い合わせる必要がなくなります。この動作により、プライマリサーバでの負荷は削減されますが、予期する動作にならない可能性があります。

## 関連情報

[接続パラメータと接続文字列 \[34 ページ\]](#)

## 1.1.12 詳細情報: テンポラリ接続

データベースサーバでは、テンポラリ接続を使用して、バックアップの実行やデータベースの初期化などの操作が行われます。

sa\_conn\_info または sa\_conn\_list システムプロシージャを使用して、テンポラリ接続に関する情報を取得できます。ParentConnection プロパティでは、テンポラリ接続を作成した接続の接続 ID が返されます。

テンポラリ接続には、10 億 (1000000000) より大きい接続 ID と接続の機能を説明する名前があります。

次の例では、sa\_conn\_info システムプロシージャを使用して、どの接続がテンポラリ接続を作成したかを示す結果セットを返します。

```
SELECT Number, Name, ParentConnection FROM sa_conn_info();
```

接続 8 では、CREATE DATABASE 文を実行するテンポラリ接続が作成されます。

Number	Name	ParentConnection
1000000048	INT: CreateDB	8
9	SQL_DBC_14675af8	(NULL)
8	SQL_DBA_152d5ac0	(NULL)

## 1.1.13 トラブルシューティング: 接続

接続の確立方法を理解すれば、接続の問題の解決に役立ちます。

通信プロトコルやネットワーク特有の問題 (ファイアウォール経由の接続など) についての知識が必要です。

接続を確立するために、ソフトウェアは次のアクションを行います。

1. インタフェースライブラリを検出します。
2. 接続パラメータリストをアセンブルします。
3. データベースサーバを検出します。データベースサーバが検出されない場合は、パーソナルサーバが起動されます。
4. データベースを検出します。

接続手順は、次の場合も同じです。

- SQLDriverConnect 関数を使用する ODBC アプリケーション。これは ODBC アプリケーションでの一般的な接続方法です。SAP PowerBuilder などのアプリケーション開発システムの多くは、このクラスのアプリケーションに属します。ODBC アプリケーションでは SQLConnect 関数も使用できます。
- Embedded SQL を使用し、データベースとの接続で推奨関数 (db\_string\_connect) を使用するクライアントアプリケーション。さらに、Embedded SQL アプリケーションと Interactive SQL では、SQL CONNECT 文も使用できます。CONNECT 文には、CONNECT AS と CONNECT USING の 2 つの形式があります。dbisqlc を含むすべてのデータベースユーティリティは、db\_string\_connect を使用します。

- ADO.NET を使用する任意の .NET アプリケーション。アプリケーションは新しい SAConnection オブジェクトを作成し、接続文字列をコンストラクタに渡すか、ConnectionString プロパティを設定します。次に、アプリケーションは SAConnection オブジェクトの Open メソッドを呼び出して接続します。
- ADODB Connection オブジェクトを使用する任意の ADO アプリケーション。Provider プロパティは、OLE DB ドライバの場所を指定するために使用されます。ConnectionString プロパティでは、DataSourceName の代わりに DataSource、UserID の代わりに User ID を使用することもできます。
- **SQL Anywhere JDBC** ドライバを使用して、Driver Manager.GetConnection メソッドのパラメータとして URL `jdbc:sqlanywhere:` の後に標準の接続文字列を渡すアプリケーション。

このセクションの内容:

[トラブルシューティング: インタフェースライブラリのロケーション \[248 ページ\]](#)

通常、この DLL または共有ライブラリのロケーションは、ユーザには見えません。

[トラブルシューティング: 接続の確立方法 \[249 ページ\]](#)

接続を試みるときに実行される一連のアクションがあります。

[トラブルシューティング: データベースサーバの検出方法 \[251 ページ\]](#)

SQL Anywhere は、実行中のサーバを検出したときに、必要なデータベースをそのサーバ上で検出または起動しようとします。

[トラブルシューティング: Broadcast Repeater ユーティリティによるデータベースサーバの検索方法 \[254 ページ\]](#)

他のサブネット上で実行されているデータベースサーバや、ファイアウォールの外側にあって UDP ブロードキャストが通常は届かないデータベースサーバを、Host 接続パラメータや LDAP を使用することなく検索します。

[トラブルシューティング: Host 接続パラメータによるデータベースサーバの検出方法 \[255 ページ\]](#)

Host 接続パラメータを使って TCP/IP 経由でデータベースサーバを検出する場合、クライアントはサーバを検出するためのいくつかの手順を実行します。

[トラブルシューティング: CommLinks=TCPIP 接続パラメータによるデータベースサーバの検出方法 \[257 ページ\]](#)

CommLinks 接続パラメータを使って TCP/IP 経由でサーバを検出しようとしている SQL Anywhere クライアントは、サーバを検出するためのいくつかの手順を実行します。

[トラブルシューティング: 高速接続のための sasrv.ini へのデータベースサーバアドレス情報のキャッシュ方法 \[261 ページ\]](#)

Host または CommLinks 接続パラメータを使用するとき、特定の名前のデータベースサーバを検索するために複数の TCP/IP 接続を試行するか、大規模ネットワーク経由でブロードキャストすることは、時間がかかる可能性があります。

[トラブルシューティング: 接続文字列のテスト方法 \(dbping\) \[261 ページ\]](#)

接続のトラブルシューティングには dbping ユーティリティを使用します。

[トラブルシューティング: Embedded SQL とネットワーク接続パフォーマンスのテスト方法 \(dbping\) \[262 ページ\]](#)

Embedded SQL 接続とネットワークのパフォーマンスに関する情報を取得するには、ping ユーティリティ (dbping) を使用し、-s または -st オプションを指定します。

[トラブルシューティング: クライアントとデータベースサーバで互換性のあるプロトコルオプション \[263 ページ\]](#)

クライアントとデータベースサーバが同じプロトコルを使用していることを確認してください。

[トラブルシューティング: 一般的な接続の問題と解決 \[264 ページ\]](#)

接続しようとしたときにデータベースサーバが見つからないというメッセージを受信した場合は、クライアントがネットワークでデータベースサーバを検索できていません。

## 関連情報

[通信プロトコル \[164 ページ\]](#)

[トラブルシューティング: データベースサーバの起動 \[974 ページ\]](#)

[ネットワーク通信のトラブルシューティング \[979 ページ\]](#)

### 1.1.13.1 トラブルシューティング: インタフェースライブラリのロケーション

通常、この DLL または共有ライブラリのロケーションは、ユーザには見えません。

#### ODBC ドライバのロケーション

ODBC では、インタフェースライブラリは ODBC ドライバとも呼ばれます。ODBC クライアントアプリケーションが ODBC ドライバマネージャを呼び出し、ドライバマネージャが SQL Anywhere ドライバを検出します。

ODBC ドライバマネージャは、ドライバを探すために指定されたデータソースを調べます。ODBC データソースアドミニストレータまたは dbdsn ユーティリティを使用してデータソースを作成すると、ユーティリティは ODBC ドライバに現在のロケーションを入力します。データソース情報は、Windows の場合はレジストリに格納され、UNIX の場合はシステム情報ファイル (デフォルトのファイル名は `.odbc.ini`) に格納されます。

#### Embedded SQL インタフェースライブラリのロケーション

Embedded SQL アプリケーションは、通常 Embedded SQL インポートライブラリに対してリンクされます。インタフェースライブラリは、アプリケーションの起動時に名前ロードされます。Embedded SQL インタフェースライブラリの名前は次のとおりです。

##### Windows

`dblib17.dll`

##### UNIX

`libdblib17` (オペレーティングシステム固有の拡張子が付きます)

#### OLE DB ドライバのロケーション

OLE DB プロバイダ DLL (`dboledb17.dll`) の検索には、レジストリのエントリに基づいたプロバイダ名 (SAOLEDB) が使用されます。エントリは、SAOLEDB のインストール時、または再登録時に作成されます。

## ADO.NET

ADO.NET プログラムは SQL Anywhere ADO.NET プロバイダ (`Sap.Data.SQLAnywhere.dll`) への参照を追加します。.NET データプロバイダ DLL は、インストール時に .NET グローバルアセンブリキャッシュ (GAC) に追加されます。

## JDBC ドライバのロケーション

アプリケーションの実行時は、Java パッケージ `sajdbc4.jar` がクラスパスに含まれている必要があります。システムでネイティブ DLL または共有オブジェクトを検出する必要があります。

### Windows

Windows では、現在のディレクトリ、システムパス、Windows ディレクトリ、`Windows¥system32` ディレクトリが検索されます。

### UNIX

UNIX では、システムパスとユーザライブラリパスが検索されます。

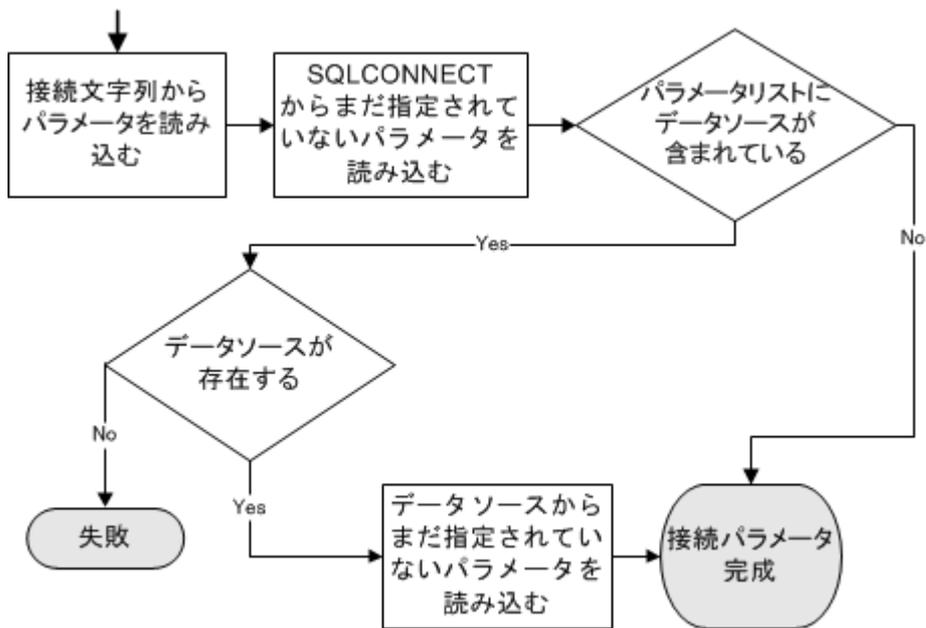
## ライブラリ検出時の処理

クライアントアプリケーションでインタフェースライブラリが検出されたらインタフェースライブラリに接続文字列が送信されます。この文字列は、接続パラメータのリストをアセンブルし、サーバ接続を確立するためにインタフェースライブラリで使用されます。

### 1.1.13.2 トラブルシューティング: 接続の確立方法

接続を試みるときに実行される一連のアクションがあります。

次の図は、インタフェースライブラリが接続パラメータのリストをアセンブルし、接続を確立する方法を示しています。



#### 優先度

複数の場所に格納されているパラメータは次の優先順位に従います。

1. 接続文字列
2. SQLCONNECT 環境変数
3. ODBC データソース

#### i 注記

接続文字列と ODBC データソースのどちらも同じ接続パラメータが指定されている場合は、接続文字列の値が使用され、データソースの値は無視されます。

#### 失敗

この段階での失敗は、接続文字列が不正な構文を使用している場合、不明な接続パラメータを使用している場合、または、接続文字列または SQLCONNECT で存在しないデータソースを指定した場合に発生する可能性があります。

#### 無視されるパラメータ

すでに使用中である他の接続に従い、次のようないくつかの接続パラメータは無視される可能性があります。

#### Autostop

データベースがすでにロードされている場合は無視されます。

#### DatabaseFile

DatabaseName が指定され、この名前を持つデータベースがすでに実行されている場合は無視されます。

インタフェースライブラリは、アセンブル後の接続パラメータのリストを使用して接続を試行します。

## 関連情報

[接続パラメータの構文ルール \[39 ページ\]](#)

### 1.1.13.3 トラブルシューティング: データベースサーバの検出方法

SQL Anywhere は、実行中のサーバを検出したときに、必要なデータベースをそのサーバ上で検出または起動しようとします。

SQL Anywhere は、実行中のサーバを検出できないと、接続パラメータに応じてパーソナルサーバを起動しようとする場合があります。

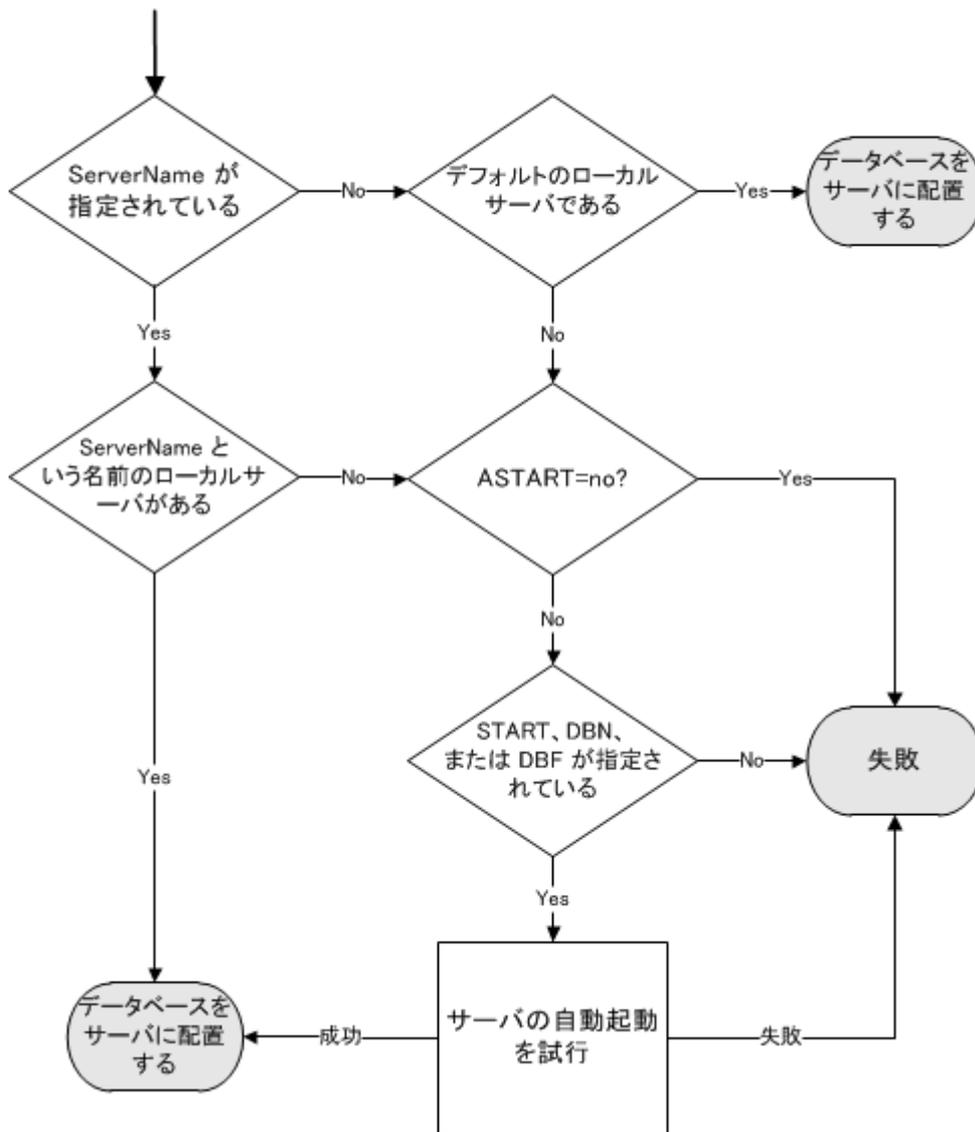
#### i 注記

- ローカル接続では、サーバの検出は簡単です。ネットワーク経由の接続では、Host 接続パラメータを使用することをお奨めします。
- サーバが自動的に起動される場合は、START、DBF、DBKEY、DBS、DBN、Server、および AutoStop 接続パラメータの情報を使用して、そのサーバのオプションが構築されます。サーバは、Host または CommLinks=TCPIP 接続パラメータが指定されている場合は、自動的に起動しません。
- サーバに代替サーバ名がある場合、代替サーバ名を指定して起動したデータベースへの接続にのみ代替サーバ名を使用できます。そのデータベースサーバ上で実行中の他のデータベースに接続するのに代替サーバ名を使用することはできません。

次の図は、SQL Anywhere がデータベースサーバを検出する方法を示しています。

#### 接続文字列が **CommLinks** 接続パラメータまたは **Host** 接続パラメータを含まない

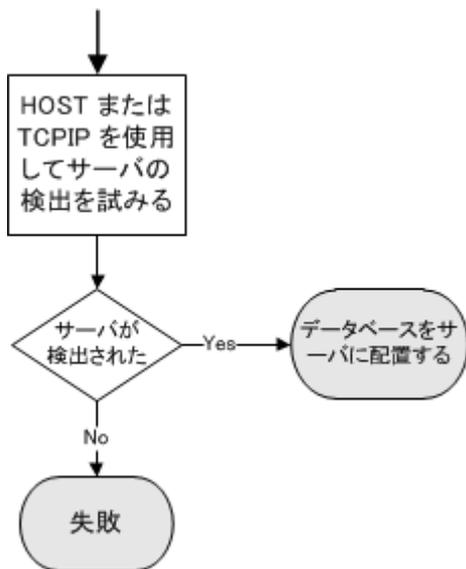
次の図は、接続文字列が CommLinks 接続パラメータまたは Host 接続パラメータを含まないときに、SQL Anywhere が共有メモリを使ってデータベースサーバを検出する方法を示しています。



SQL Anywhere は、実行中のサーバを検出できないと、接続パラメータに応じてパーソナルサーバを起動しようとする場合があります。

## Host 接続パラメータまたは CommLinks 接続パラメータを含む接続文字列は TCPIP を指定する

次の図は、接続文字列に CommLinks 接続パラメータまたは Host 接続パラメータのどちらかが含まれ、TCPIP を指定するときに、SQL Anywhere がデータベースサーバを検出する方法を示しています。



接続文字列には、**ShMem** および **TCPIP** に設定されている **CommLinks** 接続パラメータが含まれている

接続文字列に CommLinks=ShMem、TCPIP または CommLinks=TCPIP、ShMem が含まれている場合、SQL Anywhere は次の処理を行います。

- データベースサーバの自動起動を除く、共有メモリ接続を実行します。  
SQL Anywhere は、共有メモリを使用したサーバの検出に失敗した場合、TCPIP 接続を行おうとします。

## 関連情報

[トラブルシューティング: データベースサーバの検出方法 \[539 ページ\]](#)

[トラブルシューティング: Host 接続パラメータによるデータベースサーバの検出方法 \[255 ページ\]](#)

[-xd データベースサーバオプション \[504 ページ\]](#)

[サーバ列挙ユーティリティ \(dblocate\) \[1145 ページ\]](#)

[トラブルシューティング: Host 接続パラメータによるデータベースサーバの検出方法 \[255 ページ\]](#)

[トラブルシューティング: CommLinks=TCPIP 接続パラメータによるデータベースサーバの検出方法 \[257 ページ\]](#)

## 1.1.13.4 トラブルシューティング: Broadcast Repeater ユーティリティによるデータベースサーバの検索方法

他のサブネット上で実行されているデータベースサーバや、ファイアウォールの外側にあつて UDP ブロードキャストが通常は届かないデータベースサーバを、Host 接続パラメータや LDAP を使用することなく検索します。

### コンテキスト

Broadcast Repeater ユーティリティ (dbns17) を使用すると、他のサブネット上で実行されているデータベースサーバや、ファイアウォールの外側にあつて UDP ブロードキャストが通常は届かないデータベースサーバを、クライアントは Host 接続パラメータや LDAP を使用することなく検索できます。

相互通信が可能な DBNS プロセスの数に制限はありません。各 DBNS プロセスが検出したすべての DBNS プロセスに接続し、複数の DBNS プロセスが DBNS プロセスリストを共有します。たとえば、A と B の 2 つの DBNS プロセスを起動するとします。第 3 のサブネットで 3 番目の DBNS プロセス C を起動し、プロセス B のアドレスをプロセス C に渡すと、プロセス B がプロセス C にプロセス A の存在を通知し、プロセス C がプロセス A に接続します。

#### i 注記

1 つのサブネットで複数の DBNS プロセスを実行することは一般には不要であり、推奨できません。

1 つのサブネット内で複数の DBNS プロセスを実行しなければならない場合は、各 DBNS プロセスに対して異なるポートを指定する必要があります。1 つの DBNS プロセスが受信できるのは 1 つのポートからだけです。Broadcast Repeater ユーティリティ (dbns17) で -ap パラメータを使用します。

Host 接続パラメータまたは HOST プロトコルオプションのいずれかが使用される場合は、Broadcast Repeater ユーティリティは必要ありません。

### 手順

1. サブネット内の任意のコンピュータで DBNS (データベースネームサービス) プロセスを起動します。
2. 別のサブネット内の任意のコンピュータで DBNS プロセスを起動し、最初のコンピュータのコンピュータ名または IP アドレスをパラメータとして渡します。  
2 つの DBNS プロセスが TCP/IP 相互接続を確立します。
3. いずれの DBNS プロセスもそれぞれのサブネット上でブロードキャストを受信します。各 DBNS プロセスが TCP/IP 接続上で他方の DBNS プロセスに要求を送信し、受信した DBNS プロセスはそのサブネット上で要求を再ブロードキャストし、さらに送信元の DBNS プロセスに応答を送信します。送信元の DBNS プロセスは、受信した応答を要求元のクライアントに送信します。

## 結果

いずれかのサブネット上の通常のブロードキャストは、リモートサブネット上のデータベースサーバに到達し、クライアントは HOST パラメータを指定しなくてもリモートサブネット上のデータベースサーバに接続できます。

## 関連情報

[Broadcast Repeater ユーティリティ \(dbns17\) \[1077 ページ\]](#)

### 1.1.13.5 トラブルシューティング: Host 接続パラメータによるデータベースサーバの検出方法

Host 接続パラメータを使って TCP/IP 経由でデータベースサーバを検出する場合、クライアントはサーバを検出するためのいくつかの手順を実行します。

クライアントはデータベースサーバへの接続に成功すると、データベースに接続しようとします。データベースサーバが見つからない場合は、サーバが見つからなかったことを示すエラーが返されます。別のエラーが発生した場合、接続はそれ以上の手順を行うことなく失敗します。

クライアントは、次の手順でサーバを検出します。

1. データベースサーバアドレスキャッシュ (sasrv.ini) を確認する
2. Host 接続パラメータで指定されたアドレスに TCP/IP 接続を試みる
3. UDP データベースサーバ検索要求を送信する

#### 手順 1: データベースサーバアドレスキャッシュ (sasrv.ini) を確認する

クライアントは、ServerName 接続パラメータで指定されたデータベースサーバ名に一致するエントリをクライアント内の sasrv.ini ファイルで確認します。接続文字列に Host 接続パラメータは含まれているものの ServerName は含まれていない場合、クライアントは sasrv.ini ファイルを使用できません。

一致するキャッシュエントリが見つからない場合、クライアントは直接 TCP/IP 接続を試みます。

キャッシュエントリが ServerName 接続パラメータと、Host 接続パラメータで指定されたアドレスの両方に一致する場合、クライアントはキャッシュに示されている IP アドレスへの TCP/IP 接続を試みます。

- クライアントは、この IP アドレスを使ってデータベースサーバに接続する場合、データベースサーバの名前と ServerName で指定された値を比較します。  
データベースサーバ名が一致する場合、接続は成功します。  
データベースサーバ名が一致しない場合、クライアントは自身の sasrv.ini ファイルからエントリを削除し、直接 TCP/IP 接続を試みます。
- クライアントは、接続できない場合、自身の sasrv.ini ファイルからエントリを削除し、直接 TCP/IP 接続を試みます。

たとえば、ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピュータがあるとします。joey という名前のデータベースサーバがこのコンピュータのポート 49152 で実行中です。

接続文字列が `Host=kangaroo;ServerName=joey` の場合、クライアントはサーバ名 joey と一致するエントリを自身の `sasrv.ini` ファイルで検索します。キャッシュには IP アドレス 10.25.13.5:49152 が含まれています。クライアントは指定されたホスト名のアドレスを 10.25.13.5 と比較します。アドレスが一致するので、クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続してから、接続したサーバの名前が joey であることを確認します。接続は成功です。

## 手順 2: Host 接続パラメータで指定されたアドレスに TCP/IP 接続を試みる

複数のアドレスが指定された場合、クライアントは、Host 接続パラメータに指定された順序で各アドレスに接続しようとします。クライアントは、接続に成功するまで、またはすべてのアドレスを試してみるまで接続を試みます。

アドレスにポートが含まれる場合、クライアントは、そのアドレスと指定されたポート番号への TCP/IP 接続を試みます。ポートが指定されない場合、クライアントはデフォルトポートである 2638 で指定されたアドレスの TCP/IP 接続を試みます。

- クライアントは、データベースサーバに接続し、ServerName が指定された場合、データベースサーバの名前と ServerName で指定された値を比較します。  
データベースサーバ名が一致する場合、接続は成功です。クライアントは自身の `sasrv.ini` ファイルを更新します。  
データベースサーバ名が一致しない場合、クライアントは UDP サーバ検索要求を使って接続しようとします。
- クライアントがデータベースサーバに接続して ServerName が指定されない場合、接続は成功です。
- クライアントが接続できず、ServerName が指定された場合、クライアントは UDP サーバ検索要求を使って接続しようとします。
- クライアントが接続できず、ServerName が指定されない場合、接続は失敗します。

たとえば、ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピュータがあるとします。joey という名前のデータベースサーバがこのコンピュータのポート 49152 で実行中です。クライアントがこのサーバに接続するのは初めてなので、`sasrv.ini` ファイルにはキャッシュされたアドレスがありません。

接続文字列が `Host=kangaroo:49152;ServerName=joey` の場合、クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続してから、接続したサーバの名前が joey であることを確認します。接続は成功なので、クライアントはその `sasrv.ini` ファイルを更新します。

## 手順 3: UDP データベースサーバ検索要求を送信する

SQL Anywhere クライアントは、接続文字列に ServerName 接続パラメータが含まれており、Host 値がポートを指定しないときにだけ、UDP データベースサーバ検索要求を送信します。UDP サーバ検索要求は、指定されたアドレスのポート 2638 に送信されます。

デフォルトでは、同じコンピュータ上のすべてのデータベースサーバがポート 2638 で UDP パケットを受信します。データベースサーバが UDP サーバ検索要求を受信すると、データベースサーバは、その名前と UDP 要求で指定された名前を比較します。名前が一致する場合、データベースサーバはその IP アドレスとポート番号を含む UDP 応答パケットを送信し戻します。一致するデータベースサーバだけが応答パケットを送信します。

デフォルトでは、クライアントは UDP 応答を 5 秒間待ちます。応答が受信されない場合、このプロセスがタイムアウトするまで UDP パケットが毎秒再送信されます。

クライアントが UDP 応答パケットを受信すると、クライアントはパケットで指定されたアドレスとポートへの TCP/IP 接続を試みます。

クライアントがデータベースサーバに接続できれば、接続は成功です。クライアントは自身の `sasrv.ini` ファイルを更新します。

#### **i** 注記

UDP パケットは次のデータベースサーバを検出できません。

- `-sb 0` オプションが指定されたデータベースサーバ。
- ファイアウォール、ルータ、またはゲートウェイによってブロックされる UDP 要求の送信元であるデータベースサーバ。

これらの場合はいずれも、Host 接続パラメータでデータベースサーバのポートを指定すれば、クライアントは接続に成功します。

たとえば、ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピュータがあるとします。joey という名前のデータベースサーバがこのコンピュータのポート 49152 で実行中です。クライアントがこのサーバに接続するのは初めてなので、`sasrv.ini` ファイルにはキャッシュされたアドレスがありません。

接続文字列 `Host=kangaroo;ServerName=joey` において、クライアントは TCP/IP 経由で 10.25.13.5:2638 に接続しようとして、失敗しました (手順 2)。接続文字列にはポート番号が指定されていないので、クライアントは UDP サーバ検索要求をアドレス 10.25.13.5:2638 に送信します。データベースサーバ joey は UDP アドレス 10.25.13.5:2638 で受信し、アドレス 10.25.13.5:49152 を返します。クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続します。接続は成功なので、クライアントはその `sasrv.ini` ファイルを更新します。

## 関連情報

[ファイアウォール接続 \[169 ページ\]](#)

[トラブルシューティング: データベースサーバの検出方法 \[539 ページ\]](#)

[トラブルシューティング: 高速接続のための sasrv.ini へのデータベースサーバアドレス情報のキャッシュ方法 \[261 ページ\]](#)

[-sb データベースサーバオプション \[468 ページ\]](#)

[ServerPort \(PORT\) プロトコルオプション \[225 ページ\]](#)

### 1.1.13.6 トラブルシューティング: CommLinks=TCPIP 接続パラメータによるデータベースサーバの検出方法

CommLinks 接続パラメータを使って TCP/IP 経由でサーバを検出しようとしている SQL Anywhere クライアントは、サーバを検出するためのいくつかの手順を実行します。

データベースサーバが見つからない場合は、サーバが見つからなかったことを示すエラーが返されます。別のエラーが発生した場合、接続はそれ以上の手順を行うことなく失敗します。クライアントはデータベースサーバへの接続に成功すると、データベースに接続しようとします。

CommLinks (LINKS) 接続パラメータは、HOST または ServerPort (PORT) 以外の TCP/IP プロトコルオプションを指定する必要がある場合にのみ使用してください。これ以外の場合は、Host 接続パラメータを使用します。

クライアントが実行する手順は次の通りです。

## 手順 1: データベースサーバアドレスキャッシュ (`sasrv.ini`) を確認する

接続文字列に HOST プロトコルオプションが含まれているが、ServerName 接続パラメータが含まれていない場合、LocalOnly プロトコルオプションが YES に設定されている場合、または DoBroadcast プロトコルオプションが NONE に設定されている場合、クライアントは直接 TCP/IP 接続を試みます。つまり、クライアントは、データベースサーバアドレスキャッシュも LDAP サーバも使用しません。

クライアントは、ServerName 接続パラメータで指定されたデータベースサーバ名に一致するエントリをクライアント内の `sasrv.ini` ファイルで確認します。接続文字列に HOST プロトコルオプションも含まれる場合、キャッシュエントリは指定されたホストアドレスにも一致する必要があります。

一致するキャッシュエントリが見つからない場合、クライアントは LDAP を使ってデータベースサーバを検出しようとします。

一致するキャッシュエントリが見つかった場合、クライアントはキャッシュでリストされている IP アドレスに TCP/IP 接続を試みます。

- クライアントがこの IP アドレスを使ってデータベースサーバに接続し、VerifyServer プロトコルオプションが No に設定される場合、接続は成功します。
- クライアントは、この IP アドレスを使ってデータベースサーバに接続し、VerifyServer が Yes (デフォルト) に設定される場合、クライアントはデータベースサーバの名前と ServerName で指定された値を比較します。データベースサーバ名が一致する場合、接続は成功します。データベースサーバ名が一致しない場合、クライアントは自身の `sasrv.ini` ファイルからエントリを削除し、LDAP を使ってデータベースサーバを検出しようとします。
- 接続できない場合、クライアントは自身の `sasrv.ini` ファイルからエントリを削除し、LDAP を使ってデータベースサーバを検出しようとします。

たとえば、ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピュータがあるとします。joey という名前のデータベースサーバがこのコンピュータのポート 49152 で実行中です。

接続文字列が `CommLinks=TCPIP (Host=kangaroo) ; ServerName=joey` の場合、クライアントはサーバ名 joey と一致するエントリを自身の `sasrv.ini` ファイルで検索します。キャッシュには IP アドレス 10.25.13.5:49152 が含まれています。クライアントは指定されたホスト名のアドレスを 10.25.13.5 と比較します。アドレスが一致するので、クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続してから、接続したサーバの名前が joey であることを確認します。接続は成功です。

## 手順 2: サーバ名について LDAP サーバを確認する

SQL Anywhere クライアントとデータベースサーバのコンピュータが LDAP サーバを使用するように設定されている場合、クライアントは、ServerName 接続パラメータに一致するエントリを LDAP サーバに問い合わせます。

## 手順 3: HOST プロトコルオプションで指定されたアドレスに TCP/IP 接続を試みる

HOST プロトコルオプションが指定されない場合、クライアントは UDP サーバ検索要求を使ってサーバを検出しようとします。

クライアントは、TCP/IP を経由して HOST プロトコルオプションに指定されたアドレスに接続しようとします。複数のアドレスが指定された場合、クライアントは、HOST プロトコルオプションに指定された順序で各アドレスに接続しようとします。クライアントは、接続に成功するまで、またはすべてのアドレスを試してみるまで接続を試みます。

接続文字列がポートを含むとき (HOST プロトコルオプションまたは ServerPort 接続プロパティのいずれか)、クライアントは指定されたアドレスとポート番号への TCP/IP 接続を試みます。ポートが指定されない場合、クライアントはデフォルトポートである 2638 で指定されたアドレスの TCP/IP 接続を試みます。

- クライアントが指定された IP アドレスを使ってデータベースサーバに接続し、VerifyServer プロトコルオプションが No に設定される場合、接続は成功します。
- クライアントは、指定されたアドレスを使ってデータベースサーバに接続した場合、データベースサーバの名前と ServerName で指定された値を比較します。データベースサーバ名が一致する場合、接続は成功し、クライアントは自身の sasrv.ini ファイルを更新します。データベースサーバ名が一致しない場合、クライアントは UDP サーバ検索要求を使って接続しようとします。
- クライアントが接続できず、ServerName が指定された場合、クライアントは UDP サーバ検索要求を使って接続しようとします。

たとえば、ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピュータがあるとします。joey という名前のデータベースサーバがこのコンピュータのポート 49152 で実行中です。クライアントがこのサーバに接続するのは初めてなので、sasrv.ini ファイルにはキャッシュされたアドレスがありません。

接続文字列が CommLinks=TCPIP (Host=kangaroo:49152);ServerName=joey の場合、クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続してから、接続したサーバの名前が joey であることを確認します。接続は成功し、クライアントは自身の sasrv.ini ファイルを更新します。

## 手順 4: UDP データベースサーバ検索要求を送信する

SQL Anywhere クライアントは、接続文字列に ServerName 接続パラメータが含まれており、DoBroadcast プロトコルオプションは DIRECT または ALL に設定されているときにだけ、UDP データベースサーバ検出要求を送信します。

- DoBroadcast=DIRECT または HOST プロトコルオプションが指定されたとき、クライアントは UDP サーバ検索パケットを指定されたアドレスに送信します。ポートが指定されたとき (HOST プロトコルオプションまたは ServerPort プロトコルオプションのいずれかで)、UDP サーバ検索要求は指定されたポートに送信されます。それ以外の場合、要求はデフォルトポート 2638 に送信されます。
- DoBroadcast=ALL が指定され、HOST プロトコルオプションが指定されていない場合、クライアントは、それぞれの IP アドレスに対するサブネットのブロードキャストアドレスを決定します。ServerPort プロトコルオプションが指定された場合、クライアントは UDP ブロードキャストサーバ検索パケットを指定のポートに送信します。それ以外の場合、UDP ブロードキャストパケットはデフォルトのポート 2638 に送信されます。

すべてのデータベースサーバは、起動時に -sb 0 オプションが指定されていない限り、ポート 2638 で UDP パケットを受信します。データベースサーバは、UDP サーバ検索要求を受信すると、自身の名前と要求パケットに指定された名前を比較します。名前が一致する場合、データベースサーバはその IP アドレスとポート番号を含む UDP 応答パケットを送信し戻します。一致するデータベースサーバだけが応答パケットを送信します。

デフォルトでは、クライアントは UDP 応答を 5 秒間待ちます。応答が受信されない場合、このプロセスがタイムアウトするまで UDP パケットが毎秒再送信されます。

クライアントが UDP 応答パケットを受信すると、クライアントはパケットで指定されたアドレスとポートへの TCP/IP 接続を試みます。

クライアントがデータベースサーバに接続できれば、接続は成功です。クライアントは自身の `sasrv.ini` ファイルを更新しません。

### i 注記

UDP パケットは次のデータベースサーバを検出できません。

- 異なるサブネット上に存在するデータベースサーバ。UDP サーバ検出ブロードキャストパケットは、Broadcast Repeater ユーティリティが使用されない限り、データベースサーバを検索できません。
- sb 0 オプションが指定されたデータベースサーバ。
- ファイアウォール、ルータ、またはゲートウェイによってブロックされる UDP 要求の送信元であるデータベースサーバ。

これらの場合、HOST プロトコルオプションにデータベースサーバのアドレス (ポート番号を含む) を指定すれば、クライアントは接続に成功します。

たとえば、ホスト名が kangaroo、IP アドレスが 10.25.13.5 のコンピュータがあるとします。joey という名前のデータベースサーバがこのコンピュータのポート 49152 で実行中です。クライアントがこのサーバに接続するのは初めてなので、`sasrv.ini` ファイルにはキャッシュされたアドレスがありません。接続は kangaroo と同じサブネット上にあるコンピュータから試行され、このサブネットのブロードキャストアドレスは 10.25.13.255 です。

接続文字列が `CommLinks=TCPIP;ServerName=joey` の場合、クライアントは UDP サーバ検出ブロードキャスト要求を、ブロードキャストアドレス 10.25.13.255:2638 に送信します。データベースサーバ joey は UDP アドレス 10.25.13.5.2638 で受信し、アドレス 10.25.13.5:49152 を返します。クライアントは TCP/IP 経由で 10.25.13.5:49152 に接続します。接続は成功なので、クライアントはその `sasrv.ini` ファイルを更新します。

## 関連情報

[ネームサーバとして LDAP を使用した接続 \[171 ページ\]](#)

[ファイアウォール接続 \[169 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[トラブルシューティング: データベースサーバの検出方法 \[539 ページ\]](#)

[トラブルシューティング: 高速接続のための sasrv.ini へのデータベースサーバアドレス情報のキャッシュ方法 \[261 ページ\]](#)

[VerifyServerName \(VERIFY\) プロトコルオプション \(クライアント側のみ\) \[236 ページ\]](#)

[Broadcast Repeater ユーティリティ \(dbns17\) \[1077 ページ\]](#)

[ServerPort \(PORT\) プロトコルオプション \[225 ページ\]](#)

[-sb データベースサーバオプション \[468 ページ\]](#)

## 1.1.13.7 トラブルシューティング: 高速接続のための `sasrv.ini` へのデータベースサーバアドレス情報のキャッシュ方法

Host または CommLinks 接続パラメータを使用するとき、特定の名前のデータベースサーバを検索するために複数の TCP/IP 接続を試行するか、大規模ネットワーク経由でブロードキャストすることは、時間がかかる可能性があります。

データベースサーバアドレスのキャッシュは、後で接続するときに使用できるようにデータベースサーバ名と IP アドレスを `sasrv.ini` ファイルに保存することによって、ネットワーク接続を高速化します。

キャッシュファイルには一連のセクションがあり、それぞれが次の形式になっています。

```
[Server name]
LINK=TCPIP
Address=IP_address:port
```

### i 注記

それぞれのデータベースサーバの名前がユニークであることは、重要です。異なるデータベースサーバに同じ名前を付けると、識別の問題を引き起こす可能性があります。

## `sasrv.ini` ファイルのロケーション

`sasrv.ini` ファイルのデフォルトロケーションは次のとおりです。

オペレーティングシステム	<code>sasrv.ini</code> のデフォルトロケーション
Windows	<code>%ALLUSERSPROFILE%\Application Data\SQL Anywhere17</code>
UNIX	<code>\$HOME/.sqlanywhere17</code>

## 関連情報

[トラブルシューティング: Host 接続パラメータによるデータベースサーバの検出方法 \[255 ページ\]](#)

[トラブルシューティング: CommLinks=TCPIP 接続パラメータによるデータベースサーバの検出方法 \[257 ページ\]](#)

## 1.1.13.8 トラブルシューティング: 接続文字列のテスト方法 (dbping)

接続のトラブルシューティングには `dbping` ユーティリティを使用します。

`-c` オプション付きで `dbping` ユーティリティを使用すると、接続文字列パラメータをテストできます。デフォルトでは、`dbping` はデータベースには接続せず、サーバに接続するだけであり、サーバを起動しません。`dbping` ユーティリティに `-d` オプションを使用すると、データベースに接続します。必要に応じてデータベースサーバを起動します。

## 例

たとえば、demo17 という名前のパーソナルサーバでサンプルデータベース (コマンド `dbeng17 -n demo17 "%SQLANYSAM17%demo.db"` で起動できます) が実行中であるとして、ローカルコンピュータ上で demo17 という名前のデータベースサーバが実行中であり、demo という名前のデータベースが実行されている場合、次のコマンドは ping が成功したことを示すメッセージを返します。

```
dbping -d -c "Server=demo17;DBN=demo;UID=DBA;PWD=sql"
```

次のコマンドは、demo17 というデータベースサーバと、demo というデータベースが実行中かどうかをテストし、実行中ではなければサーバとデータベースを起動します。

```
dbping -d -c "Server=demo17;DBN=demo;DBF=c:¥sa17¥samples ¥demo.db;ASTOP=no;UID=DBA;PWD=sql"
```

次のコマンドは、デフォルトポート 2638 の Waterloo という名前のコンピュータでデータベースサーバを検出できるかどうかをテストします。

```
dbping -c "Host=Waterloo"
```

次のコマンドは、デフォルトデータベースサーバが現在のコンピュータ上で実行中かどうかをテストします。

```
dbping
```

## 関連情報

[Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

[トラブルシューティング: Embedded SQL とネットワーク接続パフォーマンスのテスト方法 \(dbping\) \[262 ページ\]](#)

### 1.1.13.9 トラブルシューティング: Embedded SQL とネットワーク接続パフォーマンスのテスト方法 (dbping)

Embedded SQL 接続とネットワークのパフォーマンスに関する情報を取得するには、ping ユーティリティ (dbping) を使用し、`-s` または `-st` オプションを指定します。

次の統計値が収集されます。

統計情報	説明
DBLib の接続と切断	DBLIB の接続および切断を 1 度実行する時間。ODBC などの他のインタフェースを使用した接続および切断は、接続の完了に必要な要求が多いため、一般的にパフォーマンスは DBLIB より低くなります。

統計情報	説明
簡単な要求の往復時間	クライアントからサーバに要求を送信するのにかかる時間とサーバからクライアントに応答を送信するのにかかる時間との和。往復時間は平均遅延時間の 2 倍になります。
送信スループット	dbping からデータベースサーバへの反復ごとに 100 KB のデータを転送する場合のスループット。
受信スループット	データベースサーバから dbping への反復ごとに 100 KB のデータを転送する場合のスループット。

往復時間が長く、スループットが高いネットワークの場合、往復時間が長いために、レポートされるスループットはネットワークの実際のスループットより低くなります。通信圧縮によりパフォーマンスが向上するかどうかを確認するには、dbping -s を使用すると便利です。パフォーマンス統計は概算値であり、クライアントコンピュータとサーバコンピュータの両方がアイドル状態である方が統計値の精度は高くなります。通信圧縮を使用すると、送信されるデータは元のサイズの約 25% に圧縮されます。

dbping -s コマンドからの出力例を次に示します。次の dbping コマンドを実行しました。

```
dbping -s -c "UID=DBA;PWD=sql;Server=demo17;DBN=demo"
```

```
SQL Anywhere Server Ping ユーティリティ バージョン 17.0.4.1691
SQL Anywhere 17.0.4.1691 サーバ "demo17" とデータベース "demo" に接続しました。
パフォーマンス統計      数値      合計時間      平均
-----
DBLib の接続と切断      200 回      512 ミリ秒      2 ミリ秒
単純要求のラウンドトリップ      22100 要求      1024 ミリ秒      <1 ミリ秒
送信スループット      267500 KB      1024 ミリ秒      261230 KB/秒
受信スループット      230400 KB      1024 ミリ秒      225000 KB/秒
データベースへの ping が成功しました。
```

## 関連情報

[Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

[トラブルシューティング: 接続文字列のテスト方法 \(dbping\) \[261 ページ\]](#)

### 1.1.13.10 トラブルシューティング: クライアントとデータベースサーバで互換性のあるプロトコルオプション

クライアントとデータベースサーバが同じプロトコルを使用していることを確認してください。

#### 共有メモリプロトコル

クライアントとデータベースサーバが同じコンピュータ上にある場合、共有メモリ (デフォルトプロトコル) をおすすめます。

#### TCP/IP プロトコル

クライアントとデータベースサーバが異なるコンピュータ上にある場合は、TCP/IP を使用する必要があります。ネットワークサーバ (dbsrv17) では、デフォルトでは TCP/IP 接続を利用できます。

クライアントは、Host または CommLinks 接続パラメータを使用して、データベースサーバへの TCP/IP 接続を確立します。

パーソナルサーバ (dbeng17) では、デフォルトで TCP/IP 接続を利用できません。パーソナルサーバは、-x tcpip オプションを指定することによって、ローカルの (ただし、リモートは不可) TCP/IP 接続を受け付けることができます。

## 関連情報

[-x データベースサーバオプション \[500 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

### 1.1.13.11 トラブルシューティング: 一般的な接続の問題と解決

接続しようとしたときにデータベースサーバが見つからないというメッセージを受信した場合は、クライアントがネットワークでデータベースサーバを検索できていません。

次のような問題がないか調べてください。

- 接続文字列にサーバ名を指定します。  
コンピュータシステムでは複数のデータベースサーバを実行できるため、データベースへの接続時には必ず、データベースサーバ名を指定してください (ServerName=`server-name`)。
- リモートデータベースサーバに対しては、Host 接続パラメータを指定し、TCP/IP ホスト名またはデータベースサーバを実行中のコンピュータのアドレスを提供することをお奨めします。指定したアドレスが正しいことを確認します。サーバがデフォルトの 2638 以外のポートを使用している場合、Host=`tcpip-address:port-number` を指定する必要があります。  
詳細 CommLinks 接続パラメータを使用している場合、データベースサーバの検索に使用されるパケットが、データベースサーバに到達しない可能性があります。Host、ServerPort、または DoBroadcast 以外の TCP/IP プロトコルオプションを使用する必要がある場合を除いて、Host 接続パラメータを使用することをお奨めします。HOST プロトコルオプションを指定せずに CommLinks 接続パラメータを使用している場合、UDP ブロードキャストまたは要求は、現在のサブネットに限定されるか、ルータ、ゲートウェイ、またはファイアウォールによってブロックされます。
- クライアントとサーバの間にファイアウォールがあり、接続が妨害されている可能性があります。
- パーソナルサーバは同じコンピュータからの接続だけを受け入れます。クライアントとサーバが異なるコンピュータ上にある場合は、ネットワークサーバを使用する必要があります。
- ネットワークドライバが正しくインストールされていないか、またはネットワーク配線が正しくありません。クライアントコンピュータがサーバコンピュータと通信できるかどうかを確認するには、ping ユーティリティを使用します。
- SAP Open Client または jConnect 経由で接続している場合、サーバが TCP/IP プロトコルを使用する必要があります。

## 関連情報

[ファイアウォール接続 \[169 ページ\]](#)

[ネットワークプロトコルオプション \[177 ページ\]](#)

[トラブルシューティング: 接続 \[246 ページ\]](#)

[トラブルシューティング: データベースサーバの起動 \[974 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

## 1.2 データベースの作成

データベースを作成するには、データベースに格納されるテーブル (エンティティ)、各テーブルのカラム (属性)、テーブル間の関係 (キーと制約) を定義します。

SQL Central、初期化ユーティリティ (dbinit)、CREATE DATABASE 文を使用して、データベースの作成や初期化を行うことができます。データベースを作成したら、そのデータベースに接続して、テーブルやその他のオブジェクトを追加できます。

### トランザクションログ

データベースを作成するには、トランザクションログを配置する場所を決定しなければなりません。このログには、データベースへ加えられた変更内容が、変更された順に格納されます。データベースファイルでメディア障害が発生した場合、トランザクションログはデータベースのリカバリに重要な役割を果たします。また、トランザクションログを使用すると、より効率的に作業できます。デフォルトでは、トランザクションログはデータベースファイルと同じディレクトリに配置されますが、この設定での運用はおすすめしません。

### データベースファイルの互換性

ファイルシステムのファイルサイズが制限される場合や、大きなファイルがサポートされる場合以外は、すべてのオペレーティングシステム間でデータベースファイルには互換性があります。

オペレーティングシステム間の互換は、データベースファイルをコピーすれば可能です。同様に、パーソナルデータベースサーバ (dbeng17) で作成されたデータベースは、ネットワークデータベースサーバ (dbsrv17) で使用できます。データベースサーバは、旧バージョンのソフトウェアで作成されたデータベースを管理できますが、旧バージョンのサーバは新バージョンのデータベースを管理できません。

このセクションの内容:

[データベースの作成 \(SQL Central の場合\) \[266 ページ\]](#)

SQL Central を使用してデータベースを作成します。

[データベースの作成 \(dbinit ユーティリティの場合\) \[267 ページ\]](#)

初期化ユーティリティ (dbinit) を使用してデータベースを作成します。

#### [データベースファイルのタイプ \[268 ページ\]](#)

各データベースには、いくつかの関連ファイルがあります。

#### [カラムデータ型の考慮事項 \[302 ページ\]](#)

カラムにはいくつかのデータ型を使用できます。

#### [カラムの圧縮の考慮事項 \[304 ページ\]](#)

CHAR、VARCHAR、BINARY カラムを圧縮して、ディスク領域を節約できます。

#### [制約の考慮事項 \[305 ページ\]](#)

カラムのデータ型は許可されるデータ値を制限しますが (たとえば数字または日付のみなど)、データ値をさらに制限したい場合もあります。

## 関連情報

#### [トランザクションログ \[279 ページ\]](#)

#### [データベースのサイズおよび数に関する物理的な制限 \[891 ページ\]](#)

#### [ER \(実体関連\) 図の表示 \[1057 ページ\]](#)

## 1.2.1 データベースの作成 (SQL Central の場合)

SQL Central を使用してデータベースを作成します。

## 前提条件

デフォルトでは、SERVER OPERATOR システム権限が必要です。-gu データベースサーバオプションを使用すると、必要な権限を変更できます。

## 手順

1. SQL Central で、**ツール** > **SQL Anywhere17** > **データベースの作成** をクリックします。
2. データベース作成ウィザードの指示に従います。

## 結果

データベースが作成されます。

## 関連情報

[接続 ID \[243 ページ\]](#)

[データベースの作成 \(dbinit ユーティリティの場合\) \[267 ページ\]](#)

## 1.2.2 データベースの作成 (dbinit ユーティリティの場合)

初期化ユーティリティ (dbinit) を使用してデータベースを作成します。

### コンテキスト

このユーティリティを使用すると、コマンドラインオプションを含めて別のデータベース設定を指定できます。

トランザクションログミラーによって、重要なデータがさらに保護され、トランザクションログでメディア障害が発生した場合に完全なデータリカバリが可能になります。データベースの作成時にトランザクションログミラーを保持するには、-m オプションを指定します。

### 手順

dbinit コマンドを実行します。

### 結果

データベースが作成されます。

#### 例

たとえば、company.db というデータベースをページサイズ 4 KB で作成するには、次のコマンドを実行します。

```
dbinit -dba DBA,passwd -p 4k company.db
```

次のコマンド (1 行で入力) は、company.db という名前のデータベースを初期化し、トランザクションログを別のデバイスに保持し、ミラーを 3 つ目のデバイスに保持します。

```
dbinit -dba DBA,passwd -t d:¥log-dir¥company.log -m  
e:¥mirr-dir¥company.mlg c:¥db-dir¥company.db
```

## 関連情報

[接続 ID \[243 ページ\]](#)

[トランザクションログミラー \[288 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

### 1.2.3 データベースファイルのタイプ

各データベースには、いくつかの関連ファイルがあります。

#### データベースファイル

このファイルは、データベースの内容全体を保持します。1つのファイルで1つのデータベースを構成する方法と、最高12のDB領域を追加する方法があります。DB領域は、同じデータベースの一部を入れる追加ファイルです。データベースファイルとDB領域のロケーションを選択します。通常、データベースファイルの拡張子は .db です。

#### トランザクションログ

このファイルはデータベースの変更記録を格納するものであり、障害が発生した場合に、データベースの情報の同期とリカバリを行うのに必要です。通常、トランザクションログの拡張子は .log です。

#### ロールバックログ (取り消しログ)

データベースの内容に変更が加えられると、ロールバックログが作成されます。このログは、トランザクションがロールバックされた場合、またはシステム障害の発生時にトランザクションがコミットされていなかった場合に、変更をキャンセルするために使用されます。接続ごとに個別のロールバックログがあります。トランザクションのコミットまたはロールバックが行われると、その接続に関するロールバックログの内容は削除されます。ロールバックログは、データベースに格納されます。ロールバックログページは、変更される他のページとともにチェックポイントログにコピーされます。

#### テンポラリファイル

データベースサーバは、ソートや UNION 処理などの操作のため、キャッシュで使用できる領域よりも多くの領域が必要な場合、データベースセッション中にテンポラリファイルを使用します。こうした領域が必要な場合、データベースサーバは、通常はその領域を集中的に使用します。データベース全体のパフォーマンスは、テンポラリファイルを含むデバイスの速度に大きく依存します。データベースが停止すると、データベースサーバはたとえ実行中であっても、テンポラリファイルを破棄します。サーバが生成した名前に拡張子 .tmp を付けたものが、テンポラリファイル名です。

テンポラリファイルのロケーションは、データベースサーバの起動時に -dt サーバオプションを使用して指定できます。テンポラリファイルのロケーションを指定せずにデータベースサーバを起動した場合は、以下の環境変数がこの順序のとおりチェックされます。

- SATMP 環境変数
- TMP 環境変数
- TMPDIR 環境変数
- TEMP 環境変数

これらの環境変数がいずれも定義されていない場合、テンポラリファイルは、Windows オペレーティングシステムの場合は現在のディレクトリ、UNIX の場合は /tmp ディレクトリに作成されます。

テンポラリファイルは、データベースサーバが作成、管理、削除します。テンポラリファイル用に使用できる、十分な空き領域があることだけ確認してください。テンポラリファイルに使用できる領域に関する情報は、sa\_disk\_free\_space プロシージャを使用すると取得できます。

## 事前定義の DB 領域ファイル

これらのファイルには、データと、データベースで使用されるその他のファイルが格納されます。

### DB 領域ファイル

データベースファイルに加えて、データを複数の個別ファイルに分散できます。

### トランザクションログミラーファイル

データの保全のため、トランザクションログのミラーコピーを作成できます。通常、このファイルの拡張子は .mlg です。

このセクションの内容:

#### [事前定義の DB 領域 \[269 ページ\]](#)

データベースサーバは、データベースに事前定義の DB 領域を使用します。

#### [追加の DB 領域の考慮事項 \[270 ページ\]](#)

データベースファイルを追加すると、関連情報を個別のファイルにまとめることができます。

#### [DB 領域の削除 \(SQL の場合\) \[278 ページ\]](#)

SQL で DROP DBSPACE 文を使用して DB 領域を削除します。

#### [トランザクションログ \[279 ページ\]](#)

トランザクションログは、データベースへの変更をすべて保存するファイルです。たとえば、挿入、更新、削除、コミット、ロールバック、データベーススキーマの変更が、すべて記録されます。

#### [ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

ユーティリティデータベースは、物理的な実体を持たない幻のデータベースです。

#### [データベースの削除 \(SQL Central の場合\) \[300 ページ\]](#)

SQL Central でデータベースを削除します。

#### [データベースの削除 \(dberase ユーティリティの場合\) \[301 ページ\]](#)

dberase ユーティリティを使用して、データベースを削除します。

## 関連情報

[トランザクションログミラー \[288 ページ\]](#)

### 1.2.3.1 事前定義の DB 領域

データベースサーバは、データベースに事前定義の DB 領域を使用します。

DB 領域	名前
メインデータベースファイル	system
テンポラリファイル	temporary または temp
トランザクションログファイル	translog

DB 領域	名前
トランザクションログミラー	translogmirror

これらの名前を使用して、ユーザ定義の DB 領域を作成することはできません。また、事前定義の DB 領域を削除することはできません。

事前定義の DB 領域と同じ名前が付いたユーザ定義の DB 領域が含まれる、10.0.0 以前のバージョンのデータベースをアップグレードすると、SQL 文でのこれらの DB 領域へのすべての参照は、事前定義の DB 領域ではなく、ユーザ定義の DB 領域を参照していると見なされます。事前定義の DB 領域を参照するように設定するには、ユーザ定義の DB 領域を削除するか、ユーザ定義の DB 領域の名前を変更して事前定義の DB 領域とは別の名前を指定する方法しかありません。

ALTER DBSPACE 文は、事前定義の DB 領域の名前をサポートしているため、これらに対して領域を追加できます。

DB\_EXTENDED\_PROPERTY 関数も、事前定義の DB 領域の名前を受け入れます。

### 1.2.3.2 追加の DB 領域の考慮事項

データベースファイルを追加すると、関連情報を個別のファイルにまとめることができます。

#### i 注記

ほとんどのデータベースでは、データベースファイルは 1 つだけで十分です。しかし、大容量データベースを使用していると、多くの場合、追加データベースファイルが必要になります。

データベースを初期化すると、1 つのデータベースファイルが格納されます。この最初のデータベースファイルをメインファイルまたは **systemDB** 領域と呼びます。デフォルトでは、すべてのデータベースオブジェクトとすべてのデータがこのメインファイルに配置されます。

DB 領域は、データ用の領域をさらに作成する追加のデータベースファイルです。1 つのデータベースは 13 個までのファイルに保管されます (メインファイル 1 つと 12 の DB 領域)。各テーブルは、そのインデックスとともに、単一のデータベースファイルに含まれている必要があります。CREATE DBSPACE という SQL 文で、新しいファイルをデータベースに追加できます。

テンポラリテーブルは、temporary DB 領域にのみ作成されます。

ベーステーブルまたはその他のデータベースオブジェクトを作成する DB 領域は複数の方法で指定できます。次のリストで先に示す方法で指定されている場所が、後に示すものよりも優先されます。

1. IN DBSPACE 句 (指定されている場合)
2. default\_dbspace オプション (設定されている場合)
3. system DB 領域

DB 領域名にピリオドが含まれていて、引用符で囲まれていない場合、データベースサーバでエラーが生成されます。

各データベースファイルの最大容量は、 $2^{28}$  (約 2 億 6800 万) データベースページです。たとえば、データベースページサイズが 4 KB のデータベースファイルが作成されると、そのファイルのサイズは 1 テラバイト ( $2^{28} * 4$  KB) まで増やすことができます。しかし実際には、ファイルが作成された物理ファイルシステムで許容される最大ファイルサイズが、最大許容サイズに大きく影響します。

一部の古いファイルシステムではファイルの最大サイズが 2 GB に制限されていますが、Windows が使用している NTFS ファイルシステムのように、多くのファイルシステムでは、データベースファイルを最大サイズまで利用できます。データベースに

あるデータの量が最大ファイルサイズを超える場合は、データを複数のデータベースファイルに分割する必要があります。また、関連オブジェクトをまとめる場合など、サイズ制限以外の理由で複数の DB 領域を作成する場合があります。

sa\_disk\_free システムプロシージャを使用して、DB 領域に使用可能な領域に関する情報を取得できます。

SYSDBSPACE システムビューには、データベースのすべての DB 領域に関する情報が含まれています。

## 既存のデータベースの分割

既存のデータベースオブジェクトをいくつかの DB 領域に分割する場合は、データベースをアンロードし、生成済みのスクリプトファイル (デフォルトでは reload.sql という名前のファイル) をデータベース再構築用に修正します。reload.sql ファイルで、メインファイルに配置しないテーブルごとに、CREATE TABLE 文に IN 句を追加して DB 領域を指定します。

このセクションの内容:

### [DB 領域の権限 \[271 ページ\]](#)

DB 領域では、CREATE 権限のみがサポートされています。

### [DB 領域の作成 \[272 ページ\]](#)

新しいデータベースファイル (DB 領域) を、SQL Central から、または CREATE DBSPACE 文を使用して作成します。

### [データベースファイル領域の事前割り付け \(SQL Central の場合\) \[275 ページ\]](#)

DB 領域やトランザクションログ用のディスク領域を事前に割り付け、大量データのロードのパフォーマンスを向上させます。

### [データベースファイル領域の事前割り付け \(SQL の場合\) \[276 ページ\]](#)

DB 領域やトランザクションログ用のディスク領域を事前に割り付け、大量データのロードのパフォーマンスを向上させます。

### [DB 領域の削除 \(SQL Central の場合\) \[278 ページ\]](#)

SQL Central で DB 領域を削除します。

## 関連情報

[データベースのサイズおよび数に関する物理的な制限 \[891 ページ\]](#)

### 1.2.3.2.1 DB 領域の権限

DB 領域では、CREATE 権限のみがサポートされています。

CREATE 権限によって、ユーザは指定した DB 領域でデータベースオブジェクトを作成できます。GRANT CREATE ON 文を実行して、DB 領域の CREATE 権限を付与できます。

DB 領域の CREATE 権限は次のように動作します。

- 基本となるデータを使用して新しいオブジェクトを作成するユーザには、データを配置する DB 領域の CREATE 権限が必要です。
- GRANT CREATE ON 文が発行されていても、新しいデータベースオブジェクトを作成するユーザ (grantee) には CREATE ANY OBJECT システム権限が必要です。
- 特定の DB 領域に配置でき、CREATE 権限が必要なオブジェクトの現在のリストには、テーブル、インデックス、テキストインデックス、マテリアライズドビューが含まれています。通常のビューやプロシージャなどのオブジェクトには、基本となるデータはなく、CREATE 権限は必要ありません。
- ユーザには CREATE 権限を直接付与することも、権限が付与されているロールのメンバーシップによって権限を継承させることもできます。
- 特定の DB 領域に対する CREATE 権限を PUBLIC に付与することも可能です。この場合、CREATE ANY OBJECT システム権限を持つすべてのユーザが DB 領域にオブジェクトを作成できます。
- DB 領域を新規に作成すると、その DB 領域に対する CREATE 権限が自動的に PUBLIC に付与されます。
- DB 領域を保護する場合などは、権限を取り消すことができます。内部 DB 領域の system と temporary の権限を管理することで、アクセスを制御することもできます。
- ローカルテンポラリテーブルの作成には権限は必要なく、DB 領域の権限はローカルテンポラリテーブルの作成には影響しません。しかし、グローバルテンポラリテーブルの作成には、CREATE ANY OBJECT システム権限と、テンポラリ DB 領域に対する CREATE 権限が必要です。

## 1.2.3.2.2 DB 領域の作成

新しいデータベースファイル (DB 領域) を、SQL Central から、または CREATE DBSPACE 文を使用して作成します。

新しい DB 領域のためのデータベースファイルは、メインファイルが存在するディスクドライブ、または別のディスクドライブに配置できます。

DB 領域の作成には MANAGE ANY DBSPACE システム権限が必要です。

各データベースについて、メイン DB 領域に加えて最大 12 の DB 領域を作成できます。新しく作成された DB 領域は空です。新しいテーブルまたはインデックスを作成するときは、CREATE 文に IN 句を指定して特定の DB 領域に配置することもできれば、事前に default\_dbspace オプションを設定しておくこともできます。IN 句を指定しなかった場合や default\_dbspace オプションの設定を変更しなかった場合、テーブルは system DB 領域に作成されます。

各テーブルは、そのテーブルが作成された DB 領域にすべて格納されます。デフォルトでは、テーブルと同じ DB 領域にインデックスが配置されますが、CREATE 文の一部として IN 句を指定して別の DB 領域に配置することもできます。

このセクションの内容:

### [DB 領域の作成 \(SQL Central の場合\) \[273 ページ\]](#)

SQL Central を使用して DB 領域を作成します。

### [DB 領域の作成 \(SQL の場合\) \[274 ページ\]](#)

CREATE DBSPACE 文を使用して DB 領域を作成します。

## 関連情報

[default\\_dbSPACE オプション \[690 ページ\]](#)

## 1.2.3.2.2.1 DB 領域の作成 (SQL Central の場合)

SQL Central を使用して DB 領域を作成します。

### 前提条件

MANAGE ANY DBSPACE システム権限が必要です。

### コンテキスト

ほとんどのデータベースでは単一のデータベースファイルで十分ですが、追加の DB 領域を使用してデータ領域を増やすことができます。

### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[DB 領域](#)をクリックします。
3. フォルダを右クリックし、**新規** > [DB 領域](#) をクリックします。
4. [DB 領域作成ウィザード](#)の指示に従います。

### 結果

新しい DB 領域が [DB 領域](#) フォルダに表示されます。

### 関連情報

[default\\_dbSPACE オプション \[690 ページ\]](#)

## 1.2.3.2.2.2 DB 領域の作成 (SQL の場合)

CREATE DBSPACE 文を使用して DB 領域を作成します。

### 前提条件

MANAGE ANY DBSPACE システム権限が必要です。

### コンテキスト

ほとんどのデータベースでは単一のデータベースファイルで十分ですが、追加の DB 領域を使用してデータ領域を増やすことができます。

### 手順

CREATE DBSPACE 文を実行します。

### 結果

DB 領域は、特に指定しないかぎり、メインデータベースファイルと同じディレクトリに出現します。

#### 例

次のコマンドは、メインファイルと同じディレクトリにあるファイル `library.db` 内に、MyLibrary という名前の新しい DB 領域を作成します。

```
CREATE DBSPACE MyLibrary
AS 'library.db';
```

次のコマンドは、LibraryBooks テーブルを作成し、それを MyLibrary DB 領域に配置します。

```
CREATE TABLE LibraryBooks (
  title CHAR(100),
  author CHAR(50),
  isbn CHAR(30)
) IN MyLibrary;
```

次のコマンドは、MyLibrary という名前の新しい DB 領域を作成し、この DB 領域をデフォルト DB 領域に設定したうえで、この DB 領域に LibraryBooks テーブルを作成します。

```
CREATE DBSPACE MyLibrary
AS 'e:¥¥dbfiles¥¥library.db';
```

```
SET OPTION default_dbpace = 'MyLibrary';
CREATE TABLE LibraryBooks (
  title CHAR(100),
  author CHAR(50),
  isbn CHAR(30),
);
```

## 関連情報

[default\\_dbpace オプション \[690 ページ\]](#)

### 1.2.3.2.3 データベースファイル領域の事前割り付け (SQL Central の場合)

DB 領域やトランザクションログ用のディスク領域を事前に割り付け、大量データのロードのパフォーマンスを向上させます。

## 前提条件

MANAGE ANY DBSPACE システム権限が必要です。

## コンテキスト

新しいデータベースファイルを作成するときに、データベースファイル領域を事前に割り付けることができます。領域の事前割り付けによって、大量データのロードのパフォーマンスを向上させることができ、データベースファイルの連続性が高まります。

データベースを使用していると、データベースファイルの数が必要に応じて自然に増加します。データベースファイルを頻繁に更新していると、ディスク上のファイルが過度に断片化し、パフォーマンスが低下することがあります。また、小さい領域を多数割り付ける方が、大きい領域を 1 回で割り付けるよりも時間がかかります。変更の頻度が高いデータベースの場合は、DB 領域やトランザクションログのディスク領域を事前に割り付けることができます。

### ➔ ヒント

ディスク領域を事前に割り付けてからディスク断片化解除ユーティリティを実行すると、ディスクドライブのさまざまな箇所にデータベースファイルが断片化されるのを確実に防ぐことができます。データベースファイルの断片化が進むと、パフォーマンスが低下します。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[DB 領域](#)をクリックします。
3. 右ウィンドウ枠で、DB 領域を右クリックし、[領域の事前割り付け](#)をクリックします。
4. DB 領域に追加する領域のサイズを入力します。領域は、ページ、バイト、キロバイト (KB)、メガバイト (MB)、ギガバイト (GB)、またはテラバイト (TB) 単位で追加できます。
5. [OK](#) をクリックします。

## 結果

データベースファイルの領域は事前に割り付けられます。

## 関連情報

[DB 領域の作成 \[272 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

### 1.2.3.2.4 データベースファイル領域の事前割り付け (SQL の場合)

DB 領域やトランザクションログ用のディスク領域を事前に割り付け、大量データのロードのパフォーマンスを向上させます。

## 前提条件

MANAGE ANY DBSPACE システム権限が必要です。

## コンテキスト

新しいデータベースファイルを作成するときに、CREATE DATABASE 文の DATABASE SIZE 句を使用してデータベースファイル領域を事前に割り付けることができます。事前割り付けを行うことで、データベースファイルの連続性が保たれやすくなります。

データベースを使用していると、データベースファイルの数が必要に応じて自然に増加します。データベースファイルを頻繁に更新していると、ディスク上のファイルが過度に断片化し、パフォーマンスが低下することがあります。また、小さい領域を多数割り付ける方が、大きい領域を 1 回で割り付けるよりも時間がかかります。変更の頻度が高いデータベースの場合は、SQL

Central または ALTER DBSPACE 文を使用して、DB 領域やトランザクションログに対し、ディスク領域を事前に割り付けることができます。

#### ➔ ヒント

ディスク領域を事前に割り付けてからディスク断片化解除ユーティリティを実行すると、ディスクドライブのさまざまな箇所にデータベースファイルが断片化されるのを確実に防ぐことができます。データベースファイルの断片化が進むと、パフォーマンスが低下します。

## 手順

1. データベースに接続します。
2. ALTER DBSPACE 文を実行します。

## 結果

データベースファイルの領域は事前に割り付けられます。

#### 例

system DB 領域のサイズを 200 ページ増やします。

```
ALTER DBSPACE system  
ADD 200;
```

system DB 領域のサイズを 400 メガバイト増やします。

```
ALTER DBSPACE system  
ADD 400 MB;
```

## 関連情報

[DB 領域の作成 \[272 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

## 1.2.3.2.5 DB 領域の削除 (SQL Central の場合)

SQL Central で DB 領域を削除します。

### 前提条件

MANAGE ANY DBSPACE システム権限が必要です。

DB 領域を削除する前に、その DB 領域を使用しているすべてのテーブルとインデックスを削除しておく必要があります。

### 手順

1. *SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. **DB 領域**フォルダを開きます。
3. DB 領域を右クリックし、**削除**をクリックします。

### 結果

DB 領域が削除されます。

## 1.2.3.3 DB 領域の削除 (SQL の場合)

SQL で DROP DBSPACE 文を使用して DB 領域を削除します。

### 前提条件

MANAGE ANY DBSPACE システム権限が必要です。

DB 領域を削除する前に、その DB 領域を使用しているすべてのテーブルとインデックスを削除しておく必要があります。

### 手順

1. データベースに接続します。

2. DROP DBSPACE 文を実行します。

## 結果

DB 領域が削除されます。

### 1.2.3.4 トランザクションログ

トランザクションログは、データベースへの変更をすべて保存するファイルです。たとえば、挿入、更新、削除、コミット、ロールバック、データベーススキーマの変更が、すべて記録されます。

トランザクションログは、転送ログまたは REDO ログとも呼ばれます。

トランザクションログは、バックアップとリカバリの重要なコンポーネントであり、Mobile Link を使用したデータ同期や、SQL Remote を使用したデータレプリケーションにも不可欠です。

デフォルトでは、すべてのデータベースがトランザクションログを使用します。トランザクションログの使用はオプションですが、使用しない理由が特になければトランザクションログを使用してください。データベースの実行時にトランザクションログを使用すると、障害からの保護が強化され、パフォーマンスが向上し、データをレプリケートできるようになります。

データベースファイルとトランザクションログを、コンピュータの別々のディスクに保存することをお奨めします。DB 領域とトランザクションログが同じディスク上にある状態でディスク障害が発生すると、すべて失われる可能性があります。しかし、データベースとトランザクションログが別々のディスクに保存されている場合は、ディスク障害が発生した場合でも、すべてまたはほとんどのデータをリカバリできます。これは、フルデータベースや、データベースをリカバリするベースとなるトランザクションログが存在するからです。

データベースファイルまたはトランザクションログファイルのタイムスタンプは、ファイルが増大するか、またはファイルが閉じられた場合のみ更新されます。データベース操作によってトランザクションログファイルが増大するが、データベースファイルが増大しない場合は、トランザクションログファイルのタイムスタンプがデータベースファイルのタイムスタンプよりも最近のものになります。データベースが停止すると、トランザクションログファイルとデータベースファイルのタイムスタンプが更新されます。

#### 警告

データベースファイルとトランザクションログファイルは、データベースサーバと同じ物理コンピュータに保存してください。または SAN や iSCSI 設定でアクセスできるようにしてください。リモートネットワークディレクトリにデータベースファイルやトランザクションログファイルを配置すると、パフォーマンスが低下したり、データが破壊されたり、サーバが不安定になったりする可能性があります。

## 変更がディスクに書き込まれるとき

データベースファイルと同様に、トランザクションログはページ、つまり固定サイズのメモリ領域に保持されます。トランザクションログに変更が記録される時、その内容はメモリ内のページに書き込まれます。変更は、次の状況のいずれかが発生した段階で、強制的にディスクに書き込まれます。

- ページが満杯になったとき
- COMMIT が実行されたとき

完了したトランザクションは確実にディスクに格納され、また操作のたびにディスクに書き込む必要がないため、パフォーマンスも向上します。

cooperative\_commits オプションと delayed\_commits オプションを使用すると、トランザクションログの動作を細かくチューニングできます。

このセクションの内容:

#### [トランザクションログの検証 \[281 ページ\]](#)

データベースサーバは、トランザクションログミラーを使用するデータベースの起動時に、トランザクションログの検証を行います。

#### [トランザクションログの場所の変更 \(SQL Central の場合\) \[281 ページ\]](#)

SQL Central 内で、トランザクションログの場所を変更します。

#### [トランザクションログの場所の変更 \(コマンドラインの場合\) \[283 ページ\]](#)

dblog ユーティリティを使用して、トランザクションログの場所を変更します。

#### [トランザクションログの名前変更またはトランケート \(SQL Central の場合\) \[284 ページ\]](#)

SQL Central でトランザクションログの名前変更またはトランケートを行います。

#### [トランザクションログの名前変更またはトランケート \(SQL の場合\) \[285 ページ\]](#)

SQL によってトランザクションログの名前変更またはトランケートを行います。

#### [トランザクションログの名前変更またはトランケート \(コマンドラインの場合\) \[286 ページ\]](#)

コマンドラインから、トランザクションログの名前変更またはトランケートを行います。

#### [トランザクションログサイズの考慮事項 \[287 ページ\]](#)

トランザクションログのサイズは、リカバリの所要時間に影響します。

#### [トランザクションログミラー \[288 ページ\]](#)

トランザクションログミラーはトランザクションログの完全なコピーであり、トランザクションログと同時に管理されます。

#### [未処理のトランザクションがある接続の特定 \(SQL の場合\) \[291 ページ\]](#)

SQL で、未処理のトランザクションがあるユーザを特定します。

#### [チェックポイント \[292 ページ\]](#)

チェックポイントとは、ダーティページがディスクに書き込まれる時点のことで、ディスク上のデータベースが既知の一貫性のある状態であることを示します。

#### [オフライントランザクションログ \[296 ページ\]](#)

バックアップ操作では、トランザクションログのバックアップに加え、オンライントランザクションログのファイル名を YYYYMMDDxx.llog という形式に変更できます。これをオフライントランザクションログと呼びます。

## 関連情報

### [メディア障害からのリカバリ \[937 ページ\]](#)

[Running a SQL Anywhere Database File that is Stored Remotely from the Server Machine \(サーバマシンからリモートで格納された SQL Anywhere データベースファイルの実行\)](#) 

### [-m データベースサーバオプション \[447 ページ\]](#)

[-m データベースオプション \[528 ページ\]](#)

[delete\\_old\\_logs オプション \[Mobile Link\]\[SQL Remote\] \[695 ページ\]](#)

[cooperative\\_commits オプション \[廃止予定\] \[681 ページ\]](#)

[delayed\\_commits オプション \[694 ページ\]](#)

### 1.2.3.4.1 トランザクションログの検証

データベースサーバは、トランザクションログミラーを使用するデータベースの起動時に、トランザクションログの検証を行います。

一連の検査と自動リカバリ操作を実行してトランザクションログとそのミラーが壊れていないかを確認します。壊れている場合は、いくつかの問題を修正します。

起動時には、トランザクションログとミラーを比較して、2つのファイルが同一であるかを調べます。同一であれば、データベースは通常どおりに起動します。データベースの開始は、このログとミラーの比較のために時間がかかります。

システム障害のためにデータベースが停止した場合、操作の一部がトランザクションログには書き込まれても、ミラーには書き込まれていない可能性があります。トランザクションログとミラーのどちらか短い方のファイルを最後までチェックして2つのファイルが同じであることをサーバが確認すると、長い方のファイルの残りの部分が短い方のファイルにコピーされます。これによって、ログとミラーは同一になります。このリカバリ作業の後に、サーバは正常に起動します。

トランザクションログとトランザクションログミラーの内容が異なる場合は、どちらかのファイルが破損しています。この場合、データベースは起動しないで、エラーメッセージが表示され、トランザクションログかミラーのどちらかが無効であることを知らせます。

トランザクションログの検証には、対象となるトランザクションログがオンラインかオフラインかに関係なく、ログ変換ユーティリティ (dbtran) を使用することもできます。ログ変換ユーティリティによるトランザクションログファイルの読み込みが正常に実行される場合、そのログは有効です。

## 関連情報

[ログ変換ユーティリティ \(dbtran\) \[1133 ページ\]](#)

### 1.2.3.4.2 トランザクションログの場所の変更 (SQL Central の場合)

SQL Central 内で、トランザクションログの場所を変更します。

## 前提条件

トランザクションログの場所を変更する場合、データベースは停止している必要があります。

## 警告

データベースファイルとトランザクションログファイルは、データベースサーバと同じ物理コンピュータに保存してください。または SAN や iSCSI 設定でアクセスできるようにしてください。リモートネットワークディレクトリにデータベースファイルやトランザクションログファイルを配置すると、パフォーマンスが低下したり、データが破壊されたり、データベースサーバが不安定になったりする可能性があります。

最適な結果を得るために、トランザクションログは、データベースファイルとは別のディスクに保存してください。

## コンテキスト

トランザクションログを同じコンピュータ上の別のドライブに移動しなければならない場合に、このタスクを実行します。

## 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2.  ツール  *SQL Anywhere17*  ログファイル設定の変更  をクリックします。
3. ログファイル設定の変更ウィザードの指示に従います。

## 結果

トランザクションログの場所が変更されます。

## 関連情報

[Running a SQL Anywhere Database File that is Stored Remotely from the Server Machine \(サーバマシンからリモートで格納された SQL Anywhere データベースファイルの実行\)](#) 

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

### 1.2.3.4.3 トランザクションログの場所の変更 (コマンドラインの場合)

dblog ユーティリティを使用して、トランザクションログの場所を変更します。

#### 前提条件

トランザクションログの場所を変更する場合、データベースは停止している必要があります。

#### コンテキスト

トランザクションログを同じコンピュータ上の別のドライブに移動しなければならない場合に、このタスクを実行します。

#### 手順

次のコマンドを実行します。

```
dblog -t new-transaction-log-file database-file
```

#### 結果

トランザクションログの場所が変更されます。

#### 関連情報

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

## 1.2.3.4.4 トランザクションログの名前変更またはトランケート (SQL Central の場合)

SQL Central でトランザクションログの名前変更またはトランケートを行います。

### 前提条件

BACKUP DATABASE 権限が必要です。

### コンテキスト

データベースの同期やレプリケーションを行っている場合、トランザクションログの名前変更を行い、データベースの新しいトランザクションログを作成します。データベースのレプリケーションを行っていないで、コンピュータのディスク容量が限られている場合には、トランザクションログのトランケートを行います。

データベースの同期やレプリケーションを行っている場合は、確実に不要になるまで、古いトランザクションログのコピーを保存しておくことをお奨めします。トランザクションログの名前変更を行わず、データベースのバックアップ時に既存のトランザクションを使い続けることもできます。

バックアップ中にトランザクションログのトランケートを行ってオンライントランザクションログの内容を削除する場合、データベースファイル上のメディア障害からデータベースをリカバリするためには、前回のフルバックアップ以降に作成されたすべてのバックアップコピーを使用する必要があります。

### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. データベースを右クリックし、**イメージバックアップの作成**をクリックします。
3. **イメージバックアップ作成ウィザード**の指示に従います。

オプション	アクション
トランザクションログの名前を変更する	トランザクションログの処理方法を指定してくださいというリストで、トランザクションログの名前を変更をクリックします。
トランザクションログをトランケートする	トランザクションログの処理方法を指定してくださいというリストで、トランザクションログをトランケートをクリックします。

### 結果

バックアップが作成され、トランザクションログは名前変更またはトランケートされます。

## 関連情報

[データベースのリカバリ \[915 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

### 1.2.3.4.5 トランザクションログの名前変更またはトランケート (SQL の場合)

SQL によってトランザクションログの名前変更またはトランケートを行います。

#### 前提条件

BACKUP DATABASE システム権限が必要です。

#### コンテキスト

データベースの同期やレプリケーションを行っている場合、トランザクションログの名前変更を行い、データベースの新しいトランザクションログを作成します。データベースのレプリケーションを行っていない、コンピュータのディスク容量が限られている場合には、トランザクションログのトランケートを行います。

データベースの同期やレプリケーションを行っている場合は、確実に不要になるまで、古いトランザクションログのコピーを保存しておくことをお勧めします。トランザクションログの名前変更を行わず、データベースのバックアップ時に既存のトランザクションを使い続けることもできます。

バックアップ中にトランザクションログのトランケートを行ってオンライントランザクションログの内容を削除する場合、データベースファイル上のメディア障害からデータベースをリカバリするためには、前回のフルバックアップ以降に作成されたすべてのバックアップコピーを使用する必要があります。

#### 手順

次の句を使用して BACKUP DATABASE 文を実行します。

オプション	アクション
トランザクションログの名前を変更する	<pre>BACKUP DATABASE DIRECTORY backup-directory [ TRANSACTION LOG ONLY ] TRANSACTION LOG RENAME;</pre>

オプション	アクション
トランザクションログをトランケートする	<pre>BACKUP DATABASE   DIRECTORY backup-directory   [ TRANSACTION LOG ONLY ]   TRANSACTION LOG TRUNCATE;</pre>

インクリメンタルバックアップを作成する場合だけ TRANSACTION LOG ONLY 句を使用します。

トランザクションログとデータベースファイルのバックアップコピーは、backup-directory にあります。パスを指定する場合、クライアントアプリケーションではなく、データベースサーバの作業ディレクトリからの相対パスを指定します。

## 結果

バックアップが作成され、トランザクションログは名前変更またはトランケートされます。

## 関連情報

[データベースのリカバリ \[915 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

### 1.2.3.4.6 トランザクションログの名前変更またはトランケート (コマンドラインの場合)

コマンドラインから、トランザクションログの名前変更またはトランケートを行います。

## 前提条件

BACKUP DATABASE システム権限が必要です。

## コンテキスト

データベースの同期やレプリケーションを行っている場合、トランザクションログの名前変更を行い、データベースの新しいトランザクションログを作成します。データベースのレプリケーションを行っていない、コンピュータのディスク容量が限られている場合には、トランザクションログのトランケートを行います。

データベースの同期やレプリケーションを行っている場合は、確実に不要になるまで、古いトランザクションログのコピーを保存しておくことをお奨めします。トランザクションログの名前変更を行わず、データベースのバックアップ時に既存のトランザクションを使い続けることもできます。

バックアップ中にトランザクションログのトランケートを行ってオンライントランザクションログの内容を削除する場合、データベースファイル上のメディア障害からデータベースをリカバリするためには、前回のフルバックアップ以降に作成されたすべてのバックアップコピーを使用する必要があります。

## 手順

次のコマンドを実行します。

オプション	アクション
トランザクションログの名前を変更する	<code>dbbackup -c "connection-string" -r [ -t ] backup-directory</code>
トランザクションログをトランケートする	<code>dbbackup -c "connection-string" -x [ -t ] backup-directory</code>

インクリメンタルバックアップを作成する場合だけ、-t オプションを指定します。

トランザクションログとデータベースファイルのバックアップコピーは、`backup-directory` にあります。パスを指定する場合には、コマンドを実行するディレクトリからの相対パスを指定します。

## 結果

バックアップが作成され、トランザクションログは名前変更またはトランケートされます。

## 関連情報

[データベースのリカバリ \[915 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

### 1.2.3.4.7 トランザクションログサイズの考慮事項

トランザクションログのサイズは、リカバリの所要時間に影響します。

すべてのテーブルに簡単なプライマリキーを設定することによって、トランザクションログファイルの拡大を制御できます。プライマリキーまたは NULL 入力不可のユニークインデックスがないテーブルで更新または削除を実行すると、対象ローの内容がすべてトランザクションログに保存されます。プライマリキーが定義されている場合は、データベースサーバはそのカラム値を保存するだけでローをユニークに識別できます。テーブルのカラム数が多い場合、またはテーブルのカラムに長いデータを

含む場合は、プライマリキーが定義されていないと、トランザクションログのページがすぐに満杯になります。このように、データの余分な書き込みによって、ディスク領域を多く必要とするだけでなく、パフォーマンスが低下します。

プライマリキーがない場合、サーバはテーブル上で UNIQUE NOT NULL インデックス (または UNIQUE 制約) を探します。NULL 値を許容する UNIQUE インデックスでは十分に識別できないためです。

## 関連情報

[-m データベースサーバオプション \[447 ページ\]](#)

[-m データベースオプション \[528 ページ\]](#)

[delete\\_old\\_logs オプション \[Mobile Link\]\[SQL Remote\] \[695 ページ\]](#)

## 1.2.3.4.8 トランザクションログミラー

トランザクションログミラーはトランザクションログの完全なコピーであり、トランザクションログと同時に管理されます。

データベースにトランザクションログミラーがある場合、データベースに対する変更は、トランザクションログとトランザクションログミラーの両方に書き込まれます。デフォルトでは、データベースは、トランザクションログミラーを持ちません。

トランザクションログミラーは、重要なデータを二重に保護します。トランザクションログでメディア障害が発生した場合に、完全なデータリカバリが可能になります。また、トランザクションログミラーがあると、データベースの開始時に、データベースサーバによるトランザクションログの自動検証が可能になります。

大容量または重要なアプリケーションを実行する場合は、トランザクションログミラーを使用することをおすすめします。たとえば、SQL Remote 設定における統合データベースでは、レプリケーションはトランザクションログに依存します。トランザクションログが損傷した場合、データのレプリケーションは失敗します。

トランザクションログミラーを使用している場合、ログのいずれかに書き込もうとしてエラーが発生すると (ディスクが満杯の場合など)、データベースサーバが停止します。トランザクションログミラーの目的は、いずれかのログデバイスでメディア障害が発生したときに完全にリカバリできるようにすることです。1 つのトランザクションログを使用してサーバを実行し続けると、この目的は達成できません。

データベースサーバの起動時に `-fc` オプションを指定して、データベースサーバでファイルシステムがいっぱいになった場合のコールバック関数を実装できます。

## トランザクションログミラーの保存先

トランザクションログミラーを使用すると、各データベースログへの書き込みを 2 回ずつ行う必要があるため、パフォーマンスが低下します。低下の程度は、データベース内のデータ転送の形態と量、データベースとログの物理的な設定の方法によって異なります。

トランザクションログミラーは、トランザクションログとは別のデバイスに保管してください。これによってパフォーマンスが向上し、また一方のデバイスが故障しても、ログのもう一方のコピーにリカバリのためのデータが残ります。

## トランザクションログミラーに代わる方法

トランザクションログミラーに代わる方法として、次の構成を使用できます。

- データベースのミラーリング。
- ハードウェアミラーリングを行うディスクコントローラ。通常、ハードウェアミラーリングはオペレーティングシステムレベルのソフトウェアミラーリングよりもコストがかかりますが、パフォーマンスは高くなります。
- Microsoft Windows に用意されているオペレーティングシステムレベルのソフトウェアミラーリング。

ライブバックアップを使用すると、トランザクションログミラーに似た方法でデータを保護することができます。

このセクションの内容:

[既存のデータベースでのトランザクションログミラーの開始 \(SQL Central の場合\) \[289 ページ\]](#)

既存のデータベースに対するトランザクションログミラーを開始します (SQL Central の場合)。

[既存のデータベースに対するトランザクションログミラーの開始 \(コマンドラインの場合\) \[290 ページ\]](#)

既存のデータベースに対するトランザクションログミラーを開始します (コマンドラインの場合)。

## 関連情報

[データベースミラーリング \[1660 ページ\]](#)

[ライブバックアップ \(継続バックアップ\) \[899 ページ\]](#)

[-fc データベースサーバオプション \[409 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

### 1.2.3.4.8.1 既存のデータベースでのトランザクションログミラーの開始 (SQL Central の場合)

既存のデータベースに対するトランザクションログミラーを開始します (SQL Central の場合)。

## 前提条件

トランザクションログがあるディレクトリへのアクセス権が必要です。

## 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. **▶ ツール ▶ SQL Anywhere17 ▶ ログファイル設定の変更 ▶**をクリックします。
3. **ログファイル設定の変更ウィザード**の指示に従います。

## 結果

トランザクションログミラーはデータベースに対して開始されます。

## 関連情報

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

### 1.2.3.4.8.2 既存のデータベースに対するトランザクションログミラーの開始 (コマンドラインの場合)

既存のデータベースに対するトランザクションログミラーを開始します (コマンドラインの場合)。

## 前提条件

トランザクションログがあるディレクトリへのアクセス権が必要です。

## コンテキスト

既存のデータベースのトランザクションログミラーの管理は、データベースが実行中でなければいつでも実行できます。また、dblog ユーティリティを使用して、データベースによるトランザクションログミラーの使用を停止することもできます。

## 手順

1. データベースが実行中でないことを確認します。

2. 次のコマンドを実行します。

```
dblog -m mirror-file database-file
```

## 結果

トランザクションログミラーはデータベースに対して開始されます。

## 関連情報

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

### 1.2.3.4.9 未処理のトランザクションがある接続の特定 (SQL の場合)

SQL で、未処理のトランザクションがあるユーザを特定します。

## 前提条件

SERVER OPERATOR、MONITOR、または DROP CONNECTION システム権限が必要です。

## コンテキスト

トランザクションログの名前変更または削除を行うバックアップを実行すると、完了していないトランザクションは、新しいトランザクションログに持ち越されます。

## 手順

1. Interactive SQL からデータベースに接続します。
2. sa\_conn\_info システムプロシージャを実行します。

```
CALL sa_conn_info;
```

## 結果

結果ウィンドウ枠の *UncommitOps* カラムで、接続に未処理のトランザクションがないか調べます。

## 次のステップ

必要に応じて、DROP CONNECTION 文を使用してユーザを切断できます。

## 関連情報

[SQL Anywhere コックピット \[1349 ページ\]](#)

### 1.2.3.4.10 チェックポイント

チェックポイントとは、ダーティページがディスクに書き込まれる時点のことで、ディスク上のデータベースが既知の一貫性のある状態であることを示します。

チェックポイントの後で、チェックポイントログの内容が削除されます。空のチェックポイントログページは、現在のセッション中はチェックポイントログに存在し、新しいチェックポイントログデータに再利用できます。チェックポイントログのサイズが大きくなると、データベースファイルも大きくなります。

チェックポイント時には、データベースの全データがディスクのデータベースファイルに格納されます。データベースファイルの情報は、トランザクションログの情報と一致します。リカバリ時には、データベースはまず最新のチェックポイントでリカバリされ、次にチェックポイント以降の変更内容が適用されます。

各セッションの最後では、チェックポイントログの使用履歴がデータベースに格納され、次のセッションのチェックポイントログに適切なサイズを決定するために使用されます。

データベースサーバは、チェックポイントを開始し、チェックポイントの発生中にその他の操作を実行できます。ただし、チェックポイントがすでに進行中だった場合、新しいチェックポイントを開始する文は、現在のチェックポイントの完了を待ってからでないと実行できません。

次の文は、暗黙的または明示的なチェックポイントを開始します。

- ALTER INDEX REBUILD 文
- ALTER TABLE 文 (NULL 入力可のカラムまたは参照整合性制約を追加せず、テーブルのページを物理的に変更しない場合)
- BACKUP DATABASE 文
- CHECKPOINT 文
- COMMIT 文 (データベースがトランザクションログを作成していないとき)

- CREATE DBSPACE 文
- CREATE INDEX 文 (関数にインデックスを作成するために使用されるとき (暗黙の計算カラム))
- DROP DBSPACE 文
- DROP TABLE 文と DROP MATERIALIZED VIEW 文 (テーブルまたはビューに少なくとも 1 つのローが含まれている場合のみ)
- LOAD TABLE 文
- REFRESH MATERIALIZED VIEW 文
- REORGANIZE TABLE 文

このセクションの内容:

[データベースサーバがチェックポイントのタイミングを決定する方法 \[293 ページ\]](#)

データベースサーバがチェックポイントを実行する場合、いくつかの条件があります。

[チェックポイントログ \[294 ページ\]](#)

チェックポイントログは、データベースファイルの末尾にあり、system DB 領域に保存されます。

## 関連情報

[データベースのバックアップ \[895 ページ\]](#)

### 1.2.3.4.10.1 データベースサーバがチェックポイントのタイミングを決定する方法

データベースサーバがチェックポイントを実行する場合、いくつかの条件があります。

データベースが正常に停止された場合、データベースファイルにはデータベースにある全データの現行の完全なコピーが保持されています。しかし、データベースの稼働中は、データベースファイルは通常現行のものまたは完全なものではありません。データベースファイルが全データの完全な現行のコピーを保持していることが保証されるのは、チェックポイントの完了直後だけです。チェックポイントの後には、データベースキャッシュの全内容がディスク上にあります。

データベースサーバは、次の場合にデータベースに対するチェックポイントを実行します。

- データベースの停止操作の一環として
- 最後のチェックポイントからの経過時間が、-gc サーバオプションの設定を超えたとき
- リカバリ予想時間が、-gr サーバオプションの設定を超えたとき
- データベースサーバのアイドル状態が、ダーティページをすべて書き込めるほどに長く続いているとき
- 特定の DDL 文 (ALTER TABLE、DROP TABLE、DROP INDEX、LOAD TABLE、BACKUP など) が実行されたとき
- 接続により CHECKPOINT 文が発行されたとき
- データベースサーバがトランザクションログなしで稼働している場合に、トランザクションがコミットされたとき

最後のチェックポイント以降の時間と作業量の増加に伴い、ダーティページをディスクに書き込む優先度も増します。この優先度は、以下の要因によって決まります。

チェックポイントの緊急度

最後のチェックポイント以降の経過時間を、データベースのチェックポイント時間の設定に対するパーセンテージで表したものです。チェックポイント間の最大時間は、-gc サーバオプションまたは checkpoint\_time データベースオプションを使って分単位で設定できます。-gc を指定した場合、データベース内の checkpoint\_time オプションの設定は無視されません。

#### リカバリの緊急度

データベースの障害が直ちに発生した場合のリカバリに必要な時間の推計です。システム障害が発生したときにリカバリにかかる最大時間は、-gr サーバオプションまたは recovery\_time データベースオプションを使って分単位で設定できます。-gr を指定した場合、データベース内の recovery\_time オプションの設定は無視されます。

チェックポイントとリカバリの緊急度の値は、データベースサーバがダーティページの書き込みを行うだけのアイドル時間を持っていない場合のみ重要です。チェックポイントの間隔の下限値は、recovery\_time と checkpoint\_time オプションの組み合わせによって決まります。recovery\_time オプションの設定に従うとチェックポイントの間隔が短すぎる場合、その設定は尊重されません。

チェックポイントの頻度が高いとシステム障害からのリカバリは速くなります。しかし、データベースエンジンがダーティページを書き出す作業が増えます。

データベースの他のアクティビティがあるためにダーティページ数が 0 になり、チェックポイントの緊急度が 33% 以上である場合、チェックポイントは自動的に発生します。

チェックポイントの緊急度とリカバリの緊急度の値は、チェックポイントが発生するまで増加を続け、チェックポイントが発生すると 0 に戻ります。

## 関連情報

[チェックポイントログ \[294 ページ\]](#)

[-gc データベースサーバオプション \[413 ページ\]](#)

[checkpoint\\_time オプション \[667 ページ\]](#)

[-gr データベースサーバオプション \[428 ページ\]](#)

[recovery\\_time オプション \[768 ページ\]](#)

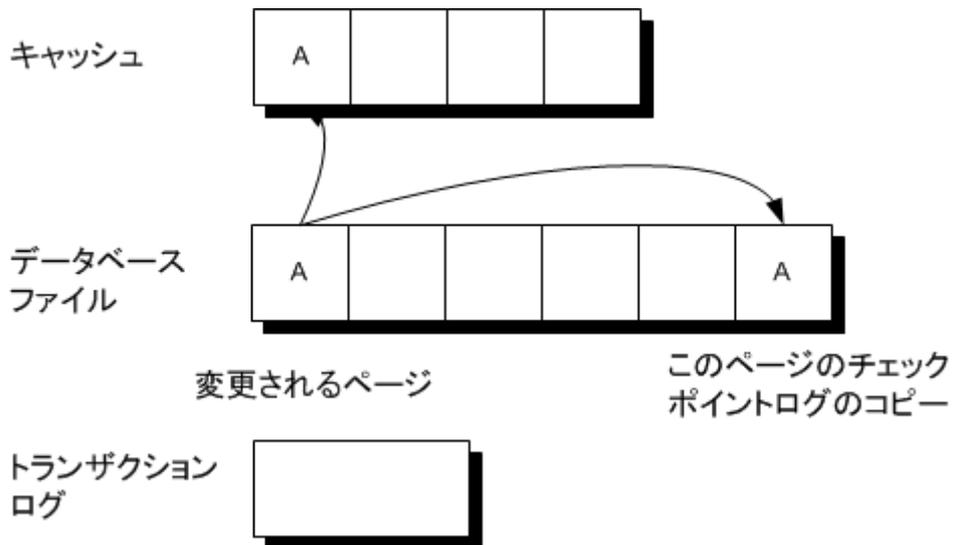
### 1.2.3.4.10.2 チェックポイントログ

チェックポイントログは、データベースファイルの末尾にあり、system DB 領域に保存されます。

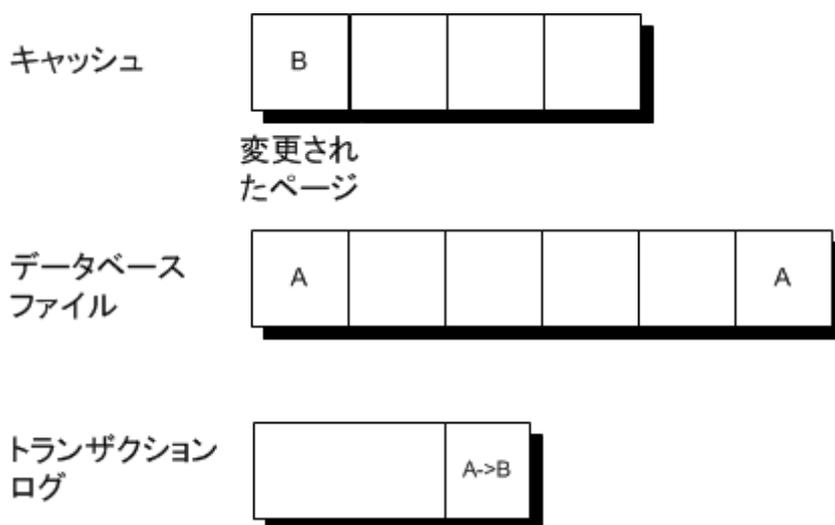
データベースファイルは、複数のページ、つまり、ハードディスクの一定サイズの領域で構成されています。セッション中とセッションの最後には必要に応じてチェックポイントログにページが追加され、チェックポイントの使用履歴がデータベースに格納されます。この履歴は、以降のセッションのチェックポイントログに適切なサイズを決定するために使用されます。

データベースサーバでは、ページが更新される (ダーティになる) 前に次の処理が実行されます。

- ページがメモリに読み込まれ、データベースキャッシュ内に格納されます。
- 元のページのコピーが作成されます。これらのコピーされたページを「チェックポイントログ」と呼びます。



ページに加えた変更は、キャッシュ内のコピーに適用されます。パフォーマンス上の理由から、ページはすぐにはディスクのデータベースファイルに書き込まれません。



キャッシュが満杯になると、変更されたページはディスクに書き込まれます。チェックポイントログのコピーは変更されません。

キャッシュ



データベース  
ファイル



トランザクション  
ログ



## 関連情報

[データベースのバックアップ \[895 ページ\]](#)

[データベースサーバがチェックポイントのタイミングを決定する方法 \[293 ページ\]](#)

### 1.2.3.4.11 オフライントランザクションログ

バックアップ操作では、トランザクションログのバックアップに加え、オンライントランザクションログのファイル名を `YYMMDDxx.log` という形式に変更できます。これをオフライントランザクションログと呼びます。

このファイルはデータベースサーバでは使用されませんが、Message Agent では利用できます。新しいオンライントランザクションログの最初の名前には、古いオンライントランザクションログの名前が付けられます。

`YYMMDDxx.log` というファイル名は、順序付けではなく、ファイルを区別するために使用されます。たとえば、最初のバックアップが 2000 年 12 月 10 日である場合、ログファイル名は `001210AA.log` になります。最初の 2 桁は年、次の 2 桁は月、その次の 2 桁は日付を示し、最後の 2 文字によって、同じ日に実行された複数のバックアップを識別します。

Message Agent は、オフラインコピーを使用して古いトランザクションを必要に応じて提供できます。 `delete_old_logs` データベースオプションを On に設定すると、Message Agent が必要としなくなった古いオフラインファイルは削除されるので、ディスク領域を節約できます。

### 1.2.3.5 ユーティリティデータベース (utility\_db)

ユーティリティデータベースは、物理的な実体を持たない幻のデータベースです。

ユーティリティデータベースにはデータベースファイルがないため、データを入れることができません。この機能では次のことができます。

- 先に既存の物理データベースに接続しなくても、CREATE DATABASE などのデータベースファイル管理文を実行できます。
- ユーティリティデータベースを使用して、接続プロパティやサーバプロパティの値を取得できます。
- データベースへの接続が不可能な場合に、実行中のデータベースサーバに接続できます。たとえば、ミラーリングシステムでは、ユーティリティデータベースに接続してミラーリングサーバを停止したり、データベースを強制的にプライマリサーバにしたりできます。

ユーティリティデータベースの名前は utility\_db です。この名前を持つデータベースを作成しようとする、操作は失敗します。

-su データベースサーバオプションを使用すると、ユーティリティデータベースパスワードを設定したり (ユーザ ID の設定も任意で可能)、ユーティリティデータベースへの接続を無効にしたりできます。ユーティリティデータベースのデフォルトのユーザ ID は、DBA です。ユーティリティデータベースのパスワードは 6 文字以上にする必要があります。また、大文字と小文字は区別されます。

サーバへの接続時に utility\_db をデータベース名として指定すると、データベースサーバ上のユーティリティデータベースを起動できます。

ユーティリティデータベースに接続しているときに実行できるのは、次の文だけです。

- ALTER DATABASE dbfile ALTER TRANSACTION LOG 文。
- ALTER DATABASE database-name FORCE START 文。
- CREATE DATABASE 文。
- CREATE DECRYPTED DATABASE 文。
- CREATE DECRYPTED FILE 文。
- CREATE ENCRYPTED DATABASE 文。
- CREATE ENCRYPTED FILE 文。
- DROP DATABASE 文。
- CREATE USER DBA IDENTIFIED BY new-password 文。
- RESTORE DATABASE 文。
- REVOKE CONNECT FROM DBA 文。
- SET TEMPORARY OPTION progress\_messages = [OFF | RAW | FORMATTED]。
- FROM 句または WHERE 句なしの SELECT 文。
- START DATABASE 文。
- STOP DATABASE 文。
- STOP SERVER 文。

-gu データベースサーバオプションを使用すると、データベースファイル管理文の実行に必要な権限を設定できます。-gu utility\_db と指定すると、ユーティリティデータベースだけがこれらの文を実行できるようになります。

### 例

ユーティリティデータベースに接続してから次の文を実行すると、new.db という名前のデータベースがディレクトリ c:¥temp に作成されます。

```
CREATE DATABASE 'c:¥temp¥new.db' DBA USER 'DBA' DBA PASSWORD 'passwd';
```

ユーティリティデータベースで次の文を実行すると、デフォルトの照合順が返され、作成するデータベースに使用できます。

```
SELECT PROPERTY( 'DefaultCollation' );
```

このセクションの内容:

[ユーティリティデータベースへの接続 \(接続ウィンドウの場合\) \[298 ページ\]](#)

ユーティリティデータベースに接続し、データベースファイル管理文を実行したり、接続プロパティやサーバプロパティのクエリを実行したり、データベースへの接続が不可能な場合に実行中のデータベースサーバに接続したりします。

## 関連情報

[ミラーリングシステムのデータベースサーバの停止 \(dbstop ユーティリティ\) \[1684 ページ\]](#)

[トラブルシューティング: プライマリサーバが起動できない \[1694 ページ\]](#)

[.ini ファイルの内容を非表示にする \[566 ページ\]](#)

[-gu データベースサーバオプション \[435 ページ\]](#)

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.2.3.5.1 ユーティリティデータベースへの接続 (接続ウィンドウの場合)

ユーティリティデータベースに接続し、データベースファイル管理文を実行したり、接続プロパティやサーバプロパティのクエリを実行したり、データベースへの接続が不可能な場合に実行中のデータベースサーバに接続したりします。

## 前提条件

-su データベースサーバオプションは、ユーティリティデータベースのパスワードを指定します。また、必要に応じてユーザ ID も指定します。-su オプションを使用して起動したサーバ上のユーティリティデータベースに接続するには、-su オプションで設定したユーザ ID とパスワードが必要になります。

ネットワークサーバに対して -su データベースサーバオプションを指定しなかった場合は、ユーティリティデータベースに接続できません。

パーソナルデータベースサーバに対しては -su データベースサーバオプションを指定しなくても、ユーティリティデータベースに接続でき、パスワードは必要ありません。

## コンテキスト

実行中のデータベースサーバがあり、そのユーティリティデータベースに接続したいとします。

データベースサーバがネットワークサーバ (dbsrv17) の場合は、-su オプションを使用して起動されており、ユーザ ID (任意) とパスワードがあります。

```
dbsrv17 -n server-name -su user-ID,password
```

データベースサーバがパーソナルデータベースサーバ (dbeng17) の場合は、-su オプションを使用して起動されていてユーティリティデータベースにユーザ ID (任意) とパスワードがある場合もありますし、-su オプションを使用せずに起動されていてユーティリティデータベースにユーザ ID とパスワードがない場合もあります。

```
dbeng17 -n server-name
```

## 手順

1. 接続ウィンドウを開きます。
2. ユーザ ID フィールドで、次のいずれかのオプションを選択します。

オプション	アクション
-su データベースサーバオプションが指定され、ユーザ ID が含まれている場合	-su データベースサーバオプションで指定したパスワードを入力します。
-su データベースサーバオプションが指定されていないか、または -su にパスワードだけが指定された場合	DBA と入力します。

3. パスワードフィールドで、次のいずれかのオプションを選択します。

オプション	アクション
パーソナルサーバ (dbeng17) で、-su データベースサーバオプションが指定されていない場合	6 文字以上の任意のパスワードを入力します。
ネットワークサーバ (dbsrv17) の場合	-su オプションで指定したパスワードを入力します。

4. アクションドロップダウンリストから、このコンピュータで稼働しているデータベースに接続をクリックします。
5. データベース名フィールドに、utility\_db と入力します。
6. サーバ名フィールドに server-name と入力します。
7. 接続をクリックします。

## 結果

Interactive SQL から、パーソナルサーバ上またはネットワークサーバ上のユーティリティデータベースに接続します。

ユーティリティデータベースに接続したら、REVOKE CONNECT FROM user-ID を実行して、以降のユーティリティデータベースへの接続を無効にします。以降、REVOKE CONNECT の実行前に存在していた接続を使用するか、データベースサーバを再起動しないかぎり、ユーティリティデータベースへの接続はできなくなります。

## 関連情報

[データベース接続 \[5 ページ\]](#)

[ミラーリングシステムのデータベースサーバの停止 \(dbstop ユーティリティ\) \[1684 ページ\]](#)

[トラブルシューティング: プライマリサーバが起動できない \[1694 ページ\]](#)

[-su データベースサーバオプション \[480 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

### 1.2.3.6 データベースの削除 (SQL Central の場合)

SQL Central でデータベースを削除します。

#### 前提条件

データベース消去ウィザードを使用する場合、削除するデータベースは実行中にしないでください。

#### コンテキスト

データベースを削除すると、データベースへの変更を記録するトランザクションログを含むすべてのテーブルとデータが、ディスクから削除されます。

データベースファイルはすべて、誤ってファイルを変更したり削除したりするのを防ぐために読み込み専用となっています。

#### 手順

1. SQL Central で、**ツール** > **SQL Anywhere17** > **データベースの消去** をクリックします。
2. ウィザードの指示に従います。

#### 結果

データベースが削除されます。

## 関連情報

[ユーティリティデータベースへの接続 \(接続ウィンドウの場合\) \[298 ページ\]](#)

[-gu データベースサーバオプション \[435 ページ\]](#)

[消去ユーティリティ \(dberase\) \[1103 ページ\]](#)

### 1.2.3.7 データベースの削除 (dberase ユーティリティの場合)

dberase ユーティリティを使用して、データベースを削除します。

#### 前提条件

dberase ユーティリティを使用する場合、削除するデータベースは実行中にしてしないでください。

#### コンテキスト

データベースを削除すると、データベースへの変更を記録するトランザクションログを含むすべてのテーブルとデータが、ディスクから削除されます。

データベースファイルはすべて、誤ってファイルを変更したり削除したりするのを防ぐために読み込み専用となっています。

dberase ユーティリティでは DB 領域は削除されません。DB 領域を削除するには、DROP DBSPACE 文を使用します。

#### 手順

dberase ユーティリティを実行します。

#### 結果

データベースが消去されます。

#### 例

次のコマンドは、temp データベースを削除します。

```
dberase c:¥temp¥temp.db
```

## 関連情報

[ユーティリティデータベースへの接続 \(接続ウィンドウの場合\) \[298 ページ\]](#)

[-gu データベースサーバオプション \[435 ページ\]](#)

[消去ユーティリティ \(dberase\) \[1103 ページ\]](#)

### 1.2.4 カラムデータ型の考慮事項

カラムにはいくつかのデータ型を使用できます。

- バイナリデータ型
- 文字データ型
- 日付/時刻データ型
- decimal データ型
- ドメイン (ユーザ定義データ型)
- 浮動小数点データ型
- integer データ型
- 空間データ型

CHAR、VARCHAR、LONG VARCHAR、NCHAR、BINARY、VARBINARY などの文字列やバイナリ文字列のデータ型を使用して、イメージ、ワープロ文書、音声ファイルなどのラージオブジェクトを格納できます。

このセクションの内容:

[NULL と NOT NULL カラム \[302 ページ\]](#)

カラム値がローに対して必須である場合、そのカラムを NOT NULL になるように定義します。そうしないと、カラムに NULL 値、すなわち値なしが許容されることとなります。

[BLOB の考慮事項 \[303 ページ\]](#)

BLOB は、未解釈のバイト文字列または文字列で、値としてカラムに格納されます。

#### 1.2.4.1 NULL と NOT NULL カラム

カラム値がローに対して必須である場合、そのカラムを NOT NULL になるように定義します。そうしないと、カラムに NULL 値、すなわち値なしが許容されることとなります。

デフォルトは NULL 値を許容しますが、NULL 値を許容する正当な理由がないかぎり、カラムには NOT NULL を明示的に宣言してください。

サンプルデータベースには Departments という名前のテーブルがあり、そのテーブルには DepartmentID、DepartmentName、DepartmentHeadID カラムがあります。カラムの定義は次のとおりです。

カラム	データ型	サイズ	NULL/NOT NULL	制約
DepartmentID	integer	-	NOT NULL	プライマリキー
DepartmentName	char	40	NOT NULL	なし
DepartmentHeadID	integer	-	NULL	外部キー

NOT NULL を指定する場合は、テーブルのどのローにも値がなければなりません。

## 1.2.4.2 BLOB の考慮事項

BLOB は、未解釈のバイト文字列または文字列で、値としてカラムに格納されます。

BLOB の一般的な例としては、画像や音声ファイルがあります。BLOB は大きなものが多く、CHAR、VARCHAR、NCHAR、BINARY、VARBINARY などの文字列やバイナリ文字列のデータ型で格納できます。格納する BLOB の内容や長さに応じて、データ型や長さを選択してください。

### i 注記

キャラクターラージオブジェクトは一般に CLOB と呼ばれるのに対し、バイナリラージオブジェクトは BLOB と呼ばれ、両方を組み合わせたものは LOB と呼ばれます。このマニュアルでは、BLOB という略語のみ使用します。

CREATE TABLE 文を使用して BLOB 値を格納するカラムを作成する場合は、ストレージの特性を制御できます。たとえば、指定したサイズ以下の BLOB をロー (インライン) に格納し、指定したサイズを超える BLOB をロー外のテーブル拡張ページに格納するよう指定できます。また、ロー外に格納した BLOB については、プレフィクスとも呼ばれる先頭の *n* バイトをロー内に複製するよう指定できます。このようなストレージ特性は、CREATE TABLE や ALTER TABLE 文で指定された INLINE や PREFIX 設定によって制御されます。これらの設定で指定した値が、パフォーマンスやディスク記憶領域の要件に予期しない影響を与える場合があります。

INLINE と PREFIX のどちらも指定しない場合、あるいは INLINE USE DEFAULT または PREFIX USE DEFAULT を指定した場合、デフォルト値は次のように適用されます。

- CHAR、NCHAR、LONG VARCHAR、XML などの文字データ型のカラムの場合、INLINE のデフォルト値は 256 で、PREFIX のデフォルト値は 8 です。
- BINARY、LONG BINARY、VARBINARY、BIT、VARBIT、LONG VARBIT、BIT VARYING、UUID などのバイナリデータ型のカラムの場合、INLINE のデフォルト値は 256 で、PREFIX のデフォルト値は 0 です。

デフォルト値では対応できない特定の要件がある場合を除き、INLINE と PREFIX の値は設定しないようにしてください。デフォルト値は、パフォーマンスとディスク領域の要件のバランスを取って選択されています。たとえば、INLINE に大きな値を設定し、すべての BLOB をインラインで格納するようにした場合、ローの処理パフォーマンスは低下することがあります。また、PREFIX の値を大きくしすぎると、BLOB の一部を複製したプレフィクスデータのために、BLOB の格納に必要なディスク領域のサイズが増えることになります。

INLINE や PREFIX の値を設定する場合は、INLINE の長さがカラム長を超えないようにしてください。同様に、PREFIX の長さも INLINE の長さを超えないようにしてください。

圧縮されたカラムのプレフィクスデータは圧縮されずに格納されるため、要求を満たすために必要なすべてのデータがプレフィクス内に格納されている場合は、解凍は必要ありません。

## BLOB 共有

BLOB のサイズがインラインサイズを超えて、BLOB の格納に 2 ページ以上のデータベースページが必要になった場合、データベースサーバは同じテーブルの別のローから参照できるよう BLOB を格納します。これは BLOB 共有と呼ばれます。BLOB 共有は内部で処理されます。BLOB 共有はデータベース内で不要に BLOB が複製されないようにするためのものです。

BLOB 共有は、特定の列に別の列と同じ値を設定する場合にのみ発生します。たとえば、`UPDATE t column1=column2;` のように記述します。この例では、`column2` に BLOB が含まれている場合に、`column1` に BLOB を複製する代わりに、`column2` の値を示すように設定します。

BLOB を共有すると、データベースサーバは BLOB を参照している数を追跡します。データベースサーバがテーブル内で参照されている BLOB がないと判断すると、BLOB は削除されます。

圧縮されていない 2 つの列で BLOB を共有している場合に、一方の列を圧縮すると、BLOB は共有でなくなります。

## 1.2.5 列の圧縮の考慮事項

CHAR、VARCHAR、BINARY 列を圧縮して、ディスク領域を節約できます。

たとえば、BMP や TIFF などの大きい BLOB ファイルを格納する列を圧縮できます。圧縮にはデフォルト圧縮アルゴリズムが使用されます。これは、COMPRESS 関数で 사용되는圧縮方式と同じであり、Windows ZIP ファイルで 사용되는アルゴリズムでもあります。

圧縮された列は暗号化されたテーブルの内側に格納できます。この場合、データは最初に圧縮されてから暗号化されます。

文字列が値に圧縮される時に、その値が元の値より 1 ページ以上小さくない場合、文字列は非圧縮で保存されます。また、列のインライン値よりも小さい文字列は圧縮されません。

列を圧縮するには、CREATE TABLE 文と ALTER TABLE 文の COMPRESSED 句を使用します。

列の圧縮による効果を確認するには、sa\_column\_stats システムプロシージャを使用します。

## 関連情報

[テーブル暗号化 \[1621 ページ\]](#)

## 1.2.6 制約の考慮事項

カラムのデータ型は許可されるデータ値を制限しますが (たとえば数字または日付のみなど)、データ値をさらに制限したい場合もあります。

カラム中のデータ値は検査制約を設定して制限できます。CHECK 制約は、指定された条件をカラムやカラムのセットに課す制約です。WHERE 句に表示される有効な条件を使用して、許可される値を制限できます。通常、検査制約では BETWEEN または IN 条件を使用します。

## 1.3 データベースサーバ

SQL Anywhere データベースサーバには、2 つのバージョンがあります。パーソナルサーバ (dbeng17) と、ネットワークサーバ (dbsrv17) です。

### SQL Anywhere データベースサーバ

パーソナルデータベースサーバで作成されたデータベースは、ネットワークデータベースサーバで使用でき、逆についても同様です。パーソナルサーバとネットワークサーバでは、同じ要求処理エンジンが使用され、どちらも同一の SQL 言語と多くの同じデータベース機能がサポートされています。ただし、この 2 つのサーバにはいくつかの違いがあります。

#### パーソナルデータベースサーバ

この実行プログラムは、単一ユーザによる単一コンピュータでの使用を目的として、(たとえば、組み込みデータベースサーバとして) 提供されており、ネットワーク経由のクライアント/サーバ通信はサポートしていません。

Windows オペレーティングシステムでは、パーソナルサーバの実行プログラムの名前は *dbeng17.exe* です。UNIX オペレーティングシステムでは、*dbeng17* です。

#### ネットワークデータベースサーバ

この実行プログラムは、ネットワーク経由のクライアント/サーバ通信をサポートし、複数ユーザでの使用を目的としています。

Windows オペレーティングシステムでは、ネットワークサーバの実行プログラムの名前は *dbsrv17.exe* です。Linux および UNIX オペレーティングシステムでは、*dbsrv17* です。

### 必要なネットワークソフトウェア

ネットワークサーバを実行している場合、適切なネットワークソフトウェアがインストールされて実行されている必要があります。

TCP/IP ネットワークプロトコルがサポートされています。

このセクションの内容:

## SQL Anywhere データベースサーバの内部構造 [307 ページ]

SQL Anywhere データベースサーバは、多くの要求を効率的に処理できる内部構造となっています。

## ネットワークサーバ (dbsrv17) とパーソナルサーバ (dbeng17) の違い [307 ページ]

パーソナルデータベースサーバとネットワークデータベースサーバでは、一般的なデータベースサーバ機能に違いがあります。

## 単一のデータベースサーバ上での複数のデータベースの実行 [309 ページ]

パーソナルデータベースサーバとネットワークデータベースサーバの両方とも、複数のデータベースを同時に管理できます。

## データベースサーバの起動方法 [310 ページ]

データベースサーバの起動方法は、使用しているオペレーティングシステムによって異なります。

## データベースサーバの停止方法 [311 ページ]

データベースサーバの停止には、いくつかの方法があります。

## データベースの起動と停止 [314 ページ]

データベースサーバは、一度に複数のデータベースをロードできます。

## データベースサーバ設定 [319 ページ]

さまざまなデータベースサーバ設定機能が提供されています。

## 最大ページサイズの考慮事項 [319 ページ]

データベースサーバキャッシュは、ページ単位で配置されます。ページとは、メモリ上の固定長の領域です。

## データベースサーバ名とデータベース名 [320 ページ]

-n は、データベースサーバオプション (データベースサーバの指定) またはデータベースオプション (データベースの指定) として使用できます。

## 設定ファイルとデータベースサーバの起動オプション [322 ページ]

設定ファイルにデータベースサーバの起動に使用する一連のオプションを格納し、そのファイルをデータベースサーバコマンドで呼び出すことができます。

## データベースサーバのロギング [322 ページ]

データベースサーバメッセージログには、情報メッセージ、エラー、警告、MESSAGE 文からのメッセージが含まれています。

## Windows イベントログのメッセージ出力の抑制方法 [326 ページ]

レジストリエントリを設定することによって、Windows イベントログのエントリを抑制できます。

## 特殊モード [327 ページ]

SQL Anywhere を特殊モードで実行できます。

## パフォーマンスとメモリの制御方法 [328 ページ]

いくつかの設定は、データベースサーバのパフォーマンスに影響することがあります。

## スレッド [330 ページ]

スレッドモデルを理解するには、スレッドと要求処理の基本用語と概念を理解する必要があります。

## サービスまたはデーモンとしてデータベースサーバを実行する方法 [339 ページ]

データベースサーバをサービスまたはデーモンとして実行するには、いくつかの方法があります。

## OEM Edition の認証アプリケーション [354 ページ]

SQL Anywhere OEM Edition はデータベースのあらゆる操作を実行できます。

## Windows での SQL Anywhere [364 ページ]

SQL Anywhere ソフトウェアを Windows 上で実行する場合には、いくつかの考慮事項が適用されます。

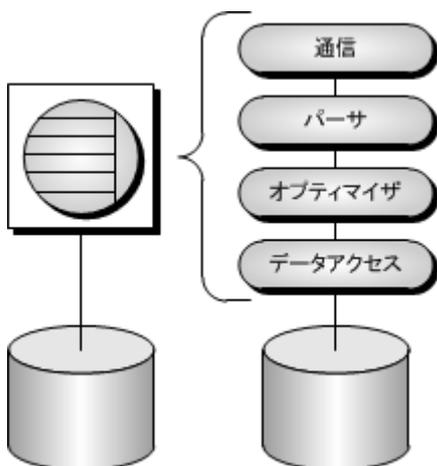
## 関連情報

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

### 1.3.1 SQL Anywhere データベースサーバの内部構造

SQL Anywhere データベースサーバは、多くの要求を効率的に処理できる内部構造となっています。

- 通信レイヤが、クライアントアプリケーションとデータ交換を処理します。このレイヤがクライアントアプリケーションから要求を受信して、その結果を返します。この動作のタイミングは、ネットワークトラフィックを最小に保ちながら、データができるだけ早くクライアント側に届くように、クライアントとサーバのネゴシエーションによって決定されます。
- パーサが、データベースサーバに送信された各 SQL 文をチェックして、それを内部フォームに変換して処理します。
- 要求がクエリ、または更新や削除を行う文の場合は、データにアクセスするいろいろな方法があり、アクセスする時間もそれによってかなり異なる可能性があります。オブティマイザは、必要なデータをすばやく取得するのに最適な方法を選択します。
- データベースサーバの最下位レベルがディスクからデータの読み込みと書き込みを行い、不必要なディスクアクセスを避けるためにデータをメモリにキャッシュし、複数のユーザからの要求のバランスを調整します。



### 1.3.2 ネットワークサーバ (dbsrv17) とパーソナルサーバ (dbeng17) の違い

パーソナルデータベースサーバとネットワークデータベースサーバでは、一般的なデータベースサーバ機能に違いがあります。

パーソナルデータベースサーバでは、同時接続の最大数が 10 に制限されています。要求処理に使用されるのは 1 CPU あたり最大 4 コアまでで、ネットワーククライアント/サーバ接続はサポートされていません。

デフォルトでは、パーソナルデータベースサーバは共有メモリプロトコルのみを使用します。パーソナルデータベースサーバで TCP/IP を使用する場合は、-x オプションを使用する必要があります。

それ以外にも、新しいデータベースの起動に必要なデフォルトの権限レベルや、CHECKPOINT 文の実行に必要な権限などで、多少の違いがあります。

機能	ネットワークデータベースサーバ (dbsrv17)	パーソナルデータベースサーバ (dbeng17)
チェックポイント	CHECKPOINT 文を使用するには、CHECKPOINT システム権限が必要です。	CHECKPOINT 文を使用するために必要なシステム権限は何もありません。
通信パケットの圧縮	サポートされています。	サポートされていません。
ユーティリティデータベースへの接続	デフォルトでは、ユーティリティデータベースへの接続は許可されていません。-su サーバオプションを使用して、ユーティリティデータベースに接続するパスワードを指定する必要があります。	デフォルトでは、ユーティリティデータベースへの接続は、ユーザ ID DBA とパスワードで許可されています。
接続タイプ	デフォルトで、共有メモリと TCP/IP。	デフォルトで共有メモリ。TCP/IP プロトコルに接続するには、-x オプションを使用します。
データベースミラーリング	サポートされています。	サポートされていません。
ネームサーバとしての LDAP	TCP/IP のみと組み合わせて使用できます。	サポートされていません。これは、サポートされている LDAP ユーザ認証と混同しないようにしてください。
マルチプログラミングレベルチューニング	サポートされています。	サポートされていません。
ネットワーク通信	サポートされています。	サポートされていません。
接続数	ライセンスで制限されています。	最大 10 の同時接続。そのうち 3 つは、標準接続用に予約されています。
CPU 数	ライセンスで制限されています。	1 つの CPU 上に最大 4 つのコア。
読み込み専用スケールアウト	サポートされています。	サポートされていません。

## データベースサーバとデータベース起動オプション

次のデータベースサーバとデータベースの起動オプションは、ネットワークデータベースサーバのみによってサポートされています。

- -gna データベースサーバオプション
- -gnh データベースサーバオプション
- -gnl データベースサーバオプション
- -gns データベースサーバオプション
- -pc データベースサーバオプション
- -pt データベースサーバオプション
- -sn データベースオプション
- -xa データベースサーバオプション
- -xf データベースサーバオプション
- -xp データベースオプション

## データベースサーバプロパティ

次のデータベースサーバプロパティは、ネットワークデータベースサーバのみによってサポートされています。

- AutoMultiProgrammingLevel サーバプロパティ
- AutoMultiProgrammingLevelStatistics サーバプロパティ
- MaxMultiProgrammingLevel サーバプロパティ
- MinMultiProgrammingLevel サーバプロパティ
- UniqueClientAddresses サーバプロパティ

## 関連情報

[データベースサーバ起動オプション \[375 ページ\]](#)

[アルファベット順のデータベースオプションリスト \[632 ページ\]](#)

[データベースミラーリング \[1660 ページ\]](#)

[ネームサーバとして LDAP を使用した接続 \[171 ページ\]](#)

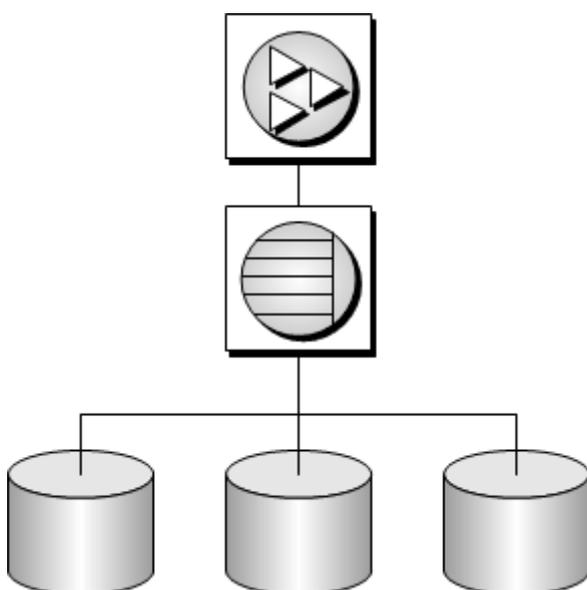
[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[読み込み専用スケールアウト \[1733 ページ\]](#)

### 1.3.3 単一のデータベースサーバ上での複数のデータベースの実行

パーソナルデータベースサーバとネットワークデータベースサーバの両方とも、複数のデータベースを同時に管理できます。

アプリケーションからの 1 つの接続では 1 つのデータベースしか実行できません。しかし、1 つのデータベースサーバを経由して、複数のアプリケーションが別々のデータベースに個別の接続をしたり、1 つのアプリケーションセットから別々のデータベースに接続したりできます。



データベースは、データベースサーバの起動時に開始できます。また、接続パラメータ DatabaseFile (DBF) や START DATABASE 文を使用してデータベースに接続しても開始できます。

このセクションの内容:

#### [他のデータベースのデータへのアクセス \[310 ページ\]](#)

リモートデータアクセス機能を使用して、複数のデータベースサーバ上にあるデータベースにアクセスしたり、また、同じサーバ上にある他の複数のデータベースにもアクセスできます。

### 1.3.3.1 他のデータベースのデータへのアクセス

リモートデータアクセス機能を使用して、複数のデータベースサーバ上にあるデータベースにアクセスしたり、また、同じサーバ上にある他の複数のデータベースにもアクセスできます。

アプリケーションは1つのデータベースに接続していますが、リモートサーバを定義すると、リモートデータベースにあるプロキシテーブルを、現在接続しているデータベースにあるのと同じように使用できます。

### 1.3.4 データベースサーバの起動方法

データベースサーバの起動方法は、使用しているオペレーティングシステムによって異なります。

また、コマンドも、使用しているオペレーティングシステムに応じて、複数の方法で指定できます。

- コマンドプロンプトでコマンドを実行します。
- コマンドをショートカットまたはデスクトップアイコンに配置します。
- バッチファイルでコマンドを実行します。
- コマンドを StartLine (START) 接続パラメータとして接続文字列に含めます。

基本的なコマンドの指定方法は、プラットフォームによって若干違いがあります。

#### i 注記

- 特に指定のないかぎり、これらのコマンドで、ネットワークサーバ (*dbsrv17*) が起動します。パーソナルサーバを起動する場合は、*dbsrv17* を *dbeng17* に置き換えてください。
- データベースファイルがコマンドを開始するディレクトリに含まれている場合は、*path* を指定する必要はありません。
- *database-file* にファイルの拡張子を指定しないと、拡張子は *.db* であると見なされます。

## 例

コマンド	コメント
Windows で、 <a href="#">スタート</a> > <a href="#">プログラム</a> > <a href="#">SQL Anywhere17</a> > <a href="#">SQL Anywhere</a> > <a href="#">パーソナルサーバ</a> をクリックします。	サーバ起動オプションウィンドウが表示され、パーソナルデータベースサーバ (dbeng17 実行プログラム) を起動するための情報を指定できます。データベースサーバでデータベースを起動することもできます。
Windows で、 <a href="#">スタート</a> > <a href="#">プログラム</a> > <a href="#">SQL Anywhere17</a> > <a href="#">SQL Anywhere</a> > <a href="#">ネットワークサーバ</a> をクリックします。	サーバ起動オプションウィンドウが表示され、ネットワークデータベースサーバ (dbsrv17 実行プログラム) を起動するための情報を指定できます。データベースサーバでデータベースを起動することもできます。
<code>dbsrv17 demo</code>	接続文字列でデータベースファイル名を使用します。  <code>demo.db</code> があるディレクトリでこのコマンドを実行して、ネットワークサーバと、 <code>demo.db</code> という名前のデータベースを起動します。
<code>dbsrv17 path¥database-file</code>	このコマンドは、Windows 上でデータベースサーバを起動します。  データベースファイルを省略した場合は、 <a href="#">サーバ起動オプションウィンドウ</a> が表示され、 <a href="#">参照</a> をクリックしてデータベースファイルを検索できます。

## 関連情報

[ネットワークプロトコルオプション](#) [177 ページ]

[-x データベースサーバオプション](#) [500 ページ]

[StartLine \(START\) 接続パラメータ](#) [107 ページ]

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\)](#) [365 ページ]

[トラブルシューティング: データベースサーバの起動](#) [974 ページ]

## 1.3.5 データベースサーバの停止方法

データベースサーバの停止には、いくつかの方法があります。

- データベースサーバメッセージウィンドウで、[シャットダウン](#)をクリックします。
- `dbstop` ユーティリティを使用します。  
`dbstop` ユーティリティは、バッチファイルの中で使用するか、別のコンピュータのサーバを停止するときに便利です。この場合、コマンドに接続文字列が必要です。
- アプリケーションの切断時にデフォルトで自動停止するようにします。
- UNIX でデータベースサーバメッセージウィンドウが前面に表示されているときに Q キーを押します。

このセクションの内容:

### [データベースサーバを停止できるユーザ \[312 ページ\]](#)

データベースサーバの起動時に、-gk オプションを使用して、ユーザが dbstop を使用してサーバを停止するために必要な権限レベルを設定します。

### [オペレーティングシステムセッションのシャットダウン \[312 ページ\]](#)

データベースサーバが実行中のオペレーティングシステムセッションを閉じたり、オペレーティングシステムコマンドを使用してデータベースサーバを停止すると、サーバは正しく停止しません。

### [データベースサーバの停止 \(dbstop ユーティリティ\) \[313 ページ\]](#)

dbstop ユーティリティを使用してデータベースサーバを停止します。

## 1.3.5.1 データベースサーバを停止できるユーザ

データベースサーバの起動時に、-gk オプションを使用して、ユーザが dbstop を使用してサーバを停止するために必要な権限レベルを設定します。

パーソナルデータベースサーバの場合、デフォルト設定は all です。ネットワークデータベースサーバの場合、必要なシステム権限のデフォルトレベルは SERVER OPERATOR ですが、この値を all または none に設定することもできます (ただし、サーバコンピュータを使用すれば、誰でもデータベースサーバメッセージウィンドウでシャットダウンをクリックできます)。

### 関連情報

[-gk データベースサーバオプション \[417 ページ\]](#)

## 1.3.5.2 オペレーティングシステムセッションのシャットダウン

データベースサーバが実行中のオペレーティングシステムセッションを閉じたり、オペレーティングシステムコマンドを使用してデータベースサーバを停止すると、サーバは正しく停止しません。

そのような操作をすると、次回データベースをロードしたときにリカバリが必要になります。このリカバリは自動的に行われます。

データベースサーバを明示的に停止してから、オペレーティングシステムのセッションを閉じることをおすすめします。

次に、サーバを正しく停止しないコマンド例を示します。

- Windows のタスクマネージャでプロセスを停止する
- UNIX の slay または kill コマンドを使用する

### 関連情報

[データベースのバックアップとリカバリ \[894 ページ\]](#)

### 1.3.5.3 データベースサーバの停止 (dbstop ユーティリティ)

dbstop ユーティリティを使用してデータベースサーバを停止します。

#### 前提条件

デフォルトでは、実行中のネットワークサーバを停止するには、SERVER OPERATOR システム権限が必要です。-gk データベースサーバオプションを使用すると、必要な権限を変更できます。

デフォルトでは、どのユーザでもパーソナルデータベースサーバを停止できます。

ただし、そのコンピュータを使用すれば、誰でもデータベースサーバメッセージウィンドウでシャットダウンをクリックできます。

#### コンテキスト

dbstop ユーティリティは、バッチファイルの中で使用するか、別のコンピュータのサーバを停止するときに便利です。この場合、コマンドに接続文字列が必要です。

#### 手順

コマンドプロンプトで、次のような文を dbstop ユーティリティで実行します。ここで、`server-name` は、停止するサーバの名前で、`user-ID` と `password` は接続パラメータです。

```
dbstop -c "Server=server-name;UID=user-ID;PWD=password"
```

#### 結果

データベースサーバが停止します。

#### 例

データベースサーバを起動します。SQL Anywhere インストールディレクトリから、次のコマンドを実行して、サンプルデータベースを使用する Ottawa という名前のサーバを起動します。

```
dbsrv17 -n Ottawa "%SQLANY%SAMP17%¥demo.db"
```

dbstop を使用してサーバを停止します。

```
dbstop -c "Server=Ottawa;UID=DBA;PWD=sql"
```

## 関連情報

[-gk データベースサーバオプション \[417 ページ\]](#)

[サーバ停止ユーティリティ \(dbstop\) \[1163 ページ\]](#)

### 1.3.6 データベースの起動と停止

データベースサーバは、一度に複数のデータベースをロードできます。

次のコマンドで、データベースとデータベースサーバを同時に起動できます。

```
dbsrv17 demo sample
```

#### 警告

データベースファイルは、データベースサーバと同じコンピュータ上に置いてください。ネットワークドライブにあるデータベースファイルを操作すると、ファイルが破損することがあります。

#### 起動中のサーバ上でのデータベースの開始

次のいずれかの方法で、サーバの起動後にもデータベースを起動できます。

- サーバに接続されているときに、DatabaseFile (DBF) 接続パラメータを使用してデータベースに接続します。DatabaseFile (DBF) 接続パラメータは、新規接続用のデータベースファイルを指定します。そのデータベースファイルが現在のサーバ上で起動します。
- START DATABASE 文を使用します。
- SQL Central でサーバを選択し、 **ファイル** > **データベースの開始**  をクリックします。

#### 制限事項

- サーバは、固定サイズのページを使用して、データベース情報をメモリに保持します。サーバがいったん起動すると、サーバより大きいページサイズのデータベースを起動することはできません。
- -gd サーバオプションは、データベースの起動と停止に必要な権限を決定します。

このセクションの内容:

[接続しないでデータベースを起動 \(SQL Central の場合\) \[315 ページ\]](#)

SQL Central を使用して、データベースに接続しないで起動します。

[接続しないでデータベースを起動 \(SQL の場合\) \[316 ページ\]](#)

START DATABASE 文を使用し、データベースに接続しないで起動します。

[接続切断後にデータベースを停止 \(SQL Central の場合\) \[317 ページ\]](#)

現在接続中のデータベースを停止するには、まずデータベースとの接続を切断し、それからデータベースを停止します。

[接続切断後にデータベースを停止 \(SQL の場合\) \[318 ページ\]](#)

接続しているデータベースを停止するため、接続を切断してから、STOP DATABASE 文でデータベースを停止します。

## 関連情報

[組み込みデータベースで使用する接続パラメータ \[41 ページ\]](#)

[最大ページサイズの考慮事項 \[319 ページ\]](#)

[DatabaseFile \(DBF\) 接続パラメータ \[65 ページ\]](#)

### 1.3.6.1 接続しないでデータベースを起動 (SQL Central の場合)

SQL Central を使用して、データベースに接続しないで起動します。

## 前提条件

デフォルトでは、SERVER OPERATOR システム権限が必要です。-gd データベースサーバオプションを使用すると、必要な権限を変更できます。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用して別のデータベースに接続します。
2. 左ウィンドウ枠でデータベースサーバを選択して、**ファイル** > **データベースの開始** をクリックします。
3. **データベースの開始** ウィンドウで、別のデータベースに接続しないで起動するために必要な値を入力します。

## 結果

データベースが、データベースサーバの下に未接続のデータベースとして表示されます。

## 関連情報

[データベースサーバ \[305 ページ\]](#)

### 1.3.6.2 接続しないでデータベースを起動 (SQL の場合)

START DATABASE 文を使用し、データベースに接続しないで起動します。

#### 前提条件

デフォルトでは、SERVER OPERATOR システム権限が必要です。-gd データベースサーバオプションを使用すると、必要な権限を変更できます。

#### 手順

1. 別のデータベースに接続します。
2. START DATABASE 文を実行します。

#### 結果

データベースが起動します。

#### 例

現在のデータベースサーバで、データベースファイル `c:¥¥temp¥¥temp.db` を起動します。

```
START DATABASE 'c:¥¥temp¥¥temp.db'  
AS tempdb  
AUTOSTOP OFF;
```

AUTOSTOP OFF 接続パラメータを使用すると、データベースはすべての接続が切断されても自動的に停止されません。

## 関連情報

[データベースサーバ \[305 ページ\]](#)

### 1.3.6.3 接続切断後にデータベースを停止 (SQL Central の場合)

現在接続中のデータベースを停止するには、まずデータベースとの接続を切断し、それからデータベースを停止します。

#### 前提条件

デフォルトでは、SERVER OPERATOR システム権限が必要です。-gd データベースサーバオプションを使用すると、必要な権限を変更できます。

データベースを停止するには、同じデータベースサーバ上の別のデータベースに接続されている必要があります。

#### 手順

1. 同じデータベースサーバにある別のデータベースに接続していることを確認します。データベースサーバ上でほかに実行されているデータベースがない場合は、ユーティリティデータベースに接続できます。
2. SQL Central で、停止するデータベースを選択し、**ファイル > データベースの停止** をクリックします。

#### 結果

データベースから切断すると、そのデータベースが左ウィンドウ枠に表示されなくなることがあります。現在の接続が残っている唯一の接続で、かつデータベースの起動時に AutoStop を指定した場合は、このような状況になります。AutoStop を指定すると、最後の接続が切断されるときに、データベースが自動的に停止します。

#### 関連情報

[データベースサーバの停止方法 \[311 ページ\]](#)

[ユーティリティデータベースへの接続 \(接続ウィンドウの場合\) \[298 ページ\]](#)

[AutoStop \(ASTOP\) 接続パラメータ \[51 ページ\]](#)

## 1.3.6.4 接続切断後にデータベースを停止 (SQL の場合)

接続しているデータベースを停止するため、接続を切断してから、STOP DATABASE 文でデータベースを停止します。

### 前提条件

デフォルトでは、SERVER OPERATOR システム権限が必要です。-gd データベースサーバオプションを使用すると、必要な権限を変更できます。

データベースを停止するには、同じデータベースサーバ上の別のデータベースに接続されている必要があります。

### コンテキスト

AutoStop (ASTOP) 接続パラメータで起動したデータベースは、最後のユーザが接続を切断すると、自動的に停止します。

### 手順

1. 同じデータベースサーバにある別のデータベースに接続していることを確認します。データベースサーバ上でほかに実行されているデータベースがない場合は、ユーティリティデータベースに接続できます。
2. STOP DATABASE 文を実行します。

### 結果

データベースが停止します。

#### 例

次の文は、ユーティリティデータベースに接続して、tempdb データベースを停止します。

```
CONNECT to 'TestEng' DATABASE utility_db
AS conn2
USER 'DBA'
IDENTIFIED BY 'passwd';
STOP DATABASE tempdb;
```

### 関連情報

[データベースサーバの停止方法 \[311 ページ\]](#)

[ユーティリティデータベースへの接続 \(接続ウィンドウの場合\) \[298 ページ\]](#)

[AutoStop \(ASTOP\) 接続パラメータ \[51 ページ\]](#)

## 1.3.7 データベースサーバ設定

さまざまなデータベースサーバ設定機能が提供されています。

- 各種のオプションを選択して、キャッシュに使用するメモリ量、使用する CPU の数 (ネットワークデータベースサーバを実行中のマルチプロセッサコンピュータ上)、使用するネットワークプロトコル (ネットワークサーバのみ) などのデータベースサーバ設定機能を指定できます。  
オプションは、動作とパフォーマンスをチューニングする主要な方法の 1 つです。
- サーバは、Windows のサービスとして実行できます。サーバをサービスとして実行すると、コンピュータをログオフしてもサーバは稼働し続けます。
- パーソナルサーバは、アプリケーションから起動し、アプリケーションの終了と同時にシャットダウンできます。これは、データベースサーバを組み込みデータベースとして使用する場合に一般的な設定です。

### 関連情報

[サービスまたはデーモンとしてデータベースサーバを実行する方法 \[339 ページ\]](#)

[組み込みデータベースで使用する接続パラメータ \[41 ページ\]](#)

[データベースサーバの起動方法 \[310 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

## 1.3.8 最大ページサイズの考慮事項

データベースサーバキャッシュは、ページ単位で配置されます。ページとは、メモリ上の固定長の領域です。

データベースサーバは停止するまで 1 つのキャッシュを使用するので、ページはすべて同じサイズでなければなりません。

データベースファイルも、データベースの作成時に指定されたサイズのページに配置されます。どのデータベースページも、キャッシュページに適合していなければなりません。デフォルトでは、データベースサーバページサイズは、データベースサーバの起動時に指定されたデータベースの最大ページサイズと同じ大きさです。データベースサーバがいったん起動すると、データベースサーバページより大きいページサイズのデータベースを起動することはできません。

サーバの起動後に大きなページサイズを持つデータベースを起動するには、-gp オプションでページサイズを指定してデータベースサーバを起動します。より大きいページサイズを使用する場合は、必ずキャッシュサイズを増やしてください。キャッシュサイズを変更しないと、大きなページの一部だけが保管され、領域調整の柔軟性が低くなります。

次のコマンドで、64 MB のキャッシュを予約し、最大 8192 バイトのページサイズを使用するデータベースを収容できるデータベースサーバを起動します。

```
dbsrv17 -gp 8192 -c 64M -n myserver
```

## 関連情報

[-gp データベースサーバオプション \[427 ページ\]](#)

### 1.3.9 データベースサーバ名とデータベース名

-n は、データベースサーバオプション (データベースサーバの指定) またはデータベースオプション (データベースの指定) として使用できます。

クライアントアプリケーションは、データベースへの接続時にデータベースサーバ名とデータベース名を指定する接続パラメータを使用できます。データベースサーバ名は、デスクトップアイコンとデータベースサーバメッセージウィンドウのタイトルバーに表示されます。

utility\_db という名前を使用してデータベースを作成したり、データベースサーバを起動したりすることはできません。これは、この名前がユーティリティデータベース用に予約されているためです。

#### データベースサーバの命名

データベースサーバに名前を付けると、ネットワーク上の他のデータベースサーバ名との重複を防ぐことができます。また、クライアントアプリケーションのユーザにわかりやすい名前を提供できます。データベースサーバは、停止するまでその名前を維持します。データベースサーバ名を指定しない場合は、最初に起動されたデータベースの名前になります。

Broadcast Repeater ユーティリティ (dbns17) を使用している場合以外は、同じ名前を持つ異なるサーバを異なるサブネットに配置できます。

最初のデータベースファイルの前に -n オプションを指定すると、データベースサーバに名前を付けることができます。たとえば、次のコマンドは、サンプルデータベースを実行するデータベースサーバを起動し、そのサーバに Cambridge という名前を付けます。

```
dbsrv17 -n Cambridge "%SQLANYSAMPl7%¥demo.db"
```

データベースサーバ名を指定すると、データベースを起動せずにデータベースサーバを起動できます。次のコマンドは、データベースを起動せずに Galt という名前のデータベースサーバを起動します。

```
dbsrv17 -n Galt
```

データベースサーバの名前は、有効な識別子であることが必要です。データベースサーバのロングネームは、プロトコルごとに異なる長さにトランケートされます。データベースサーバ名には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロン、フォワードスラッシュ (/)、またはバックスラッシュ (¥) を含む値
- 長さが 250 バイトを超える値
- UNIX 上で実行するときにスペースを含む値

## i 注記

Windows と UNIX では、データベースサーバがバージョン 10.0.0 以降で、名前が次の長さを超えている場合、バージョン 9.0.2 以前のクライアントから接続することはできません。

- Windows 共有メモリの場合は、40 バイト
- UNIX 共有メモリの場合は、31 バイト
- TCP/IP の場合は 40 バイト

## データベースの命名

クライアントアプリケーションのユーザにわかりやすいデータベース名をつけることをおすすめします。データベースは、停止されるまでその名前で識別されます。データベース名の最大長は 250 バイトです。

データベース名を指定しない場合、デフォルト名はデータベースファイル名のルートとなります (.db 拡張子を持たないファイル名)。たとえば、次のコマンドでは、最初のデータベース名は mydata、次のデータベース名は mysales です。

```
dbsrv17 c:¥mydata.db c:¥sales¥mysales.db
```

データベースファイルの後に -n オプションを指定すると、データベースに名前を付けることができます。たとえば、次のコマンドはサンプルデータベースを起動し、それに MyDB という名前を付けます。

```
dbsrv17 "%SQLANYAMP17¥demo.db" -n MyDB
```

## 大文字と小文字の区別

データベースサーバ名とデータベース名は、文字セットがシングルバイトの場合は大文字と小文字が区別されません。

## 関連情報

[ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

[データベースの起動と停止 \[314 ページ\]](#)

[接続文字列と文字セット \[582 ページ\]](#)

## 1.3.10 設定ファイルとデータベースサーバの起動オプション

設定ファイルにデータベースサーバの起動に使用する一連のオプションを格納し、そのファイルをデータベースサーバコマンドで呼び出すことができます。

設定ファイルには、複数行にわたってオプションを保存できます。たとえば、次の設定ファイルは、データベースサーバとサンプルデータベースを起動します。また、キャッシュを 10 MB に設定し、パーソナルサーバのこのインスタンスの名前を Elora にします。# で始まる行は、コメントとして処理されます。

```
# Configuration file for server Elora
-n Elora
-c 10M
"C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

これらのオプションを含むファイルに `sample.cfg` という名前を付けた場合は、このファイルを次のように指定できます。

```
dbsrv17 @sample.cfg
```

### 関連情報

[設定ファイル \[1066 ページ\]](#)

[@data データベースサーバオプション \[382 ページ\]](#)

[設定ファイルでの条件付き解析 \[1068 ページ\]](#)

## 1.3.11 データベースサーバのロギング

データベースサーバメッセージログには、情報メッセージ、エラー、警告、MESSAGE 文からのメッセージが含まれています。

これらのメッセージは、次の場所に表示されます。

- データベースサーバメッセージウィンドウ (Windows の場合はシステムトレイアイコン)
- SQL Central の [サーバメッセージと実行された SQL ウィンドウ](#) 枠
- コックピット
- データベースサーバメッセージログファイル
- データベースサーバをコマンドラインアプリケーションとして実行した場合のコマンドプロンプトウィンドウまたはシェル
- UNIX Syslog

このセクションの内容:

[ファイルへのデータベースサーバメッセージのロギング方法 \[323 ページ\]](#)

デフォルトでは、データベースサーバメッセージはデータベースサーバメッセージウィンドウに送信されますが、`-o` オプションを使用して結果をデータベースサーバメッセージログファイルに送信することもできます。

[SQL 文のロギング \[324 ページ\]](#)

SQL Central での SQL 文のロギングをオンにします。

### [データベース変更のロギングの停止 \[325 ページ\]](#)

SQL Central を使用して、データベース変更のロギングを停止します。

### [UNIX で機能識別子を使用してメッセージをファイルヘリダイレクト \[325 ページ\]](#)

UNIX で、データベースサーバメッセージを、システム管理者が使用できるファイルにリダイレクトします。

## 関連情報

[-o データベースサーバオプション \[452 ページ\]](#)

[-oe データベースサーバオプション \[453 ページ\]](#)

[-on データベースサーバオプション \[454 ページ\]](#)

[-os データベースサーバオプション \[455 ページ\]](#)

[-ot データベースサーバオプション \[456 ページ\]](#)

## 1.3.11.1 ファイルへのデータベースサーバメッセージのロギング方法

デフォルトでは、データベースサーバメッセージはデータベースサーバメッセージウィンドウに送信されますが、`-o` オプションを使用して結果をデータベースサーバメッセージログファイルに送信することもできます。

次のコマンドを実行して、結果を `mydbserver_messages.txt` という名前のメッセージログファイルに送ります。

```
dbsrv17 -o mydbserver_messages.txt -c ...
```

データベースサーバメッセージログファイルのサイズを制御したり、ファイルが最大サイズに達したときの処理を指定したりできます。

- `-o` オプションを使用して、データベースサーバメッセージログファイルを使用することを指定し、ファイル名を入力します。
- `-ot` オプションを使用して、データベースサーバメッセージログファイルを使用することを指定し、ファイル名を入力すると、新しいメッセージが送信される前にログファイルの前の内容が削除されます。
- `-o` または `-ot` に加えて `-on` オプションを使用してサイズを指定すると、そのサイズに達したときに、これまでのデータベースサーバメッセージログファイルの名前に拡張子 `.old` が付けられて変更され、元の名前を持つ新しいファイルが使用されます。
- `-o` または `-ot` に加えて `-os` オプションを使用してサイズを指定すると、そのサイズに達したときに、日付と連番に基づいた新しい名前を持つ新しいデータベースサーバメッセージログファイルが使用されます。

起動エラー、致命的なエラー、アサーションのロギング先をそれぞれ別ファイルにするには、`-oe` オプションを指定します。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する可能性があるため、データベースサーバメッセージログファイル名の最後には `.log` を付けないようにしてください。

## 関連情報

[-o データベースサーバオプション \[452 ページ\]](#)

- oe データベースサーバオプション [453 ページ]
- on データベースサーバオプション [454 ページ]
- os データベースサーバオプション [455 ページ]
- ot データベースサーバオプション [456 ページ]

## 1.3.11.2 SQL 文のロギング

SQL Central での SQL 文のロギングをオンにします。

### コンテキスト

SQL Central でデータベースを使用する場合、アクションに応じて自動的に SQL 文が生成されます。

これらの文は、**サーバメッセージと実行された SQL** という別のウィンドウ枠で追跡できます。または情報をファイルに保存できます。

**サーバメッセージと実行された SQL** ウィンドウ枠には、データベースとデータベースサーバごとにタブがあります。データベースサーバのタブには、データベースサーバメッセージウィンドウと同じ情報が表示されます。

### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. SQL Central によって生成された SQL 文の、**サーバメッセージと実行された SQL** ウィンドウ枠へのロギングをオンにするには、次の手順に従います。
  - a. **表示 > サーバメッセージと実行された SQL** をクリックします。
  - b. **サーバメッセージと実行された SQL** ウィンドウ枠で、データベースアイコンがあるタブをクリックします。
  - c. 右クリックして、ドロップダウンメニューの**オプション**をクリックします。
  - d. ロギングのオプションを編集します。
  - e. **保存** をクリックします。
3. SQL Central によって生成された SQL 文のファイルへのロギングをオンにします。
  - a. 左ウィンドウ枠で、データベースを右クリックし、**データベース変更のロギングの開始**をクリックします。
  - b. ファイル名を指定し、**保存**をクリックします。

### 結果

データベース変更のログが取られます。

## 関連情報

[文のロギング \(Interactive SQL\) \[1031 ページ\]](#)

### 1.3.11.3 データベース変更のロギングの停止

SQL Central を使用して、データベース変更のロギングを停止します。

#### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左のウィンドウ枠でデータベースを右クリックします。
3. [データベース変更のロギングの停止](#)をクリックします。

#### 結果

データベース変更のログが取られなくなります。

### 1.3.11.4 UNIX で機能識別子を使用してメッセージをファイルへリダイレクト

UNIX で、データベースサーバメッセージを、システム管理者が使用できるファイルにリダイレクトします。

#### コンテキスト

次の手順は、Solaris でのメッセージのリダイレクト方法を説明したのですが、Linux や IBM AIX など他のプラットフォームでも実行できます。

HP-UX などの他のプラットフォームでは、`syslog.conf` ファイルは別の場所にあります。`/var/adm/sqlanywhere` ファイルは、任意の場所に配置できます。

## 手順

1. システムで動作中の他のアプリケーションが使用していない一意の機能識別子を選択します。

`/etc/syslog.conf` ファイルを見ると、どの `localn` 機能が参照されているかを確認できます。

2. `/etc/syslog.conf` ファイルを開き、次の行を追加します。`localn` は、手順 1 で選択した機能識別子です。

```
localn.err;localn.info;localn.notice /var/adm/sqlanywhere
```

3. `/var/adm/sqlanywhere` ファイルを作成します。

```
touch /var/adm/sqlanywhere
```

4. `syslog` のプロセス ID を検索して、`syslog.conf` ファイルを変更したことを `syslog` プロセスに通知します。

```
ps -ef | grep syslog
```

次に、以下のコマンドを実行します。`pid` は、`syslog` のプロセス ID です。

```
kill -HEAP pid
```

5. 次のコマンドを使用してデータベースサーバを起動します。`localn` は、手順 1 で選択した機能識別子です。

```
dbsrv17 -s localn ...
```

## 結果

データベースサーバが Syslog にレポートするメッセージがすべて、`/var/adm/sqlanywhere` ファイルにリダイレクトされます。

## 関連情報

[-s データベースサーバオプション \[468 ページ\]](#)

## 1.3.12 Windows イベントログのメッセージ出力の抑制方法

レジストリエントリを設定することによって、Windows イベントログのエントリを抑制できます。

レジストリのエントリは `Software\SAP\SQL Anywhere\17.0.4` です。このエントリは、`HKEY_CURRENT_USER` または `HKEY_LOCAL_MACHINE` ハイブのいずれかに配置できます。

イベントログエントリを制御するには、種類が `REG_DWORD` の `EventLogMask` キーを設定します。この値はビットマスクで、さまざまなイベントメッセージに関する内部ビット値を保有します。

```
errors          EVENTLOG_ERROR_TYPE          0x0001
```

```
warnings      EVENTLOG_WARNING_TYPE      0x0002
information   EVENTLOG_INFORMATION_TYPE   0x0004
```

EventLogMask キーの設定を変更した場合、変更を有効にするには、データベースサーバを再起動する必要があります。

#### 例

たとえば、EventLogMask キーを 0 に設定すると、どのメッセージも出力されなくなります。このキーを 1 に設定すると、情報メッセージと警告メッセージは出力されませんが、エラーメッセージは出力されます。デフォルト設定 (エントリが存在しない場合) では、すべてのメッセージタイプが出力されます。

## 関連情報

[ネットワークプロトコルオプション \[177 ページ\]](#)

### 1.3.13 特殊モード

SQL Anywhere を特殊モードで実行できます。

#### 読み込み専用

-r オプションを入力すると、データベースを読み込み専用モードで起動できます。監査がオンになっているデータベースを読み込み専用モードで起動することはできません。

#### インメモリモード

-im オプションを指定すると、データベースを完全にインメモリで実行できます。チェックポイントモードのみ (-im c) で実行すると、データベースサーバはトランザクションログを使用しませんが、最新のチェックポイントにリカバリできます。非書き込みモード (-im nw) でデータベースを実行すると、コミットされたトランザクションはディスク上のデータベースファイルに書き込まれないため、データベースを停止すると、すべての変更は失われます。どちらのインメモリモードを使用した場合も、データベースがアクティブである間は、アプリケーションから変更を加えたり、アクセスしたりできます。

#### バルクロード

これは、Interactive SQL の INPUT 文を使用してデータベースに大量のデータをロードするときに便利です。LOAD TABLE を使用してデータをバルクロードする場合は、-b オプションは使用しないでください。

#### トランザクションログなしの起動

-f データベースオプションは、リカバリ時、つまりトランザクションログの消失後にデータベースサーバを起動したり、トランザクションログが見つからないときにデータベースサーバを起動したりする場合に使用します。-f オプションはサーバオプションではなく、データベースオプションです。

リカバリが完了したら、サーバを停止し、-f オプションを指定せずに再起動してください。

#### クワイエットモードで作動する

データベースサーバはクワイエットモードをサポートしています。サーバをどのようなクワイエットモードで操作するかを決定します。クワイエットモードの範囲には、メッセージやシステムトレイのアイコンを非表示にするモードから、完全に非表示にするモードまであります。Windows 上のデータベースサーバを完全に非表示にするモードで操作するには、-qi、-qs、-qw オプションを指定します。これらのオプションを設定すると、すべてのアイコンとすべての起動エラーメッセージが

表示されなくなるので、サーバが実行中であることを視覚的に表示するものがなくなります。データベースサーバをクワイエットモードで実行する場合、-o または -oe オプションのいずれか（または両方）を使用してエラーを診断できます。

-qi と -qs オプションを使用しても、-v (バージョン) と -ep (データベース暗号化パスワードを要求) サーバオプションによるウィンドウは表示されます。

## 関連情報

- r データベースサーバオプション [467 ページ]
- r データベースオプション [531 ページ]
- im データベースサーバオプション [437 ページ]
- b データベースサーバオプション [385 ページ]
- f データベースサーバオプション [408 ページ]
- qi データベースサーバオプション [462 ページ]
- qn データベースサーバオプション [463 ページ]
- qs データベースサーバオプション [465 ページ]
- qw データベースサーバオプション [466 ページ]

## 1.3.14 パフォーマンスとメモリの制御方法

いくつかの設定は、データベースサーバのパフォーマンスに影響することがあります。

### キャッシュサイズ

データベースサーバに割り当て可能なキャッシュメモリの容量は、パフォーマンスに影響を及ぼす主要な要因の 1 つになります。データベースサーバが利用できるメモリが多いほど、実行速度が速くなります。キャッシュには何度も要求される情報が保持されます。ディスクの情報にアクセスするよりも、キャッシュ内の情報にアクセスする方が速くなります。デフォルトの初期キャッシュサイズは、物理メモリの容量、オペレーティングシステムとデータベースファイルのサイズに基づいて計算されます。データベースサーバは、必要に応じてキャッシュサイズを自動的に調整します。

データベースサーバメッセージウィンドウには起動時のキャッシュサイズが表示されます。また、次の文を使用して現在のキャッシュサイズを取得することもできます。

```
SELECT PROPERTY( 'CurrentCacheSize' );
```

次の表は、キャッシュの制御に使用できるデータベースサーバオプションをまとめたものです。

キャッシュ機能	データベースサーバオプション	用途
キャッシュサイズ	-c	データベースサーバキャッシュ用のメモリの初期容量を設定します。
	-ca 0	静的キャッシュサイズを強制的に適用します。
	-ch	自動キャッシュサイズ変更の最大キャッシュサイズを設定します。

キャッシュ機能	データベースサーバオプション	用途
	-chx	キャッシュ以外の用途のアドレス領域を予約しないで自動キャッシュサイズ変更の最大キャッシュサイズを設定します (32 ビットデータベースサーバの場合のみ)。
	-cl	自動キャッシュサイズ変更の最小キャッシュサイズを設定します。
	-cs	データベースサーバメッセージウィンドウに、動的なキャッシュサイズ変更に関する統計を表示します。
キャッシュウォーミング	-cc	次回にデータベースが起動されるときに、キャッシュウォーミングに使用するデータベースページに関する情報を収集します。
	-cr	データベースページを保持するキャッシュを準備します。
	-cv	データベースサーバメッセージウィンドウに、キャッシュウォーミングに関するメッセージを表示します。

#### マルチプログラミングレベル

データベースサーバのマルチプログラミングレベルは、同時に実行できるデータベースサーバタスクの最大数を指定するものです。一般に、マルチプログラミングレベルを上げるほど、同時に実行する要求の数が増えるため、データベースサーバの全体的なスループットは向上します。ただし、マルチプログラミングレベルを上げすぎると、同じリソースを使用する要求が複数存在するときに、余分な競合が発生し、トランザクションの応答時間が長くなります。

デフォルトでは、データベースサーバのマルチプログラミングレベルは自動的に調整されます。マルチプログラミングレベルを上げると、場合によっては、システムのスループットが低下することもあります。次のオプションを使用すると、データベースサーバのマルチプログラミングレベルを手動で制御できます。

データベースサーバオプション	sa_server_option 値	説明
-gn データベースサーバオプション	CurrentMultiProgrammingLevel	データベースサーバのマルチプログラミングレベルを設定します。
-gna データベースサーバオプション	AutoMultiProgrammingLevel	データベースサーバのマルチプログラミングレベルの動的チューニングをオンまたはオフにします。
-gnh データベースサーバオプション	MaxMultiprogrammingLevel	データベースサーバが同時に実行できるタスクの最大数を設定します。
-gnl データベースサーバオプション	MinMultiProgrammingLevel	データベースサーバが同時に実行できるタスクの最小数を設定します。
-gns データベースサーバオプション	AutoMultiProgrammingLevelStatistics	マルチプログラミングレベルの自動変更に関する統計をデータベースサーバメッセージログに出力するかどうかを制御します。
-gta データベースサーバオプション	ProcessorAffinity	どの論理プロセッサを Windows 上または Linux 上で使用するかをデータベースサーバに指示します。

#### プロセッサの数

ネットワークデータベースサーバを使用するマルチプロセッサコンピュータを実行している場合、-gt オプションを使用してプロセッサ数を設定できます。

データベースサーバで使用できる CPU 数には、ライセンスまたは SQL Anywhere のエディションも影響する場合があります。

その他のパフォーマンスに関連するオプション

ネットワークのパフォーマンスをチューニングするオプションには、-gb (データベース処理優先度) と -u (バッファディスク I/O) などいくつかのオプションがあります。

## 関連情報

[データベースサーバ起動オプション \[375 ページ\]](#)

[動的キャッシュサイズ決定 \[1373 ページ\]](#)

[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[スレッド \[330 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

## 1.3.15 スレッド

スレッドモデルを理解するには、スレッドと要求処理の基本用語と概念を理解することが必要です。

### 要求

要求は、クエリや SQL 文などの作業単位で、接続を介してデータベースサーバに送信されます。要求の存続期間は、要求がデータベースによって最初に受信されてから、結果の最後が返されてカーソルが閉じられるまで、または要求がキャンセルされるまでの間を指します。

### タスク

タスクは、データベースサーバ内で実行されるアクティビティの単位です。また、データベースサーバによってスケジューラされる最小の作業単位でもあります。ユーザ要求は、それぞれ、データベースサーバ内で少なくとも 1 つのタスクになります。クエリ内並列処理が関連する場合は、複数のタスクになることもあります。データベースサーバは、ユーザ要求だけでなく、内部タスクを実行するために自身のタスクをスケジューラすることもできます。たとえば、データベースクリーナーの実行やタイマーの処理などがこれに該当します。同時に実行できるアクティブなタスクの最大数は、カーネル内のワーカプールのサイズによって異なります。ユーザ要求タスクがデータベースサーバに到達し、それにワーカが割り当てられると、ActiveReq サーバプロパティの値が 1 増えます。要求が完了すると、ActiveReq サーバプロパティの値は 1 減ります。ただし、ユーザ要求タスクを実行できるワーカがない場合には、タスクは実行待ちのキューに入れられ、UnschReq サーバプロパティの値が 1 増えます。ワーカが利用可能になり、タスクが実行待ちのキューから取り出されると、UnschReq サーバプロパティの値は 1 減り、ActiveReq プロパティの値は 1 増えます。

### ワーカ

ワーカは、データベースサーバカーネル内で実行されている制御用のスレッドを抽象化したものです。Windows と Linux では、ワーカは「ファイバー」と呼ばれる軽量スレッドを使用して実装されます。その他のプラットフォームでは、オペレーティングシステムスレッドを使用して実装されます。ワーカは、データベースサーバのカーネルによって割り当てられたタスクを実行します。データベースサーバは、サイズが可変のワーカプールを利用してサーバの負荷を処理します。プールのサイズは、データベースサーバのマルチプログラミングレベルに対応します。

パーソナルサーバには固定サイズのプールがあります。これらのサーバでは、サーバの起動時に作成されるワーカの数  
を `-gn` オプションで制御します。他のすべてのネットワークサーバでは、`-gnh` オプションで指定された数のワーカがデー  
タベースサーバによって作成されます。ただし、タスクに対してサービスを提供できるのは、サーバのマルチプログラミング  
レベルで指定された数のワーカのみです。デフォルトでは、ワーカプールのサイズは、データベースサーバの負荷の変  
化に応じてカーネルによって動的にチューニングされ、`-gnl` オプションと `-gnh` オプションでそれぞれ指定された下限と上  
限の間で上下します。

タスクは、先入れ先出し (FIFO) 方式でワーカに割り当てられます。各タスクの優先度は、そのタスクを生成した接続に基  
づいて設定されます。タスクが実行のためにワーカに割り当てられると、カーネルのスケジューラーはタスクの優先度を  
考慮して CPU 時間をそのワーカに割り付けます。タスクの実行中、ロックを待機したり、I/O 処理の完了を待機したりす  
るなどの理由でタスクのブロックが必要になった場合でも、ワーカはそのタスクに関連付けられたままです。タスクが完了  
した後にのみ、ワーカは他のタスクを実行するために使用できるようになります。

#### スレッド

スレッド、つまり実行スレッドは、オペレーティングシステムを構築するもので、単一プロセス内での同時実行を可能にしま  
す。データベースサーバを含め、オペレーティングシステムの各プロセスは、少なくとも1つのスレッドによって実行されま  
す。複数のスレッドによって実行されることもあります。スレッドは、オペレーティングシステムによってアプリケーション外  
部でスケジュールされ、最終的にアプリケーション実行のすべてはスレッドによって行われます。Windows と Linux で  
は、データベースサーバは決められた数のスレッドを作成します。つまり、CPU コアごとに1つのオペレーティングシステ  
ムスレッドを作成し、この数は `-gtc` オプションで制御されます。その他のプラットフォームでは、作成されるオペレーティ  
ングシステムスレッドの数はワーカプールのサイズと同じであり、ワーカプールサイズの制御に使用されるのと同じメカニ  
ズムで制御されます。

スレッドの数は、データベースへの接続の数とは関係ありません。データベースサーバは、スレッドを特定の接続専用  
に割り当てません。代わりに、タスクは、実行のためにデータベースサーバに届くと、サーバスレッドの固定サイズのプ  
ールからスレッドが動的に割り当てられます。タスクがスレッドに割り当てられると、スレッドはタスクが完了するかキャン  
セルされるまでタスクを処理します。

このセクションの内容:

#### [UNIX でのワーカ \[332 ページ\]](#)

UNIX では、ワーカはオペレーティングシステムスレッドを使用して実装されます。

#### [Windows と Linux でのワーカ \[332 ページ\]](#)

Windows と Linux の場合、ワーカはファイバーと呼ばれる軽量スレッドを使用して実装されます。

#### [スレッド動作 \[333 ページ\]](#)

スレッドの動作は、いくつかの要素によって制御されます。これらは、サーバオプションによって管理されます。すべての  
プラットフォームで、これらのオプションのすべてがサポートされるわけではありません。

#### [マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

データベースサーバのマルチプログラミングレベルは、同時にアクティブにできるタスクの最大数です

## 関連情報

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gnl データベースサーバオプション \[424 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

### 1.3.15.1 UNIX でのワーカ

UNIX では、ワーカはオペレーティングシステムスレッドを使用して実装されます。

したがって、タスクのスケジューリングはオペレーティングシステムのスケジューラーによって制御され、オペレーティングシステムはいつでもスレッドの実行を先取りする可能性があります。このプリエンティブスケジューリングは、タスクの処理に対して、目に見える形で影響を及ぼすことはありません。スレッドが再び実行されるようにスケジューリングされると、タスクは中断したところから処理が開始されます。ただし、データベース接続から継承されたタスクの優先度は、それらのタスクの実行スケジュールに影響を及ぼしません。代わりに、すべてのオペレーティングシステムスレッドの優先度が同じであるため、すべてのタスクは同じ優先度レベルで実行されます。

### 1.3.15.2 Windows と Linux でのワーカ

Windows と Linux の場合、ワーカはファイバーと呼ばれる軽量スレッドを使用して実装されます。

ファイバーを利用した実装によって、オペレーティングシステムスレッドのスケジューラーでワーカを先取りでスケジューリングするのではなく、協調的にスケジューリングできます。その結果、ファイバー間の切り替えをより詳細に制御できるようになります。データベースカーネルは、プロセッサのアフィニティを最大にして各ワーカの実行優先度を密接に制御するようにファイバーをスケジューリングします。

ファイバーは、オペレーティングシステムスケジューラーに依存しないため、他のアクティビティの完了を待機しているときは、制御を別のファイバーに明示的に譲ります。たとえば、ファイバーで実行中のタスクにおいて、I/O 操作の完了を待機するなどの理由でブロックが必要になると、ファイバーは制御を解放して他のファイバーに譲ります。元のファイバーを実行しているスレッドは、カーネルを介した切り替えを行うことなく、即座に別のファイバーを選択して実行を開始できます。ファイバーがブロックしたが制御を譲らなかった場合、そのファイバーを実行しているスレッドがブロックされ、他のファイバーがそのスレッドで実行できなくなります。複数のスレッドがファイバーを実行している場合、待機中のファイバーを実行しているスレッドだけがブロックされます。他のスレッドは、ファイバーを自由に実行できます。

Windows と Linux プラットフォームには、少なくとも、データベースサーバの最大同時実行性設定 (-gnh オプションで指定) で求められた数と同数のファイバーが存在します。データベースサーバは、サーバの実行中に、現在のサーバマルチプログラミングレベルに対応する、これらのファイバーのサブセットを利用します。ファイバーごとに、オペレーティングシステムはスタックのアドレス領域を予約する必要があります。各ファイバーに必要なスタックのサイズは、-gss サーバオプションの設定によって異なります。

ネットワークサーバに必要な実行スタックのアドレス領域のサイズは、最大マルチプログラミングレベルに比例します。

#### 関連情報

[-gss データベースサーバオプション \[429 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

### 1.3.15.3 スレッド動作

スレッドの動作は、いくつかの要素によって制御されます。これらは、サーバオプションによって管理されます。すべてのプラットフォームで、これらのオプションのすべてがサポートされるわけではありません。

#### マルチプログラミングレベル (-gn サーバオプション)

-gn オプションは、データベースサーバのマルチプログラミングレベルを制御します。この値によって、同時にアクティブにできるタスクの最大数が決定されます。データベース要求ごとに、少なくとも 1 つのタスクが作成されます。クエリ内並列処理が関連する場合は、複数のタスクが作成されることもあります。また、サーバは、内部的な処理を行うために臨時にタスクをスケジューリングすることもあります。サーバ内にあるタスクの数がマルチプログラミングレベルを超えると、未処理のタスクは、現在実行中のタスク (アクティブなタスク) が完了するのを待機します。

#### 内部実行スレッドあたりのスタックサイズ (-gss サーバオプション)

-gss オプションを使用して、データベースサーバのワーカあたりのスタックサイズを設定できます。-gss オプションによって、データベースサーバ内の各ワーカに必要なアドレス領域を減らすことができます。これは、メモリが限られている環境で役立ちます。

#### プロセッサの数 (-gt サーバオプション)

-gt オプションは、データベースサーバが使用するプロセッサの数を制御します。

#### プロセッサの同時実行性 (-gtc サーバオプション)

-gtc オプションは、データベースサーバが使用する論理プロセッサ (コア) の数を指定します。

#### トポロジを考慮したスケジューリング (-gtp server option)

-gtp オプションは、利用可能な各コアごとにスレッドを 1 つずつ使用してから、コアの 2 番目のスレッドを使用するという方法でタスクのスケジューリングを行うよう指定します。

### プロセッサの使用とスレッド処理の例

次の例は、-gt と -gtc の設定に基づいて、データベースサーバが CPU を選択する方法を示します。この例では、システムが 4 プロセッサ構成で、各プロセッサにコアが 2 つあることを前提としています。物理プロセッサは文字で、コアは数字でそれぞれ区別します。したがって、このシステムにはプロセッサユニットとして A0、A1、B0、B1、C0、C1、D0、D1 が存在することになります。

シナリオ	ネットワークデータベースサーバの設定
シングル CPU ライセンス、または -gt に 1 が指定されている	<ul style="list-style-type: none"><li>• -gt 1</li><li>• -gtc 2</li></ul> スレッドは A0 または A1 で実行できます。
CPU のライセンスに制限がなく、-gtc に 5 が指定されている	<ul style="list-style-type: none"><li>• -gt 4</li><li>• -gtc 5</li></ul> スレッドは A0、A1、B0、C0、D0 で実行できます。

シナリオ	ネットワークデータベースサーバの設定
データベースサーバに 3 CPU ライセンスがあり、-gtc に 5 が指定されている	<ul style="list-style-type: none"> <li>• -gt 3</li> <li>• -gtc 5</li> </ul> <p>スレッドは A0、A1、B0、B1、C0 で実行できます。</p>
CPU のライセンスに制限がなく、-gtc に 1 が指定されている	<ul style="list-style-type: none"> <li>• -gt 4</li> <li>• -gtc 1</li> </ul> <p>スレッドは A0 でのみ実行できます。</p>

## 関連情報

[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gss データベースサーバオプション \[429 ページ\]](#)

[-gt データベースサーバオプション \[430 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

### 1.3.15.4 マルチプログラミングレベルのデータベースサーバ設定

データベースサーバのマルチプログラミングレベルは、同時にアクティブにできるタスクの最大数です

クライアント側の要求がデータベースサーバに届くと、その要求用に作成されたタスクがワーカ (使用可能な場合) に割り当てられます。ワーカが割り当てられた要求はアクティブな要求と呼ばれます。使用可能なすべてのワーカがビジー状態の場合、要求はスケジュールされていない要求キューと呼ばれる特殊なキューに格納され、要求はスケジュールされていない要求として分類されます。同様に、アクティブなタスクは、ワーカによって現在実行されているタスクです。アクティブなタスクは、アクセスプランの演算子を実行したり、その他のファンクションを実行したりします。ただし、(I/O 操作やローロックなどのために) リソースを待機することもあります。スケジュールされていないタスクは、実行の準備が整っているものの、利用可能なワーカを待機しているタスクを指します。同時に実行できるアクティブなタスクの数は、サーバのスレッドの数と、コンピュータに搭載されている論理プロセッサの数によって異なります。

SQL Anywhere を使用すると、DBA は、データベースサーバがサーバスループットに基づいてマルチプログラミングレベルを動的にチューニングできるようにするか (デフォルト)、またはマルチプログラミングレベルを手動で設定するかを選択できます。マルチプログラミングレベルは、データベースサーバの起動時にデータベースサーバオプション (-gna、-gnl、-gnh) を指定して設定するか、またはデータベースサーバの実行後に sa\_server\_option システムプロシージャで MinMultiprogrammingLevel、MaxMultiprogrammingLevel、CurrentMultiprogrammingLevel プロパティを使用して設定できます。

次の表は、データベースサーバのマルチプログラミングレベルを制御するコマンドラインオプションとサーバオプションをまとめたものです。

データベースサーバオプション (データベースサーバの起動時)	sa_server_option 値 (データベースサーバの実行中)	説明
-gn	CurrentMultiProgrammingLevel	データベースサーバのマルチプログラミングレベルを設定します。
-gna	AutoMultiProgrammingLevel	データベースサーバのマルチプログラミングレベルの動的チューニングをオンまたはオフにします。
-gnh	MaxMultiprogrammingLevel	ネットワークデータベースサーバが同時に実行できるタスクの最大数を設定します。
-gnl	MinMultiProgrammingLevel	ネットワークデータベースサーバが同時に実行できるタスクの最小数を設定します。

## マルチプログラミングレベルのチューニング

SQL Anywhere ネットワークサーバは、サーバのマルチプログラミングレベルの動的チューニングと手動チューニングの両方をサポートしています。

### マルチプログラミングレベルの動的チューニング

SQL Anywhere ネットワークサーバは、データベースサーバのスループットレベルを自動的にモニタし、現在の負荷に応じてマルチプログラミングレベルを調整する方法を決定できます。データベースサーバは山登り法というアルゴリズムと放物線近似法を使用して、必要な調整を決定します。マルチプログラミングレベルを上げるとネットワークデータベースサーバのスループットレベルが上がる場合、ネットワークデータベースサーバはそのまま処理を続行します。マルチプログラミングレベルを上げるとスループットが低下する場合、データベースサーバはマルチプログラミングレベルを下げます。データベースサーバはスループットレベルを常時モニタし、サーバスループットが向上するようにマルチプログラミングレベルを変更します。多数の要求が短時間に集中し、その後長いアイドル時間が続く場合は、最小マルチプログラミングレベルを予想される最大同時実行レベルに設定することをお奨めします。このように設定すると、データベースサーバは要求が短時間に集中している間でも応答できます。

マルチプログラミングレベルの動的チューニングは、デフォルトでは有効になっています (-gna 1)。データベースサーバオプション -gnl と -gnh は、動的チューニングアルゴリズムで使用されるマルチプログラミングレベルの最小値と最大値をそれぞれ設定します。動的チューニングが有効になっている場合、スレッドデッドロックが発生するたびに、データベースサーバはマルチプログラミングレベルを自動的に上げてスレッドデッドロックの問題を排除しようとします。データベースサーバは、許容される MaxMultiprogrammingLevel 値になるまでマルチプログラミングレベルを上げ続けます。MaxMultiprogrammingLevel 値に達すると、データベースサーバはスレッドデッドロック問題をクライアントアプリケーションに返し始めます。

### マルチプログラミングレベルの手動チューニング

マルチプログラミングレベルを手動で調整するには、まず、その動的チューニングを無効にする必要があります。アプリケーションの負荷をテストして、サーバスループットと要求の応答時間に対する、データベースサーバのマルチプログラミングレベルの効果を分析することをお奨めします。Windows パフォーマンスモニタまたは UNIX 上のパフォーマンス統計ユーティリティ (dbstats) を使用すると、アプリケーションのテスト時にデータベースサーバの動作を簡単に分析できます。PROPERTY、DB\_PROPERTY、CONNECTION PROPERTY 関数を使用すると、パフォーマンス統計を問い合わせることができます。

この分析中に、アクティブな要求やスケジュールされていない要求に関連のあるパフォーマンスカウンタを確認してください。

アクティブな要求の数が、-gn データベースサーバオプションの値よりも常に小さい場合は、マルチプログラミングレベルを下げることを検討します。ただし、データベースサーバの実行キューに対してタスクを追加するクエリ内並列処理の効果を考慮に入れる必要があります。クエリ内並列処理の効果が不十分である場合は、全体的なシステムスループットを低下させることなく、安全に、マルチプログラミングレベルを下げることができます。ただし、要求の総数（アクティブな要求 + スケジュールされていない要求）が -gn で指定された値よりも大きくなることが多い場合は、マルチプログラミングレベルを上げることができます。マルチプログラミングレベルを上げた場合のパフォーマンスのトレードオフを検討してからレベルを変更してください。

このセクションの内容:

#### [各種マルチプログラミングレベル設定の効果 \[336 ページ\]](#)

マルチプログラミングレベルの設定はパフォーマンスに影響を与えます。

#### [マルチプログラミングレベルの手動調整 \[338 ページ\]](#)

ネットワークサーバのマルチプログラミングレベルを手動で調整できます。

## 関連情報

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gna データベースサーバオプション \[422 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gni データベースサーバオプション \[424 ページ\]](#)

### 1.3.15.4.1 各種マルチプログラミングレベル設定の効果

マルチプログラミングレベルの設定はパフォーマンスに影響を与えます。

マルチプログラミングレベルの適切な値を決定するのは、難しいことがあります。たとえば、データベースアプリケーションが Java ストアドプロシージャを利用する場合や、クエリ内並列処理が有効になっている場合、これらの要求を処理するために追加で作成されたサーバタスクは、現在のマルチプログラミングレベルを超えると、他の要求が完了するまで実行を待機することになります。このような状況では、マルチプログラミングレベルを上げることが適切です。ネットワークデータベースサーバのマルチプログラミングレベルは、いつでも調整できます。ただし、パーソナルデータベースサーバについては、サーバの起動後にマルチプログラミングレベルを変更できません。

## マルチプログラミングレベルの増加

多くの場合、マルチプログラミングレベルを上げると、データベースサーバの全体的なスループットも相応して向上します。これは、マルチプログラミングレベルを上げることによって同時に実行できるタスクの数が増えるためです。ただし、マルチプログラミングレベルを上げるときに検討が必要なトレードオフがあります。これには次のようなものがあります。

### 競合の増加

同時に実行するタスクの数を増加すると、アクティブな要求間で競合が発生する可能性が高まる場合があります。競合はリソースに関連するもので、スキーマロックやローロック、データベースサーバ内部のデータ構造や同期プリミティブなどが挙げられます。このような状況では、サーバのスループットが実際に低下することがあります。

#### サーバのオーバヘッドの増加

アクティブな各タスクは、ワーカの割り付けと管理を要求します。スケジュールを制御するために追加の管理構造も必要です。さらに、各ワーカは、実行スタックに対するアドレス領域の事前割り付けを要求します。スタックサイズはプラットフォームによって異なります。Windows オペレーティングシステムでは、スタック領域の割り付けはデータベースサーバプロセスのアドレス領域に影響を及ぼします。ただし、スタックメモリは要求に応じて割り付けられます。UNIX プラットフォームの場合 (Linux を含みます)、スタックの退避メモリが即座に割り付けられます。そのため、マルチプログラミングレベルを上げると、サーバのメモリフットプリントが増加します。また、利用できるアドレス領域が少なくなるため、キャッシュマネージャで利用できるメモリ量が減少します。

#### スラッシング

データベースサーバは、個々の要求に対する処理を実行するためではなく、実行のオーバヘッドを管理するためだけに、多量のリソースを使用する状態に陥ることがあります。この状態は一般的に「スラッシング」と呼ばれています。スラッシングが発生するのは、データベースサーバのキャッシュ内の領域に対して競合するアクティブなタスクの数があまりに多く、これらのタスクのセットが使用するデータベースページのワーキングセットを保てるほどキャッシュが大きくない場合などです。結果として、オペレーティングシステムで発生する場合と同様に、ページの横取りが発生することがあります。

#### クエリ処理への影響

データベースサーバは、同時に処理可能なメモリを大量に消費する要求の最大数を選択します。データベースサーバのマルチプログラミングレベルを上げても、メモリが利用可能になるまで、要求が待機状態になる場合があります。

#### データ構造のメモリ

データベースサーバは、リソースを使用して文を解析したり、最適化したりします。複雑な文やキャッシュサイズが小さい場合、サーバデータ構造に使用されるメモリのサイズが利用可能な上限を超えてしまう場合があります。各タスクのサーバデータ構造に使用するメモリ量は、メモリガバナナーによって制限されます。各タスクには、次の近似的な制限があります。

$$(3/4 \text{ maximum-cache-size}) / (\text{number-of-active-requests})$$

制限を超えると、文にはエラーが発生します。

## マルチプログラミングレベルの低下

同時に実行するタスクの数を減らしてデータベースサーバのマルチプログラミングレベルを下げると、一般に、データベースサーバのスループットは低下します。ただし、マルチプログラミングレベルを下げることで、個々の要求の応答時間は短縮されることがあります。これは、リソースに対して競合する要求の数が少なくなり、ロック競合の確率が低下するためです。

マルチプログラミングレベルを下げすぎると、スレッドデッドロックが発生することがあります。マルチプログラミングレベルが負荷に対して適切なレベルになっていて、スレッドデッドロックが発生する場合は、アプリケーション設計に問題があることが考えられます。結果として競合が発生し、延いてはスケラビリティが低下します。例として、データベースに新しいデータを挿入するときにすべてのアプリケーションが変更を加える必要のあるテーブルについて考えてみます。この方法は、プライマリキーを生成するスキームの一部として、よく使用されます。このようなテーブルでは、結果として、アプリケーションの挿入トランザクションのすべてが事実上直列化される状況になります。共有テーブルが直列化されることに起因して、サーバがサービスを提供する速度よりも挿入トランザクション速度の方が速くなると、通常、スレッドデッドロックが発生します。

-gnl データベースサーバオプションは、最小マルチプログラミングレベルを設定します。

## 関連情報

[キャッシュとメモリガバナ \[1371 ページ\]](#)

[Windows パフォーマンスモニタ \[1434 ページ\]](#)

[接続、データベース、データベースサーバのプロパティ \[838 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gna データベースサーバオプション \[422 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gni データベースサーバオプション \[424 ページ\]](#)

[-gns データベースサーバオプション \[425 ページ\]](#)

[パフォーマンス統計値ユーティリティ \(dbstats\) \(UNIX\) \[1138 ページ\]](#)

### 1.3.15.4.2 マルチプログラミングレベルの手動調整

ネットワークサーバのマルチプログラミングレベルを手動で調整できます。

ただし、マルチプログラミングレベルの自動チューニングが有効になっている場合は、手動設定後にサーバによって自動的にマルチプログラミングレベルが再調整されることがあります。マルチプログラミングレベルを下げた場合、サーバは新しいマルチプログラミングレベルが有効になるまでアクティブなタスクの完了を待機する必要がある場合があるため、新しい設定は直ちに有効にならないことがあります。マルチプログラミングレベルを、MinMultiProgrammingLevel 設定と MaxMultiProgrammingLevel 設定によってそれぞれ指定された下限値と上限値を超える値に設定しようとすると、エラーが返されます。

マルチプログラミングレベルの自動チューニングは、データベースサーバの起動時またはデータベースサーバの実行中に無効にすることができます。

#### サーバ起動時の自動マルチプログラミングチューニングの無効化

データベースサーバを起動するときに、-gna 0 を指定して、マルチプログラミングレベルの自動チューニングを無効にします。

```
dbsrv17 -gna 0 ...
```

デフォルトのマルチプログラミングレベルを変更するには -gn オプションを使用し、最小値と最大値を設定するにはそれぞれ -gni オプションと -gnh オプションを使用します。

#### 実行中サーバの自動マルチプログラミングチューニングの無効化

データベースサーバがすでに実行されている場合は、次の SQL 文を実行して、マルチプログラミングレベルの自動チューニングを無効にします。

```
CALL sa_server_option ( 'AutoMultiProgrammingLevel', 'NO' );
```

最小値と最大値を設定するには、それぞれ MinMultiProgrammingLevel プロパティと MaxMultiProgrammingLevel プロパティを使用します。次に例を示します。

```
CALL sa_server_option ( 'MinMultiProgrammingLevel', 10 );  
CALL sa_server_option ( 'MaxMultiProgrammingLevel', 100 );
```

現在のマルチプログラミングレベルを設定するには、次のように CurrentMultiProgrammingLevel プロパティを使用します。

```
CALL sa_server_option ( 'CurrentMultiProgrammingLevel', 25 );
```

## 関連情報

[-gna データベースサーバオプション \[422 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gni データベースサーバオプション \[424 ページ\]](#)

## 1.3.16 サービスまたはデーモンとしてデータベースサーバを実行する方法

データベースサーバをサービスまたはデーモンとして実行するには、いくつかの方法があります。

ユーザ ID とパスワードを使用してコンピュータにログインすると、ユーザはセッションを確立します。データベースサーバ、またはその他のアプリケーションを起動すると、そのセッション内で稼働します。コンピュータからログオフすると、そのセッションに関連するすべてのアプリケーションは停止します。

データベースサーバを常時利用可能にする必要がある場合は、コンピュータからログオフしてもデータベースサーバが稼働し続けるように、Windows と UNIX 用の SQL Anywhere を実行できます。

### Windows サービス

Windows データベースサーバをサービスとして実行できます。この設定は、高可用性サーバを実行する場合に便利です。

### UNIX デーモン

-ud オプションを使用すると、UNIX データベースサーバをデーモンとして実行できます。これによって、データベースサーバをバックグラウンドで実行し、ログオフ後も引き続き実行させることができます。

### Linux サービス

Linux データベースサーバをサービスとして実行できます。この設定には、高可用性サーバを実行するための便利な特性が多数あります。

dbsvc ユーティリティを実行すると、サービスとして実行できるプログラムをすべて一覧表示できます。

このセクションの内容:

### [Windows サービス \[340 ページ\]](#)

データベースサーバは、サービスではなく Microsoft Windows プログラムのように実行できます。

### [Windows サービスの作成 \(SQL Central の場合\) \[341 ページ\]](#)

SQL Central を使用して、Windows サービスを作成します。

### [Windows サービスの作成 \(dbsvc ユーティリティの場合\) \[342 ページ\]](#)

dbsvc ユーティリティで Windows サービスを作成します。

#### [Windows サービスの削除 \(SQL Central の場合\) \[343 ページ\]](#)

SQL Central を使用して Windows サービスを削除します。削除すると、サービスのリストからサービス名が削除されます。

#### [Windows サービスの削除 \(dbsvc ユーティリティの場合\) \[344 ページ\]](#)

dbsvc ユーティリティを使用して Windows サービスを削除します。削除すると、サービスのリストからサービス名が削除されます。

#### [Windows サービスの設定 \[345 ページ\]](#)

Windows サービスを設定します。

#### [再表示頻度の設定 \[349 ページ\]](#)

SQL Central の再表示頻度を設定します。

#### [Windows サービスの開始または停止 \(SQL Central の場合\) \[349 ページ\]](#)

SQL Central で Windows サービスを開始または停止します。

#### [Windows サービスマネージャ \[350 ページ\]](#)

Windows サービスマネージャは、コンピュータ上で実行されるタスクを管理します。

#### [システムグループの管理 \[350 ページ\]](#)

SQL Central でサービスが割り当てられているグループを調べて変更します。

#### [UNIX でのデーモンとしてのデータベースサーバ \[353 ページ\]](#)

UNIX データベースサーバをバックグラウンドで実行し、現在のセッションから独立して稼働させるには、データベースサーバをデーモンとして実行します。

## 関連情報

[Linux 用サービスユーティリティ \(dbsvc\) \[1152 ページ\]](#)

[Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

## 1.3.16.1 Windows サービス

データベースサーバは、サービスではなく Microsoft Windows プログラムのように実行できます。

ただし、標準のプログラムとして、また、特にマルチユーザ環境で実行する場合には制限があります。

### 標準実行プログラムとして実行する場合の制限事項

プログラムを起動すると、プログラムは Windows のログインセッションで実行されます。これは、コンピュータからログオフするとプログラムが停止されることを意味します。データベースサーバで通常行われるように、プログラムを常に動作させておくために、この設定によってコンピュータの用途が制限されます。データベースサーバを実行し続けるには、そのデータベースサーバを実行するコンピュータにログオンし続けます。この設定では、Windows コンピュータをログオンした状態にしておくことになるので、セキュリティ上の問題も発生します。

## サービスの利点

アプリケーションを Windows サービスとしてインストールすると、ログオフしても実行できます。

サービスを開始するときに、サービスは LocalSystem という特別なシステムアカウント (またはユーザが指定した別のアカウント) を使ってログオンします。サービスを起動するユーザの ID とサービスは関連していないので、起動した人がログオフしてもサービスは開いたままになります。Windows コンピュータが起動してユーザがログオンする前に、サービスを自動的に開始するように設定することもできます。

## サービスの管理

SQL Central は、Windows のサービス マネージャよりも便利でわかりやすい方法で SQL Anywhere サービスを管理します。dbsvc ユーティリティを使用してサービスの作成や変更を行うこともできます。

## 関連情報

[Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

## 1.3.16.2 Windows サービスの作成 (SQL Central の場合)

SQL Central を使用して、Windows サービスを作成します。

## コンテキスト

データベースサーバごとに複数のデータベースを実行できます。複数のデータベースを同時に実行する場合は、新しいサービスを作成するのではなく、既存の Windows サービスに新しいデータベースを追加することをおすすめします。

## 手順

1. SQL Central でデータベースに接続します。
2. 左ウィンドウ枠で、*SQL Anywhere17* をクリックします。
3. 右ウィンドウ枠で、*サービスタブ* をクリックします。
4. **ファイル** > **新規** > **サービス** をクリックします。
5. **サービス作成ウィザード** の指示に従います。
  - サービス名の最初の 8 文字は、ユニークにしてください。

- SQL Central でサービスを作成する場合は、実行プログラム自体の名前ではなく、実行プログラムのオプションをウィンドウで入力します。たとえば、20 MB のキャッシュサイズで myserver という名前のサンプルデータベースを指定して、ネットワークサーバを実行する場合は、SQL Central のサービス作成ウィザードのパラメータテキストボックスに次のように入力します。

```
-c 20M  
-n myserver "%SQLANYSAMPl7%¥demo.db"
```

改行は任意です。

- サービスを実行するアカウントを選択します (LocalSystem という特別なアカウントまたは別のユーザ ID)。
- サービスを自動で開始するよう選択すると、コンピュータが Windows を起動するときに必ずサービスが開始されます。手動で開始することを選択した場合は、毎回 SQL Central から開始する必要があります。今後使用するためにサービスを設定する場合は、無効をクリックします。

## 結果

新しい Windows サービスが作成されます。

## 関連情報

[サービスアカウントのオプション \[347 ページ\]](#)

[Windows サービスの設定 \[345 ページ\]](#)

[Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

### 1.3.16.3 Windows サービスの作成 (dbsvc ユーティリティの場合)

dbsvc ユーティリティで Windows サービスを作成します。

## コンテキスト

データベースサーバごとに複数のデータベースを実行できます。複数のデータベースを同時に実行する場合は、新しいサービスを作成するのではなく、既存の Windows サービスに新しいデータベースを追加することをお奨めします。

## 手順

-w オプションを指定してサービスユーティリティ (dbsvc) を実行します。

## 結果

新しい Windows サービスが作成されます。

### 例

次のコマンドは、myserv という Windows サービスを作成します。このサービスでは、データベースサーバが LocalSystem ユーザとして動作します。

```
dbsvc -as -w myserv "C:¥Program Files¥SQL Anywhere 17¥Bin32¥dbsrv17.exe"  
-n william -c 8m "c:¥temp¥sample.db"
```

## 関連情報

[Windows サービスの設定 \[345 ページ\]](#)

[Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

## 1.3.16.4 Windows サービスの削除 (SQL Central の場合)

SQL Central を使用して Windows サービスを削除します。削除すると、サービスのリストからサービス名が削除されます。

## 前提条件

サービスを停止する必要があります。

## コンテキスト

Windows サービスを削除しても、ハードディスクからソフトウェアが削除されることはありません。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[SQL Anywhere17](#) をクリックします。
3. 右ウィンドウ枠で、[サービスタブ](#) をクリックします。
4. 削除するサービスを右クリックし、[削除](#) をクリックします。

## 結果

Windows サービスが削除されます。

## 関連情報

[Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

### 1.3.16.5 Windows サービスの削除 (dbsvc ユーティリティの場合)

dbsvc ユーティリティを使用して Windows サービスを削除します。削除すると、サービスのリストからサービス名が削除されます。

## 前提条件

サービスを停止する必要があります。

## コンテキスト

Windows サービスを削除しても、ハードディスクからソフトウェアが削除されることはありません。

## 手順

-d オプションを指定してサービスユーティリティ (dbsvc) を実行します。

## 結果

Windows サービスが削除されます。

### 例

myserv というサービスを、確認プロンプトを表示せずに削除するには、次のコマンドを入力します。

```
dbsvc -y -d myserv
```

## 関連情報

[Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

### 1.3.16.6 Windows サービスの設定

Windows サービスを設定します。

#### コンテキスト

Windows サービスは、一連のオプションやプロパティを使って、データベースサーバまたは他のアプリケーションを実行します。このプロパティには、サービスが動作するアカウントや、サービスが開始される条件を指定するものも含まれます。

サービス設定の変更は、次のサービス実行時から有効となります。

#### 手順

1. SQL Central で、[SQL Anywhere17 プラグイン](#)を使用してデータベースに接続します。
2. 左ウィンドウ枠で、[SQL Anywhere17](#) をクリックします。
3. 右ウィンドウ枠で、[サービスタブ](#)をクリックします。
4. 設定するサービスを右クリックし、[プロパティ](#)をクリックします。
5. [サービスのプロパティ](#)ウィンドウのタブで、必要に応じてプロパティを変更します。
6. 変更が完了したら、[OK](#) をクリックします。

#### 結果

サービスのプロパティが変更されます。[起動](#)プロパティは、次回 Windows を起動するときに適用されます。

このセクションの内容:

[Windows サービス起動オプション \[346 ページ\]](#)

Windows サービス起動オプションは、SQL Anywhere サービスの起動時の動作を制御します。

[Windows サービスのオプション \[346 ページ\]](#)

サービスに対する Windows サービスオプションは、実行プログラムのオプションと同じです。

[サービスアカウントのオプション \[347 ページ\]](#)

サービスが実行されるアカウントを選択できます。

[サービスの実行ファイルの変更 \[347 ページ\]](#)

サービスに関連付けられたプログラム実行ファイルを変更します。

[Windows サービスへのデータベースの追加 \[348 ページ\]](#)

Windows サービスにデータベースを追加します。

### 1.3.16.6.1 Windows サービス起動オプション

Windows サービス起動オプションは、SQL Anywhere サービスの起動時の動作を制御します。

これらのオプションは、[サービスのプロパティウィンドウの一般タブ](#)で設定できます。

#### 自動

サービスは、Windows オペレーティングシステムが起動すると必ず起動します。この設定は、データベースサーバやその他の常に稼働しているアプリケーションに適しています。

#### 手動

Windows では、管理者でないとサービスを起動できません。

#### 無効

サービスは起動しません。

### 1.3.16.6.2 Windows サービスのオプション

サービスに対する Windows サービスオプションは、実行プログラムのオプションと同じです。

[サービスのプロパティウィンドウの設定タブ](#)には、サービスのオプションを指定する[パラメータテキストボックス](#)があります。このボックスには、実行プログラムの名前は入力しないでください。

#### 例

2つのデータベースを実行する my\_server というネットワークサーバサービスを、キャッシュサイズ 20 MB で起動するには、[パラメータフィールド](#)に次のように入力します。

```
-c 20M
-n my_server
c:¥db_1.db
c:¥db_2.db
```

サンプルデータベースにユーザ ID DBA で接続する SQL Remote Message Agent サービスを起動するには、次のように入力します。

```
-c "UID=DBA;PWD=sql;DBN=demo"
```

### 1.3.16.6.3 サービスアカウントのオプション

サービスが実行されるアカウントを選択できます。

デフォルトでは、ほとんどのサービスは LocalSystem という特別なアカウントの下で実行されます。サービスのプロパティウィンドウの **アカウント** タブを開き、アカウント情報を入力すると、別のアカウントでログオンするようにサービスを設定できます。

LocalSystem 以外のアカウントでサービスを実行するには、そのアカウントに **サービスとしてログオン** する権限が必要です。この権限は、Windows のユーザマネージャの [高度なユーザ権利] で付与できます。この権限を付与するためにサービスユーザーティリティ (dbsvc) を使用することもできます。

#### システムトレイにアイコンが表示される場合

- サービスが LocalSystem で実行されているときに、サービスのプロパティウィンドウの **デスクトップとの対話をサービスに許可** がオンになっている場合は、サービスを実行するコンピュータ上の Windows にどのユーザがログインしてもデスクトップにアイコンが表示されます。すべてのユーザがアプリケーションウィンドウを開き、サービスとして実行されているプログラムを停止できます。
- サービスが LocalSystem で実行されているときに、サービスのプロパティウィンドウの **デスクトップとの対話をサービスに許可** がオフになっている場合は、ユーザのデスクトップにアイコンは表示されません。サービスの状態を変更する権限を与えられているユーザのみ、サービスを停止できます。
- サービスが他のアカウントで実行されている場合、デスクトップにアイコンは表示されません。サービスの状態を変更する権限を与えられているユーザのみ、サービスを停止できます。

#### i 注記

Windows 7 および Windows Server 2008 以降は、サービスが直接ユーザと対話することはできません。 **デスクトップとの対話をサービスに許可** オプションでもこれを変更することはできません。

### 1.3.16.6.4 サービスの実行ファイルの変更

サービスに関連付けられたプログラム実行ファイルを変更します。

#### 手順

1. SQL Central で、**SQL Anywhere17** プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、**SQL Anywhere17** をクリックします。
3. 右ウィンドウ枠で、**サービス** タブをクリックします。
4. 設定するサービスを右クリックし、**プロパティ** をクリックします。
5. **設定** タブをクリックします。
6. **ファイル名** テキストボックスに、新しいパスとファイル名を入力します。

---

実行ファイルを新しいディレクトリに移動する場合は、この内容も変更します。

## 結果

プログラム実行ファイルが更新されます。

### 1.3.16.6.5 Windows サービスへのデータベースの追加

Windows サービスにデータベースを追加します。

## コンテキスト

データベースサーバごとに複数のデータベースを実行できます。複数のデータベースを同時に実行する場合は、新しいサービスを作成するのではなく、既存の Windows サービスにデータベースを追加することをお奨めします。

稼働中のサーバ上でデータベースを起動するには、Interactive SQL などのクライアントアプリケーションを使います。

アプリケーションからデータベースを起動しても、データベースはサービスに付加されません。サービスを停止して再起動すると、追加のデータベースは自動的に起動しません。

## 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、*SQL Anywhere17* をクリックします。
3. 右ウィンドウ枠で、**サービスタブ**をクリックします。
4. サービスを右クリックし、**プロパティ**をクリックします。
5. **設定タブ**をクリックします。
6. 新しいデータベースのパスとファイル名を、**パラメータボックス**のオプションリストの最後に追加します。
7. **OK** をクリックして変更を保存します。

## 結果

次回サービスを開始したときに、データベースが起動します。

## 1.3.16.7 再表示頻度の設定

SQL Central の再表示頻度を設定します。

### コンテキスト

デフォルトでは、SQL Central は 10 秒ごとに情報を再表示します。再表示によって、各 Windows サービスのステータス (開始または停止) がチェックされ、**ステータスカラム**が更新されて現在のステータスが表示されます。さらに、SQL Central は接続やロックなどの動的オブジェクト、イベントやメンテナンスプランの動的プロパティを再表示します。

### 手順

1. SQL Central で、**ツール** > **SQL Anywhere17** > **ユーザ設定** をクリックします。
2. **自動的に再表示タブ**をクリックします。
3. **自動再表示を有効にする**をクリックします。
4. **再表示間隔** x 秒に時間間隔を指定します。
5. **適用**をクリックします。
6. **OK** をクリックします。

### 結果

再表示の頻度が変更されます。

## 1.3.16.8 Windows サービスの開始または停止 (SQL Central の場合)

SQL Central で Windows サービスを開始または停止します。

### コンテキスト

サービスを開始すると、停止するまで実行を続けます。SQL Central を閉じたり、ログオフしてもサービスは停止しません。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[SQL Anywhere17](#) をクリックします。
3. 右ウィンドウ枠で、[サービスタブ](#)をクリックします。
4. サービスを右クリックし、[起動](#)または[停止](#)をクリックします。

## 結果

Windows サービスが開始または停止します。データベースサーバのサービスを停止すると、データベースへの接続はすべて閉じられ、データベースサーバは停止されます。他のアプリケーションのプログラムは終了します。

### 1.3.16.9 Windows サービスマネージャ

Windows サービスマネージャは、コンピュータ上で実行されるタスクを管理します。

SQL Anywhere のすべてのサービス管理は、SQL Central から行うことができます。Windows の[コントロールパネル](#)にある[サービスマネージャ](#)を使用していくつかのタスクを実行することはできますが、Windows の[サービスマネージャ](#)で SQL Anywhere サービスをインストールしたり設定したりすることはできません。

Windows の[コントロールパネル](#)から[サービスマネージャ](#)を開くと、サービスのリストが表示されます。SQL Anywhere サービスの名前は、サービスをインストールしたときに入力したサービス名の前に "SQL Anywhere" が付いた形式になっています。インストールされたすべてのサービスが、リストに表示されます。

### 1.3.16.10 システムグループの管理

SQL Central でサービスが割り当てられているグループを調べて変更します。

## 前提条件

サービスが、適切なグループのメンバーであることを確認してから、サービスが正しい順序で開始されるように設定する必要があります。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。

2. 左ウィンドウ枠で、[SQL Anywhere17](#) をクリックします。
3. 右ウィンドウ枠で、[サービスタブ](#) をクリックします。
4. サービスを右クリックし、[プロパティ](#) をクリックします。
5. [依存性](#) タブをクリックします。最上部のテキストボックスにサービスが割り当てられているグループの名前が表示されます。
6. [変更](#) をクリックして、システムで使用可能なグループのリストを表示します。
7. いずれかのグループを選択するか、新しいグループの名前を入力します。
8. [OK](#) をクリックして、そのグループにサービスを割り当てます。

## 結果

グループサービスがチェックされ、変更されます。

このセクションの内容:

### [サービス依存性の管理 \[351 ページ\]](#)

SQL Central を使用して、サービスまたはグループを依存性のリストに追加します。

### [サービスグループのリスト \[352 ページ\]](#)

サービスまたはデーモンが属することができるサービスグループをリストします。

## 関連情報

### [サービスグループのリスト \[352 ページ\]](#)

## 1.3.16.10.1 サービス依存性の管理

SQL Central を使用して、サービスまたはグループを依存性のリストに追加します。

## コンテキスト

状況によっては、複数の実行プログラムをサービスとして実行する必要があり、これらの実行プログラムが相互に依存する場合があります。たとえば、レプリケーションを補助するために、サーバと SQL Remote Message Agent を実行する場合があります。サービスは正しい順序で開始する必要があります。サーバが起動する前に SQL Remote Message Agent サービスが開始されると、サーバを検出できないためエラーになります。「サービスグループ」を使用すると、これらの問題を防止できます。サービスグループは SQL Central で管理します。SQL Central を使用すると、サービスの依存性を指定できます。次に例を示します。

- サービスグループのリストにある各グループの少なくとも1つのメンバーを開始してから現在のサービスを開始するよう  
にできます。
- 任意の数のサービスを開始してから現在のサービスを開始するようことができます。たとえば、特定のネットワークサーバを  
起動してから、そのサーバに対して実行する SQL Remote Message Agent を開始するようになりたい場合などです。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[SQL Anywhere17](#) をクリックします。
3. 右ウィンドウ枠で、[サービスタブ](#)をクリックします。
4. サービスを右クリックし、[プロパティ](#)をクリックします。
5. [依存性タブ](#)をクリックします。
6. [サービスの追加](#)または[サービスグループの追加](#)をクリックして、サービスまたはグループを依存性リストに追加します。
7. リストからサービスまたはグループを1つ選択します。
8. [OK](#)をクリックして、そのサービスまたはグループを依存性リストに追加します。

## 結果

サービスまたはグループが依存性のリストに追加されます。

### 1.3.16.10.2 サービスグループのリスト

サービスまたはデーモンが属することができるサービスグループをリストします。

デフォルトでは、次の表に示すように各サービスがグループに属しています。

サービス	デフォルトのグループ
ネットワークサーバ	SQLANYServer
パーソナルサーバ	SQLANYEngine
Mobile Link 同期クライアント	SQLANYMLSync
SQL Remote Message Agent	SQLANYRemote
Mobile Link サーバ	SQLANYMobiLink
SQL Anywhere Broadcast Repeater ユーティリティ	SQLANYNS
Mobile Link Listener	SQLANYLSN
SQL Anywhere ボリュームシャドウコピーサービス	SQLANYVSS
Mobile Link Relay Server	SQLANYRSHOST (Linux のみ)

サービス	デフォルトのグループ
Mobile Link Relay Server Outbound Enabler	SQLANYRSOE
Mobile Link エージェント	SQLANYMLAGENT

### 1.3.16.11 UNIX でのデーモンとしてのデータベースサーバ

UNIX データベースサーバをバックグラウンドで実行し、現在のセッションから独立して稼働させるには、データベースサーバをデーモンとして実行します。

#### i 注記

データベースサーバをバックグラウンドで実行するのに '&' を使用しないでください。データベースサーバをバックグラウンドで実行するのに UNIX の & (アンパサンド) コマンドを使用しても機能しません。データベースサーバがただちに停止するか、あるいは応答しなくなります。データベースサーバはデーモンとして実行してください。

同様に、一般的な `fork() -exec()` シーケンスを使用してプログラムの中からサーバをバックグラウンドで起動しようとしても機能しません。これを行うには、データベースサーバオプションのリストに `-ud` オプションを追加します。

UNIX データベースサーバは、次のいずれかの方法でデーモンとして実行できます。

1. データベースサーバの起動時に、`-ud` オプションを使用します。次に例を示します。

```
dbsrv17 -ud demo
```

2. `dbspawn` ツールを使用してデータベースサーバを起動します。次に例を示します。

```
dbspawn dbsrv17 demo
```

`dbspawn` を使用することの利点は、デーモンが起動し、要求を受け入れる状態になったことを確認するまで `dbspawn` プロセスが停止しないことです。何らかの理由でデーモンの起動が失敗した場合、`dbspawn` の終了コードは 0 以外の値になります。

`-ud` オプションを使用してデーモンを直接起動したときは、データベースサーバがデーモンプロセスを作成し、(終了して次のコマンドを実行できるように) すぐに返します。その後、デーモンがそれ自体を初期化するか、コマンドで指定されたデータベースを開こうとします。

データベースサーバを使用できる 1 つ以上のアプリケーションでデーモンが確実に実行されるようにするには、アプリケーションの起動前に、デーモンを実行する `dbspawn` を使用します。次の例では、`csh` スクリプトを使用してこれをテストする方法を示します。

```
#!/bin/csh
# start the server as a daemon and ensure that it is
# running before you start any applications
dbspawn dbsrv17 demo
if ( $status != 0 ) then
    echo Failed to start demo server
    exit
endif
# ok, now you can start the applications
...
```

次の例では、sh スクリプトを使用して、アプリケーションの起動前にデーモンが実行中であるかどうかをテストします。

```
#!/bin/sh
# start the server as a daemon and ensure that it is
# running before you start any applications
dbspawn dbsrv17 demo
if [ $? != 0 ]; then
    echo Failed to start demo server
    exit
fi
# ok, now you can start the applications
...
```

3. C プログラムの中からデーモンを生成します。-ud オプションを使用してください。次に例を示します。

```
...
if( fork() == 0 ) {
    /* child process = start server daemon */
    execl( "/opt/sqlanywhere17/bin/dbsrv17",
"dbsrv17", "-ud", "demo" );
    exit(1);
}
/* parent process */
...
```

## 関連情報

[-ud データベースサーバオプション \[490 ページ\]](#)

[サーババックグラウンド起動ユーティリティ \(dbspawn\) \[1161 ページ\]](#)

## 1.3.17 OEM Edition の認証アプリケーション

SQL Anywhere OEM Edition はデータベースのあらゆる操作を実行できます。

SQL Anywhere の OEM Edition は、SAP OEM パートナに提供されています。SQL Anywhere OEM Edition を使用すると、認証アプリケーションは、ユーザ ID に付与された権限に基づいて、データベース上で任意の操作を実行できます。

非認証で接続すると、読み込み専用アクセスとなり、テンポラリテーブルに対する挿入、更新、削除を実行できます。非認証接続を使用すると、ストアードプロシージャを使用して複雑なレポートを作成したり、Crystal Reports などのレポートングツールを使用してアクセスしたりすることができます。

認証メカニズムは、他のすべてのアプリケーションプログラミング言語やツールとは無関係であり、接続ごとに独立して実行されます。したがって、アプリケーション内で、認証接続と、制限付きの非認証接続の両方を使用できます。

認証は、セキュリティメカニズムではありません。データベースに対して非認証データベースサーバを実行するユーザは、標準的な SQL 権限スキームに基づいて任意の操作を実行できます。

このセクションの内容:

[認証アプリケーションの開発 \[355 ページ\]](#)

認証アプリケーションの開発は、単純なプロセスです。特殊な認証シグネチャがデータベースに組み込まれ、もう1つのシグネチャがアプリケーションに組み込まれます。

## 1.3.17.1 認証アプリケーションの開発

認証アプリケーションの開発は、単純なプロセスです。特殊な認証シグネチャがデータベースに組み込まれ、もう1つのシグネチャがアプリケーションに組み込まれます。

アプリケーションがデータベースに接続すると、シグネチャが比較され、アプリケーションの認証が行われます。SQL Anywhere の認証アプリケーションを開発するには、以下の手順が必要です。

1. 認証シグネチャの取得
2. データベースの認証
3. アプリケーションの認証

SQL Anywhere に付属しているデータベースツール (SQL Central、Interactive SQL、および dbbackup などのユーティリティ) は、自己認証形式です。これらのツールは、認証データベースへの操作が制限されません。データベース自体が認証されていない場合は、ツールは、制限された読み込み専用モードで動作します。

認証アプリケーションでは、OEM Edition の SQL Anywhere データベースサーバを使用する必要があります。このエディションで、通常のデータベースサーバと唯一異なる点は、認証命令を処理することです。認証命令は、その他のエディションのデータベースサーバでは無視されます。認証データベースサーバを使用しない場合は、非認証アプリケーションにも制限は適用されません。

このセクションの内容:

### [認証シグネチャの取得 \[356 ページ\]](#)

データベースとアプリケーションの認証シグネチャを取得して、OEM Edition を使用します。

### [データベースの認証 \(SQL\) \[357 ページ\]](#)

SET OPTION 文を使用してデータベースを認証します。

### [データベースの認証のチェック \[358 ページ\]](#)

DB\_PROPERTY 関数を使用して、データベースが認証されたことをチェックします。

### [アプリケーションの認証 \[359 ページ\]](#)

認証アプリケーションは、接続確立後すぐに、connection\_authentication データベースオプションを設定する必要があります。

### [認証文の実行 \[360 ページ\]](#)

使用するプログラミングインタフェースに応じて、認証オプションを設定する SET TEMPORARY OPTION 文を実行します。

### [認証データベースのアップグレード \[362 ページ\]](#)

データベースのアップグレードまたは再構築の前に %SQLANY17%/scripts ディレクトリに authenticate.sql ファイルを作成し、認証データベースをアップグレードします。

### [アプリケーションからの oem\\_string オプション設定の検出 \[363 ページ\]](#)

アプリケーションから oem\_string オプションを問い合わせます。

## 1.3.17.1.1 認証シグネチャの取得

データベースとアプリケーションの認証シグネチャを取得して、OEM Edition を使用します。

### 前提条件

認証シグネチャを取得するには、SAP との OEM 契約が必要です。

### 手順

1. [SAP SQL Anywhere 認証シグネチャの依頼](#) にアクセスします。
2. フォームに記入し、認証シグネチャを取得します。認証メカニズムに次の情報が組み込まれます。

会社名

会社の名前。

アプリケーション名

アプリケーションの名前。

### 結果

フォームを完成させると、48 時間以内に、データベースシグネチャとアプリケーションシグネチャについて知らせる電子メールが送信されます。これらのシグネチャは、英字と数字で構成される長い文字列です。

### 関連情報

[データベースの認証 \(SQL\) \[357 ページ\]](#)

## 1.3.17.1.2 データベースの認証 (SQL)

SET OPTION 文を使用してデータベースを認証します。

### 前提条件

このオプションを設定するには SET ANY SECURITY OPTION システム権限が必要です。

### 手順

1. 次の SQL 認証文を使用して、データベースの database\_authentication オプションを設定します。

```
SET OPTION PUBLIC.database_authentication='company=company-name;  
application=application-name;  
signature=database-signature';
```

#### i 注記

構文例には、読みやすいように改行が追加されています。ただし構文を実行する場合には、改行なし、および等号とセミコロン間のスペースなしで実行してください。

company-name と application-name の引数は、シグネチャの取得時に SAP に提供した値です。database-signature は、SAP から受け取ったデータベースシグネチャです。

2. データベースを再起動して、設定を有効にします。

### 結果

データベースが認証されます。

### 次のステップ

SQL スクリプトファイルに認証文を格納し、長いシグネチャを繰り返し入力することを回避できます。▶ [ファイル](#) ▶ [スクリプトの実行](#) ▶ をクリックして、Interactive SQL から SQL スクリプトファイルを実行できます。

SQL Anywhere インストールディレクトリの scripts サブディレクトリに、authenticate.sql ファイルを作成し、このファイルに認証文を格納すると、この文は、データベースの作成、再構築、またはアップグレードを行うたびに適用されます。

## 関連情報

[認証データベースのアップグレード \[362 ページ\]](#)

[database\\_authentication オプション \[682 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

### 1.3.17.1.3 データベースの認証のチェック

DB\_PROPERTY 関数を使用して、データベースが認証されたことをチェックします。

## コンテキスト

データベースサーバが認証データベースをロードするとき、[データベースサーバメッセージウィンドウ](#)に、認証された会社とアプリケーションについて説明するメッセージが表示されます。メッセージの形式は次のとおりです。

```
このデータベースは次の使用を目的としてライセンスされています。  
アプリケーション: application-name  
会社: company-name
```

## 手順

1. データベースに接続します。
2. 次のように、DB\_PROPERTY 関数を実行します。

```
SELECT DB_PROPERTY ( 'Authenticated' );
```

## 結果

データベースが認証されたかどうかを示すメッセージが返されます。

## 関連情報

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[database\\_authentication オプション \[682 ページ\]](#)

## 1.3.17.1.4 アプリケーションの認証

認証アプリケーションは、接続確立後すぐに、`connection_authentication` データベースオプションを設定する必要があります。

このオプションは、接続が確立された直後にすべての接続に対して行われなければなりません。ODBC アプリケーションや JDBC アプリケーションは、データベースに対して機能について問い合わせます。ユーザは、これらのアクションについて制御する必要はありません。このため、すべての接続には、制限が適用される前に 30 秒間の猶予期間が設けられています。猶予期間があることにより、使用している開発ツールには関係なく、アプリケーションは認証を行うことができます。

Authenticated データベースプロパティを使用して、データベースが認証を認証を有効にしたかどうかを判別できます。

```
SELECT DB_PROPERTY( 'Authenticated' );
```

Authenticated 接続プロパティを使用して、現在の接続が認証されたかどうかを判別できます。

```
SELECT CONNECTION_PROPERTY ( 'Authenticated' );
```

次の SQL 文で接続が認証されます。

```
SET TEMPORARY OPTION connection_authentication='company=company-name;  
application=application-name;  
signature=application-signature';
```

### i 注記

構文例には、読みやすいように改行が追加されています。ただし構文を実行する場合には、改行なし、および等号とセミコロンの間のスペースなしで実行してください。

このオプションは、TEMPORARY キーワードを使用して、接続の間だけ設定できます。`company-name` と `application-name` は、`database_authentication` オプションにより指定された値に一致する必要があります。`application-signature` は、SAP から取得するシグネチャです。

データベースサーバは、データベースシグネチャに対してアプリケーションシグネチャを検証します。シグネチャが検証された場合、接続が認証され、ユーザの割り当て済みロールおよび権限によって強制される制限を超えて、接続の動作の制限がなくなります。シグネチャが検証されないと、接続は、非認証アプリケーションによって許可されているアクションに制限されます。

## 関連情報

[接続プロパティの一覧 \[839 ページ\]](#)

[connection\\_authentication オプション \[674 ページ\]](#)

### 1.3.17.1.5 認証文の実行

使用するプログラミングインタフェースに応じて、認証オプションを設定する SET TEMPORARY OPTION 文を実行します。

ここに示すシグネチャは、有効なシグネチャではありません。次のインタフェースを使用して認証オプションを設定することを前提に、例を示します。

- ODBC
- SAP PowerBuilder
- JDBC
- Microsoft ADO.NET
- Embedded SQL
- OLE DB

#### i 注記

会社名に引用符やアポストロフィなどの特殊文字が含まれている場合 (たとえば、Joe's Garage など)、認証文の構成には注意が必要です。認証オプションセット全体 (Company=...;Application=...;Signature=...) で SQL 文字列になります。SQL における文字列の規則では、文字列の中に引用符を含める場合は、その文字を 2 つ続けて記述することになっています。例:

```
SET TEMPORARY OPTION connection_authentication='Company = Joe''s Garage;  
Application=Joe''s Program;  
Signature=0fa55157edb8e14d818e...';
```

構文例には、読みやすいように改行が追加されています。ただし構文を実行する場合には、改行なし、および等号とセミコロンの間のスペースなしで実行してください。

## ODBC

次の文を使用します。

```
SQLExecDirect (  
    hstmt,  
    "SET TEMPORARY OPTION connection_authentication=  
    'Company=MyCo;  
    Application=MyApp;  
    Signature=0fa55159999e14d818e...';",  
    SQL_NTS  
);
```

文字列は、1 行に入力します。または、連結して構築します。

## SAP PowerBuilder

次の PowerScript 文を使用します。

```
EXECUTE IMMEDIATE
```

```
"SET TEMPORARY OPTION connection_authentication=
'Company=MyCo;
Application=MyApp;
Signature=0fa551599998e14d818e...';"
USING SQLCA
```

## JDBC

次の文を使用します。

```
Statement Stmt1 = con.createStatement();
Stmt1.executeUpdate(
"SET TEMPORARY OPTION connection_authentication=
'Company=MyCo;
Application=MyApp;
Signature=0fa551599998e14d818e...';"
);
```

文字列は、1行に入力します。または、連結して構築します。

## Microsoft ADO.NET

次の文を使用します。

```
SACommand cmd=new SACommand(
"SET TEMPORARY OPTION connection_authentication=
'Company=MyCo;
Application=MyApp;
Signature=0fa551599998e14d818e...';",
con
);
cmd.ExecuteNonQuery();
```

文字列は、1行に入力します。または、連結して構築します。

## Embedded SQL

```
Use the following statement:
EXEC SQL SET TEMPORARY OPTION connection_authentication=
'Company=MyCo;
Application=MyApp;
Signature=0fa551599998e14d818e...';
```

文字列は、1行に入力します。または、連結して構築します。

## OLE DB

認証データベースに接続するとき、接続と認証の手順は別々に行います。ただし、Microsoft Visual Basic Grid オブジェクトなどの一部のオブジェクトは、独立した暗黙的な接続の試行が可能であり、自動的に認証が行われません。このような場合は、接続は認証されず、データベース操作は失敗します。この問題を回避するには、接続文字列に InitString 接続パラメータを指定します。次の例は、すべての接続で接続確立直後に認証が行われるように、InitString 接続パラメータを含めるように Microsoft Visual Basic アプリケーションを修正する例を示したものです。

```
mConnectionString =
  "Provider=SAOLEDB.17;
  UID=DBA;
  PWD=passwd;
  ServerName=test17;
  InitString=SET TEMPORARY OPTION connection_authentication=
  'Company=MyCo;
  Application=MyApp;
  Signature=0fa55157edb8e14d818e...'"
mdbConn.ConnectionString = mConnectionString
mdbConn.Open
```

文字列は、1 行に入力します。または、連結して構築します。

### 1.3.17.1.6 認証データベースのアップグレード

データベースのアップグレードまたは再構築の前に `%SQLANY17%/scripts` ディレクトリに `authenticate.sql` ファイルを作成し、認証データベースをアップグレードします。

#### 前提条件

PUBLIC オプションを設定するには、SET ANY PUBLIC OPTION システム権限が必要です。

データベースファイルのバックアップコピーを作成することをお奨めします。

#### コンテキスト

このファイルは、CREATE DATABASE および ALTER DATABASE UPGRADE 文、dbupgrad および dbunload ユーティリティ、およびデータベースアップグレードウィザードによって実行されます。

データベースのアップグレードや再構築を行ったときに認証情報を保存するには、ファイル `authenticate.sql` に認証文を格納するしか方法がありません。

## 手順

1. `authenticate.sql` というファイルを `%SQLANY17%\scripts` ディレクトリに作成します。
2. 次の内容をファイルに追加します。

```
SET OPTION PUBLIC.database_authentication = 'authentication-statement'  
go
```

ファイルには、`go` が含まれている必要があります。含まれていないと、この文は無視されます。

## 結果

データベースがアップグレードされます。

## 次のステップ

データベースをアップグレードまたは再構築します。手順は、アップグレードまたは再構築するデータベースファイルのバージョンによって異なります。

## 関連情報

[database\\_authentication オプション \[682 ページ\]](#)

### 1.3.17.1.7 アプリケーションからの `oem_string` オプション設定の検出

アプリケーションから `oem_string` オプションを問い合わせます。

## コンテキスト

`oem_string` データベースオプションによって、ユーザ指定の情報がデータベースファイルのヘッダページに格納されます。

`oem_string` オプションは、認証アプリケーションに対して設定されます。

## 手順

1. SET OPTION 文を実行してデータベース OEM 文字列を設定し、次に設定を問い合わせます。

```
SET OPTION PUBLIC.oem_string='Jack's OEM Database';  
SELECT CONNECTION_PROPERTY( 'oem_string' );
```

2. データベースから切断し、データベースを停止します (たとえば、データベースサーバをシャットダウンするなどにより)。
3. 16 進エディタまたは一部のユーザ定義アプリケーションを使用して、データベースシステム DB 領域ファイル (.db ファイル) を開きます。
4. ファイルの最初のページをバッファに読み込みます。
5. バッファで、OEM 文字列の前後にある 2 バイトのプレフィクス (DB\_OEM\_STRING\_PREFIX) およびサフィックス (DB\_OEM\_STRING\_SUFFIX) シーケンスを検索します。次はその例です。

```
0220: 00 00 00 00 00 00 00 00 00 00 00 00 DA 7A BA 5E |.....z.^|  
0230: 00 45 4D 3D 4A 61 63 6B 27 73 20 4F 45 4D 20 44 |.EM=Jack's OEM D|  
0240: 61 74 61 62 61 73 65 00 00 00 00 00 00 00 00 |atabase.....|  
0250: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|  
0260: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|  
0270: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|  
0280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|  
0290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|  
02a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|  
02b0: 00 00 00 00 3D 4D 45 00 5E BA 7A DA 5E 77 CB A4 |....=ME.^z.^w..|
```

## 結果

プレフィクスとサフィックスの文字列は `sqldef.h` でそれぞれ、`DB_OEM_STRING_PREFIX` および `DB_OEM_STRING_SUFFIX` として定義されています。この 2 つの文字列の間にあるすべてのバイトによって、データベースで定義されている OEM 文字列が定義されます。

## 関連情報

[oem\\_string オプション \[744 ページ\]](#)

## 1.3.18 Windows での SQL Anywhere

SQL Anywhere ソフトウェアを Windows 上で実行する場合には、いくつかの考慮事項が適用されます。

### セキュリティ

これらのオペレーティングシステムには、ユーザアカウント制御 (UAC) という新しいセキュリティモデルが採用されています。UAC はデフォルトで有効に設定され、ファイルに書き込み可能とされるプログラムの動作に影響する可能性があります (特に、コンピュータが複数のユーザをサポートしている場合)。ファイルとディレクトリを作成した場所と作成方法によって、あるユーザが作成したファイルを、他のユーザが読み込んだり、書き込むことが許可されなくなる場合があります。

SQL Anywhere をデフォルトのディレクトリにインストールした場合は、複数のユーザに読み込み/書き込みアクセスを許可する必要があるファイルおよびディレクトリが適切に設定されます。

#### SQL Anywhere 昇格操作エージェント

UAC がアクティブな状態で実行する場合に、特定のアクションで権限の昇格が必要になります。SQL Anywhere では、次のプログラムで昇格が必要になることがあります。

- dbdsn.exe
- dbelevate17.exe
- dblic.exe
- dbsvc.exe
- installULNet.exe
- SetupVSPackage.exe
- ulcond17.exe

次の DLL では、登録または登録の解除時に昇格が必要です。

- dbctr17.dll
- dbodbc17.dll
- dboledb17.dll
- dboledba17.dll

UAC がアクティブな場合、SQL Anywhere の昇格操作エージェントに対して昇格を確認するメッセージが表示されることがあります。このメッセージは、識別されたプログラムの実行を継続するかどうかを確認したり (管理者としてログオンしている場合)、管理者のクレデンシャルを提供するように求める (管理者以外でログオンしている場合) ため、ユーザアカウント制御システムによって発行されるものです。

#### 配備に関する考慮事項

プログラム dbelevate17.exe は、権限の昇格を必要とする操作を実行するために SQL Anywhere コンポーネントによって内部的に使用されます。この実行プログラムを SQL Anywhere の配備に含める必要があります。

#### SQL Anywhere 実行プログラムの署名

SQL Anywhere 実行プログラムは、SAP により署名されています。

#### Windows サービス

Windows に対応しているサービスは、デスクトップと対話できません。これらのオペレーティングシステムでは、デスクトップと対話する SQL Anywhere サービスがありません。データベースサーバは、SQL Central または SQL Anywhere コックピットを使用してモニタリングできます。

## 1.4 SQL Anywhere データベースサーバ実行プログラム (dbsrv17, dbeng17)

データベースサーバを起動して実行します。

### 構文

```
dbsrv17 [ server-options ] [ database-file [ database-options ] ...]
```

```
dbeng17 [ server-options ] [ database-file [ database-options ] ...]
```

## サーバオプション

サーバオプション	説明
@data	設定ファイルまたは環境変数からオプションを読み込みます。
-?	使用方法を表示します。
-a/userid[:userid...]	指定したユーザに対して標準のユーザ認証を許可します。
-b	バルクオペレーションモードで実行します。
-csize	初期キャッシュサイズを設定します。
-ca 0	動的キャッシュサイズ決定を無効にします。
-cc {+ -}	キャッシュウォーミングに使用するデータベースページに関する情報を収集します。
-cdboption[:option...]	コックピットを起動します。
-chsize	キャッシュサイズの上限を設定します。
-chxsize	キャッシュ以外に使用するアドレス領域を確保します。
-clsize	キャッシュサイズの下限を設定します。
-cplocation[:location...]	クラスを検索する一連のディレクトリまたは JAR ファイルを指定します。
-cr {+ -}	データベースページを保持するキャッシュを準備します。
-cs	データベースサーバメッセージウィンドウにキャッシュの使用状況を表示します。
-cv {+ -}	データベースサーバメッセージウィンドウでのキャッシュウォーミングに関するメッセージの表示を制御します。
-dttemp-file-dir	テンポラリファイルを保存するディレクトリを指定します。
-eCencryption-options	パケット暗号化を有効にします。
-edi	データベース独立性をオンにします。
-ep	暗号化キーを入力するよう要求します。
-es	共有メモリを経由した暗号化されていない接続を許可します。

サーバオプション	説明
<code>-f</code>	トランザクションログなしでデータベースを強制的に起動します。
<code>-fcfilename</code>	ファイルシステムフルのコールバック関数を含む DLL のファイル名を指定します。
<code>-fips</code>	データベースおよび通信の暗号化に FIPS 認定のアルゴリズムの使用を要求します。
<code>-ga</code>	最後の非 HTTP クライアント接続を閉じた後、データベースを自動的にアンロードします。さらに、最後のデータベースを閉じた後で停止します。
<code>-gblevel</code>	データベースプロセスの優先度クラスを <code>level</code> に設定します。
<code>-gcnum</code>	最大チェックポイントタイムアウト時間を <code>num</code> 分に設定します。
<code>-gdlevel</code>	データベース起動権限を設定します。
<code>-gesize</code>	外部関数を実行するスレッドのスタックサイズを設定します。
<code>-gf</code>	トリガの起動を無効にします。
<code>-gklevel</code>	サーバの停止に必要な権限を設定します。
<code>-gllevel</code>	データのロードまたはアンロードに必要な権限を設定します。
<code>-gmnum</code>	接続の最大数を設定します。
<code>-gnnum</code>	データベースサーバのマルチプログラミングレベルを設定します。
<code>-gna</code>	データベースサーバのマルチプログラミングレベルの自動チューニングを制御します。
<code>-gnhnum</code>	データベースサーバが同時に実行できるタスクの最大数を設定します。
<code>-gnlnum</code>	データベースサーバが同時に実行できるタスクの最小数を設定します。
<code>-gns</code>	データベースサーバメッセージログにマルチプログラミングレベルの統計をレポートします。
<code>-gpsize</code>	最大ページサイズを <code>size</code> バイトに設定します。
<code>-grminutes</code>	リカバリの最大時間を設定します。
<code>-gsssize</code>	スレッドスタックサイズを <code>size</code> バイトに設定します。
<code>-gtnum</code>	使用できる物理プロセッサの最大数を設定します (ライセンスされたプロセッサの数を上限とする)。このオプションは、マルチプロセッサシステムでのみ役立ちます。

サーバオプション	説明
<code>-gta logical-processors-to-use,...</code>	データベースサーバが利用できる論理プロセッサを設定します。
<code>-gtclogical-processors-to-use</code>	データベースサーバで許可される、プロセッサの最大同時実行性を制御します。
<code>-gtp {0 1}</code>	トポロジを考慮したスケジューリングをオンまたはオフにします。
<code>-gu level</code>	ユーティリティコマンドの権限レベルを utility_db、all、none、または DBA に設定します。
<code>-im submode</code>	データベースサーバをインメモリで実行し、ディスクへの書き込みをなくしたり軽減したりします。
<code>-k</code>	パフォーマンスモニタ統計値の収集を制御します。
<code>-k/GSS-API-library-file</code>	Kerberos GSS-API ライブラリ (UNIX では共有オブジェクト) のファイル名を指定し、データベースサーバへの Kerberos 認証接続を有効にします。
<code>-kp server-principal</code>	Kerberos サーバプリンシパルを指定し、データベースサーバへの Kerberos 認証接続を有効にします。
<code>-krserver-realm</code> (廃止予定)	Kerberos サーバプリンシパルのレルム (管理領域) を指定し、データベースサーバへの Kerberos 認証による接続を有効にします。
<code>-krb</code>	データベースサーバへの Kerberos 認証接続を有効にします。
<code>-ks</code>	パフォーマンスモニタで、データベースサーバからカウンタ値を収集するために使用される共有メモリの作成を無効にします。
<code>-ksc</code>	パフォーマンスモニタでモニタできる接続の最大数を指定します。
<code>-ksd</code>	パフォーマンスモニタでモニタできるデータベースの最大数を指定します。
<code>-m</code>	すべてのデータベースについて、各チェックポイントの実行後にトランザクションログをトランケートします。
<code>-nname</code>	データベースサーバ名を <code>name</code> にします。-n オプションは指定する位置によって意味が異なります。
<code>-ncs</code>	NCS (Native Component Supportability) ライブラリを使用した、SAP Solution Manager へのステータスイベント送信を有効にします。
<code>-ncsd filename</code>	NCS (Native Component Supportability) ライブラリを使用した、SAP Solution Manager へのステータスイベント送信を有効にし、使用する NCS 設定ファイルの場所を指定します。

サーバオプション	説明
<code>-ofilename</code>	指定したファイルにメッセージを出力します。
<code>-oefilename</code>	起動エラー、致命的なエラー、アサーションをログするファイルを指定します。
<code>-onsize</code>	データベースサーバメッセージログファイルの最大サイズを指定します。ログファイルがこのサイズに達すると、現在のファイルが拡張子 <code>.old</code> の付いた名前に変更され、新しいファイルが作成されます。
<code>-OSsize</code>	メッセージログファイルのサイズを制限します。
<code>-ofilename</code>	データベースサーバのメッセージログファイルをトランケートし、出力メッセージを追加します。
<code>-ppacket-size</code>	最大通信パケットサイズを設定します。
<code>-pc</code>	同一コンピュータ接続以外のすべての通信パケットを圧縮します。
<code>-pfilename</code>	指定されたファイルにプロセス ID を書き込みます。
<code>-phl { OFF   ON   property-name, ... }</code>	指定したデータベースサーバプロパティの履歴追跡を設定します。
<code>-phs { [HH:]MM:SS   size [ K   M ]   max   default }</code>	プロパティ履歴の追跡に使用するメモリの最大容量を、割当時間またはバイトで設定します。
<code>-ptsize-in-bytes</code>	圧縮を適用する最小のネットワークパケットサイズを設定します。
<code>-qi</code>	データベースサーバシステムトレイアイコンまたはデータベースサーバメッセージウィンドウを非表示にします。
<code>-qn</code>	起動時にデータベースサーバメッセージウィンドウを最小化しないようにします。
<code>-qp</code>	データベースサーバメッセージウィンドウにパフォーマンスに関するメッセージを表示しないようにします。
<code>-qs</code>	起動エラーウィンドウを表示しないようにします。
<code>-qw</code>	データベースサーバメッセージウィンドウを表示しないようにします。
<code>-r</code>	読み込み専用モードでデータベースを開きます。
<code>-sfacility-ID</code>	Syslog facility ID を設定します。
<code>-sb { 0   1 }</code>	ブロードキャストに対するサーバの動作を指定します。
<code>-sbx { +   - }</code>	データベースサーバ上で起動されたデータベースのうち、ディスクサンドボックス設定が明示的に行われていないすべてのデータベースについて、デフォルトのディスクサンドボックス設定を制御します。

サーバオプション	説明
<code>-sffeature-list</code>	このデータベースサーバで実行されるデータベースの機能を保護します。
<code>-sjvm { ON   OFF }</code>	データベースサーバで動作しているすべてのデータベースに対して、データベースサーバが1つの Java VM を使用するように指定します。
<code>-skkey</code>	データベースサーバで無効になっている機能を有効にするためのキーを指定します。
<code>-supassword</code>	ユーティリティデータベース (utility_db) の DBA ユーザのパスワードを設定します。または、ユーティリティデータベースへの接続を無効にします。
<code>-tdsl</code>	TDS ログイン要求モードを設定します。
<code>-tminutes</code>	シャットダウンするまでのクライアントのアイドル時間を設定します (デフォルト値は 240 分)。
<code>-tlseconds</code>	デフォルトのクライアント活性タイムアウト (秒数) を設定します (デフォルト値は 120 秒)。
<code>-tmf</code>	トランザクションマネージャで、分散トランザクションを強制リカバリします。
<code>-tmtmilliseconds</code>	分散トランザクションの再エンリストタイムアウトを設定します。
<code>-tqtime</code>	終了時間を設定します。
<code>-ts</code>	データベースサーバのトレースセッションを設定します。
<code>-u</code>	バッファディスク I/O を使用します。
<code>-ua</code>	非同期 I/O の使用をオフにします。
<code>-uc</code>	データベースサーバをシェルモードで起動します。
<code>-ud</code>	デーモンとして実行します。
<code>-uf</code>	致命的なエラーまたはアサーションの失敗がデータベースサーバで発生したときにデータベースサーバが実行するアクションを指定します。
<code>-ufd</code>	致命的なエラーまたはアサーションの失敗がデータベースで発生したときにデータベースサーバが実行するアクションを指定します。
<code>-ui</code>	サーバ起動オプションウィンドウを開いてデータベースサーバメッセージウィンドウを表示するか、使用可能な表示がない場合はデータベースサーバをシェルモードで起動します。

サーバオプション	説明
<code>-um</code>	DBLauncher.app が実行中の場合、サーバ起動オプションウィンドウを開き、データベースサーバメッセージウィンドウを表示します。
<code>-uq</code>	データベースサーバをシェルモードで起動しますが、出力を表示しません。
<code>-utminutes</code>	min 分ごとにテンポラリファイルにタッチします。
<code>-ux</code>	データベースサーバメッセージウィンドウとサーバ起動オプションウィンドウを表示します。
<code>-v</code>	データベースサーバのバージョンを表示して停止します。
<code>-vss { +   - }</code>	ボリュームシャドウコピーサービス (VSS) を有効または無効にします。
<code>-wc [ +   - ]</code>	データベースサーバで実行されるデータベースの書き込みチェックサムを有効にします。
<code>-xlist</code>	カンマで区切られた使用する通信プロトコルのリストを指定します。
<code>-xaauthentication-info</code>	監視サーバに対するデータベース名と認証文字列のリストを指定します。
<code>-xd</code>	データベースサーバがデフォルトのデータベースサーバにならないようにします。
<code>-xfstate-file</code>	データベースミラーリングシステムに関するステータス情報の管理に使用されるファイルのロケーションを指定します。
<code>-xmseconds</code>	新しい IP アドレスをチェックする時間を秒単位で設定します。最小値は 10 で、デフォルト値は 0 です。ポータブルデバイスの場合、デフォルト値は 120 です。
<code>-xs</code>	サーバ側の Web サービス通信プロトコルを指定します。
<code>-z</code>	通信リンクに関する診断情報を提供します。
<code>-ze</code>	データベースサーバメッセージウィンドウにデータベース環境変数を表示します。
<code>-zl</code>	各接続の最後に作成された SQL 文の取得をオンにします。
<code>-zninteger</code>	保持する要求ログファイルのコピー数を指定します。
<code>-zofilename</code>	要求ロギング情報を別個のファイルにリダイレクトします。
<code>-zoc filename</code>	Web サービスクライアント情報をファイルにリダイレクトします。

サーバオプション	説明
-zp	クエリオプティマイザが最近使用したプランの取得をオンにします。
-zrlevel	SQL オペレーションのログを有効にします。デフォルトは、NONE です。
-zssize	要求ログファイルのサイズを制限します。
-zt	要求タイミング情報のロギングをオンにします。

## データベースオプション

次のオプションは、データベースサーバコマンドのデータベースファイル名の後にだけ指定できます。

データベースオプション	説明
-afilename	名前付きトランザクションログファイルを適用します。
-adlog-directory-specification	データベースに適用されるトランザクションログファイルがあるディレクトリ (1 つまたは複数) を指定します。
-a/userid[;userid...]	指定したデータベースの指定したユーザに対して標準のユーザ認証を許可します。
-ar	トランザクションログと同じディレクトリ内にあるログファイルをデータベースに適用します。
-as	トランザクションログの適用後もデータベースの実行を継続します (-ad または -ar とともに使用)。
-dh	このサーバに対して dblocate が実行された場合、データベースを表示しません。
-ds	データベースの DB 領域のロケーションを指定します。
-ekkey	暗号化キーを指定します。
-m	指定のデータベースについて、各チェックポイントの実行後にトランザクションログをトランケート (削除) します。
-nname	データベースに名前を付けます。-n オプションは指定する位置によって意味が異なります。
-r	読み込み専用モードで指定のデータベースを開きます。データベースは変更できません。
-ru	ポイントインタイムリカバリの実行においてデータベースをリストアするタイムスタンプを指定します。

データベースオプション	説明
<code>-ruo</code>	ポイントインタイムリカバリの実行においてデータベースをリストアするトランザクションログのオフセットを指定します。
<code>-sbx {+ -}</code>	データベースに対するディスクサンドボックス機能を制御します。有効にすると、データベースファイルの読み込み/書き込み操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されます。
<code>-sNalternate-server-name</code>	データベースサーバ上で動作する1つのデータベースに代替サーバ名を割り当てます。
<code>-wc [+ -]</code>	データベースサーバで実行されるデータベースの書き込みチェックサムを有効にします。
<code>-xp on</code>	パートナーまたはコピーノードに対してデータベースミラーリングを有効にします。

## 備考

データベースサーバのコマンドの要素には、次のようなものがあります。

### Executable

`dbeng17` コマンドは、パーソナルデータベースサーバを起動します。

`dbsrv17` コマンドは、ネットワークデータベースサーバを起動します。

ネットワークサーバでは TCP/IP がサポートされるため、使用しているデスクトップコンピュータからタスクを実行できます。

Windows では、パーソナルデータベースサーバの実行プログラムの名前は `dbeng17.exe` です。UNIX オペレーティングシステムでは、`dbeng17` です。

Windows では、ネットワークデータベースサーバの実行プログラムの名前は `dbsrv17.exe` です。Linux および UNIX オペレーティングシステムでは、`dbsrv17` です。

### Server options

これらのオプションは、実行中のすべてのデータベースに対するデータベースサーバの動作を制御します。

### Database file

1つまたは複数のデータベースファイル名を指定するか、まったく指定しないこともできます。指定された各データベースが起動され、引き続きアプリケーションで使用できます。

### 警告

データベースファイルとトランザクションログファイルは、データベースサーバと同じ物理コンピュータに保存してください。または SAN や iSCSI 設定でアクセスできるようにしてください。リモートネットワークディレクトリにデータベースファイルやトランザクションログファイルを配置すると、パフォーマンスが低下したり、データが破壊されたり、データベースサーバが不安定になったりする可能性があります。

最適な結果を得るために、トランザクションログは、データベースファイルとは別のディスクに保存してください。

## Database options

開始するデータベースファイルごとに、その動作の特定の状態を制御するデータベースオプションを指定できます。

データベースオプションとサーバオプションでは通常、大文字と小文字が区別されます。オプションはすべて小文字で入力してください。

`database-file` には、データベースファイル名を指定します。ファイル拡張子を付けずに `database-file` を指定した場合、データベースサーバでは拡張子 `.db` を付けて `database-file` が検索されます。相対パスを使用する場合、そのパスは現在の作業フォルダと相対関係となります。フルパスも指定できます。

オプションもデータベースファイルも指定しなかった場合、Windows オペレーティングシステムではウィンドウが表示され、データベースファイルを検索できます。

データベースサーバをバッチファイルから起動する場合は、`dbspawn` ユーティリティを使用します。

パーソナルデータベースサーバでは、同時接続の最大数が 10 に制限されています。要求処理に使用されるのは 1 CPU あたり最大 4 コアまでで、ネットワーククライアント/サーバ接続はサポートされていません。デフォルトでは、パーソナルデータベースサーバは共有メモリプロトコルのみを使用します。パーソナルデータベースサーバで TCP/IP を使用する場合は、`-x` オプションを使用する必要があります。

それ以外にも、新しいデータベースの起動に必要なデフォルトの権限レベルや、CHECKPOINT 文の実行に必要な権限などで、多少の違いがあります。

デフォルトでは、データベースサーバページサイズは、コマンドラインで指定されたデータベースの最大ページサイズと同じ大きさです。データベースサーバがいったん起動すると、データベースサーバページより大きいページサイズのデータベースを起動することはできません。

### 例

次のコマンドは、パーソナルデータベースサーバで動作する SQL Anywhere サンプルデータベースを起動します。

```
dbeng17 "%SQLANYSAMP17%¥demo.db"
```

次のコマンドは、ネットワークデータベースサーバで動作する SQL Anywhere サンプルデータベースを起動します。

```
dbsrv17 "%SQLANYSAMP17%¥demo.db"
```

次の例は、キャッシュサイズ 3 MB で始まる `myserver` という名前のデータベースサーバを起動し、サンプルデータベースをロードします。

```
dbsrv17 -c 3m -n myserver "%SQLANYSAMP17%¥demo.db"
```

このセクションの内容:

#### [データベースサーバ起動オプション \[375 ページ\]](#)

データベースサーバオプションは、個々のデータベースだけでなく、データベースサーバ全体に適用されます。

#### [データベース起動オプション \[519 ページ\]](#)

データベースオプションは、コマンドラインでデータベースファイルの後に指定し、そのデータベースだけに適用されません。

#### [トラブルシューティング: データベースサーバの検出方法 \[539 ページ\]](#)

クライアントは、サーバの検出に成功すると、データベースの検出に移ります。

## 関連情報

[トランザクションログ \[279 ページ\]](#)

[データベースサーバ \[305 ページ\]](#)

[最大ページサイズの考慮事項 \[319 ページ\]](#)

[データベースサーバの起動方法 \[310 ページ\]](#)

[設定ファイル \[1066 ページ\]](#)

[Running a SQL Anywhere Database File that is Stored Remotely from the Server Machine \(サーバマシンからリモートで格納された SQL Anywhere データベースファイルの実行\)](#)

[サーババックグラウンド起動ユーティリティ \(dbspawn\) \[1161 ページ\]](#)

## 1.4.1 データベースサーバ起動オプション

データベースサーバオプションは、個々のデータベースだけでなく、データベースサーバ全体に適用されます。

このセクションの内容:

[@data データベースサーバオプション \[382 ページ\]](#)

指定した環境変数または設定ファイルからオプションを読み込みます。

[-? データベースサーバオプション \[383 ページ\]](#)

使用方法を表示します。

[-al データベースサーバオプション \[384 ページ\]](#)

指定したユーザに対して標準のユーザ認証を許可します。

[-b データベースサーバオプション \[385 ページ\]](#)

バルクオペレーションモードを使用します。

[-c データベースサーバオプション \[385 ページ\]](#)

データベースページや他のデータベースサーバ情報をキャッシュするために予約される初期メモリを設定します。

[-ca データベースサーバオプション \[387 ページ\]](#)

静的キャッシュサイズを強制的に適用します。

[-cc データベースサーバオプション \[388 ページ\]](#)

次回にデータベースが起動されるときに、キャッシュウォーミングに使用するデータベースページに関する情報を収集します。

[-cdb データベースサーバオプション \[389 ページ\]](#)

コックピットを起動します。

[-ch データベースサーバオプション \[392 ページ\]](#)

自動キャッシュ増加機能を利用した際の最大キャッシュサイズ上限値を設定します。

[-chx データベースサーバオプション \[393 ページ\]](#)

自動キャッシュ増加機能の上限となる最大キャッシュサイズを設定します。

- cl データベースサーバオプション [395 ページ]  
キャッシュサイズの動的な調整の下限値として、最小キャッシュサイズを設定します。
- cp データベースサーバオプション [396 ページ]  
クラスの検索先となる一連の追加ディレクトリまたは JAR ファイルを指定します。
- cr データベースサーバオプション [397 ページ]  
データベースが最後に実行されたときに収集した情報を使用して、キャッシュとデータベースページを再ロード (準備) します。
- cs データベースサーバオプション [398 ページ]  
データベースサーバメッセージウィンドウに、動的なキャッシュサイズ設定に関する統計情報を表示します。
- cv データベースサーバオプション [399 ページ]  
データベースサーバメッセージウィンドウでのキャッシュウォーミングに関するメッセージの表示を制御します。
- dt データベースサーバオプション [400 ページ]  
テンポラリファイルを保存するディレクトリを指定します。
- ec データベースサーバオプション [402 ページ]  
トランスポートレイヤセキュリティまたは単純難読化を使用して、すべてのクライアントとの間で転送される通信プロトコルパケット (DBLIB や ODBC など) をエンコードします。TDS 接続はこのオプションを指定してもエンコードされません。
- edi データベースサーバオプション [405 ページ]  
データベース独立性をオンにします。
- ep データベースサーバオプション [406 ページ]  
強い暗号で保護されたデータベースを起動する際、ユーザに暗号化キーの入力を求めます。
- es データベースサーバオプション [407 ページ]  
共有メモリを経由した暗号化されていない接続を許可します。
- f データベースサーバオプション [408 ページ]  
トランザクションログが失われた後で、データベースサーバを強制的に起動します。
- fc データベースサーバオプション [409 ページ]  
ファイルシステムフルのコールバック関数がある DLL (UNIX では共有オブジェクト) のファイル名を指定します。
- fips データベースサーバオプション [411 ページ]  
データベースや通信の強力な暗号化を行うため、データベースサーバが FIPS 認定の暗号化を使用するよう設定します。
- ga データベースサーバオプション [412 ページ]  
最後の非 HTTP クライアント接続を切断した後、データベースをアンロードします。
- gb データベースサーバオプション [413 ページ]  
サーバプロセスの優先度クラスを設定します。
- gc データベースサーバオプション [413 ページ]  
チェックポイント間の最大間隔を設定します。
- gd データベースサーバオプション [414 ページ]  
実行中のデータベースサーバ上でデータベースを起動または停止するために必要な権限を設定します。
- ge データベースサーバオプション [416 ページ]  
外部関数のスタックサイズを設定します。
- gf データベースサーバオプション [417 ページ]

- サーバがトリガを起動できないようにします。
- gk データベースサーバオプション [417 ページ]  
データベースサーバの停止に必要な権限を設定します。
  - gl データベースサーバオプション [418 ページ]  
LOAD TABLE 文を使用したデータのロードに必要な権限と、UNLOAD 文または UNLOAD TABLE 文を使用したデータのアンロードに必要な権限を設定します。
  - gm データベースサーバオプション [420 ページ]  
データベースサーバへの同時接続の数を制限します。
  - gn データベースサーバオプション [421 ページ]  
データベースサーバのマルチプログラミングレベルを設定します。
  - gna データベースサーバオプション [422 ページ]  
データベースサーバのマルチプログラミングレベルの自動チューニングを制御します。
  - gnh データベースサーバオプション [423 ページ]  
最大マルチプログラミングレベル (ネットワークデータベースサーバが同時に実行できるタスクの最大数) を設定します。
  - gnl データベースサーバオプション [424 ページ]  
データベースサーバのマルチプログラミングレベルの最小値を設定します。
  - gns データベースサーバオプション [425 ページ]  
データベースサーバメッセージログにマルチプログラミングレベルの自動調整の統計を表示します。
  - gp データベースサーバオプション [427 ページ]  
許可されるデータベースページサイズの最大値を設定します。
  - gr データベースサーバオプション [428 ページ]  
システム障害からのリカバリに要する最長時間を分数で設定します。
  - gss データベースサーバオプション [429 ページ]  
データベースサーバでワーカごとのスタックサイズを設定します。
  - gt データベースサーバオプション [430 ページ]  
使用できる物理プロセッサの最大数を設定します (ライセンスされたプロセッサの数を上限とする)。このオプションは、マルチプロセッサシステムでのみ役立ちます。
  - gta データベースサーバオプション [431 ページ]  
データベースサーバが使用できる論理プロセッサを設定します。
  - gtc データベースサーバオプション [432 ページ]  
データベースサーバで許可される、プロセッサの最大同時実行性を制御します。
  - gtp データベースサーバオプション [434 ページ]  
トポロジを考慮したスケジューリングをオンまたはオフにします。このオプションは、マルチプロセッサシステムでのみ役立ちます。
  - gu データベースサーバオプション [435 ページ]  
データベースファイル管理文 (データベースの作成や削除などの文) の実行に必要な権限を設定します。
  - im データベースサーバオプション [437 ページ]  
データベースサーバをインメモリで実行し、ディスクへの書き込みをなくしたり軽減したりします。検証オプションを除き、インメモリモードには別途ライセンスが必要です。
  - k データベースサーバオプション [440 ページ]

Windows パフォーマンスモニタ統計とステートメントパフォーマンスサマリ統計の収集を制御します。

- kl [データベースサーバオプション \[441 ページ\]](#)  
Kerberos GSS-API ライブラリ (UNIX の場合は共有オブジェクト) のファイル名を指定し、データベースサーバへの Kerberos 認証による接続を有効にします。
- kp [データベースサーバオプション \[442 ページ\]](#)  
Kerberos サーバプリンシパルを指定し、データベースサーバへの Kerberos 認証接続を有効にします。
- kr [データベースサーバオプション \(廃止予定\) \[443 ページ\]](#)  
Kerberos サーバプリンシパルの領域を指定し、データベースサーバへの Kerberos 認証による接続を有効にします。
- krb [データベースサーバオプション \[444 ページ\]](#)  
データベースサーバへの Kerberos 認証接続を有効にします。
- ks [データベースサーバオプション \[445 ページ\]](#)  
Windows パフォーマンスモニタで、データベースサーバからカウンタ値を収集するために使用される共有メモリの作成を無効にします。
- ksc [データベースサーバオプション \[446 ページ\]](#)  
Windows パフォーマンスモニタでモニタできる接続の最大数を指定します。
- ksd [データベースサーバオプション \[447 ページ\]](#)  
Windows パフォーマンスモニタでモニタできるデータベースの最大数を指定します。
- m [データベースサーバオプション \[447 ページ\]](#)  
チェックポイントの実行後にトランザクションログをトランケートします。
- n [データベースサーバオプション \[448 ページ\]](#)  
データベースサーバの名前を設定します。
- ncs [データベースサーバオプション \[450 ページ\]](#)  
NCS (Native Component Supportability) ライブラリを使用した、SAP Solution Manager へのステータスイベント送信を有効にします。
- ncsd [データベースサーバオプション \[451 ページ\]](#)  
NCS (Native Component Supportability) ライブラリを使用した、SAP Solution Manager へのステータスイベント送信を有効にし、使用する NCS 設定ファイルの場所を指定します。
- o [データベースサーバオプション \[452 ページ\]](#)  
すべてのデータベースサーバメッセージをデータベースサーバメッセージログファイルに出力します。
- oe [データベースサーバオプション \[453 ページ\]](#)  
起動エラー、致命的なエラー、アサーションをロギングするファイルの名前を指定します。
- on [データベースサーバオプション \[454 ページ\]](#)  
データベースサーバメッセージログの最大サイズを指定します。ログファイルがこのサイズに達すると、現在のファイルが拡張子 `.old` の付いた名前に変更され、新しいファイルが作成されます。
- os [データベースサーバオプション \[455 ページ\]](#)  
データベースサーバメッセージログファイルの最大サイズを指定します (このサイズに達するとログのファイル名が変更されます)。
- ot [データベースサーバオプション \[456 ページ\]](#)  
データベースサーバのメッセージログファイルをトランケートし、出力メッセージを追加します。
- p [データベースサーバオプション \[457 ページ\]](#)

通信パケットの最大サイズを設定します。

**-pc データベースサーバオプション [457 ページ]**

同一コンピュータ接続以外のすべての接続のパケットを圧縮します。

**-pf データベースサーバオプション [458 ページ]**

指定されたファイルにプロセス ID を書き込みます。

**-phl データベースサーバオプション [459 ページ]**

指定したデータベースサーバプロパティの履歴追跡を設定します。

**-phs データベースサーバオプション [460 ページ]**

プロパティ履歴の追跡に使用するメモリの最大容量を、割当時間またはバイトで設定します。

**-pt データベースサーバオプション [461 ページ]**

TCP/IP パケットの圧縮サイズの制限値を増減します。

**-qi データベースサーバオプション [462 ページ]**

データベースサーバのシステムトレイアイコンと、データベースサーバのメッセージウィンドウを表示するかどうかを制御します。

**-qn データベースサーバオプション [463 ページ]**

起動時にデータベースサーバメッセージウィンドウを最小化しないことを指定します。

**-qp データベースサーバオプション [464 ページ]**

データベースサーバメッセージウィンドウに、パフォーマンスに関するメッセージを表示しないように指定します。

**-qs データベースサーバオプション [465 ページ]**

起動エラーウィンドウを表示しないようにします。

**-qw データベースサーバオプション [466 ページ]**

データベースサーバメッセージウィンドウを表示しないことを指定します。

**-r データベースサーバオプション [467 ページ]**

データベースサーバ上で起動されるすべてのデータベースを、強制的に読み込み専用にします。

**-s データベースサーバオプション [468 ページ]**

Syslog メッセージのユーザ ID を設定します。

**-sb データベースサーバオプション [468 ページ]**

ブロードキャストへのデータベースサーバの反応を指定します。

**-sbn データベースサーバオプション [470 ページ]**

データベースサーバ上で起動されたデータベースのうちディスクサンドボックス設定が明示的に行われていないすべてのデータベースについて、デフォルトのディスクサンドボックス設定を行います。

**-sclr データベースサーバオプション [471 ページ]**

データベースサーバで実行されているすべてのデータベースに対してデータベースサーバが 1 つの CLR 外部環境を使用するように指定します。

**-sf データベースサーバオプション [471 ページ]**

現在のデータベースサーバで実行中のデータベースで使用できる機能にユーザがアクセスできるかどうかを制御します。

**-sjvm データベースサーバオプション [478 ページ]**

データベースサーバで実行されているすべてのデータベースに対してデータベースサーバが 1 つの Java VM を使用するように指定します。

**-sk データベースサーバオプション [479 ページ]**

SYSTEM セキュリティ機能キーを作成し、認証コードを設定します。これにより、データベースサーバの保護された機能にアクセスできるようになります。

- su [データベースサーバオプション \[480 ページ\]](#)  
ユーティリティデータベース (utility\_db) のユーザ ID とパスワードを設定します。または、ユーティリティデータベースへの接続を無効にします。
- tdsl [データベースサーバオプション \[482 ページ\]](#)  
TDS ログインモードを設定します。
- ti [データベースサーバオプション \[483 ページ\]](#)  
非アクティブな接続を切断します。
- tl [データベースサーバオプション \[484 ページ\]](#)  
活性パケットを送信する期間を設定します。
- tmf [データベースサーバオプション \[485 ページ\]](#)  
トランザクションマネージャで、分散トランザクションを強制リカバリします。
- tmt [データベースサーバオプション \[485 ページ\]](#)  
分散トランザクションに参加するための再エンリストタイムアウトを設定します。
- tq [データベースサーバオプション \[486 ページ\]](#)  
指定の時刻にサーバを停止します。
- ts [データベースサーバオプション \[487 ページ\]](#)  
データベースサーバのトレースセッションを設定します。
- u [データベースサーバオプション \[488 ページ\]](#)  
オペレーティングシステムのディスクキャッシュを使用してファイルを開きます。
- ua [データベースサーバオプション \[489 ページ\]](#)  
非同期 I/O の使用をオフにします。
- uc [データベースサーバオプション \[489 ページ\]](#)  
データベースサーバをシェルモードで起動します。
- ud [データベースサーバオプション \[490 ページ\]](#)  
データベースサーバをデーモンとして実行します。
- uf [データベースサーバオプション \[491 ページ\]](#)  
致命的なエラーまたはアサーションの失敗がデータベースサーバで発生したときにデータベースサーバが実行するアクションを指定します。
- ufd [データベースサーバオプション \[492 ページ\]](#)  
致命的なエラーまたはアサーションの失敗がデータベースで発生したときにデータベースサーバが実行するアクションを指定します。
- ui [データベースサーバオプション \[494 ページ\]](#)  
サーバ起動オプションウィンドウが開き、データベースサーバメッセージウィンドウが表示され、X Window Server が起動するかどうかにかかわらず、データベースサーバが起動します。
- um [データベースサーバオプション \[495 ページ\]](#)  
DBLauncher . app 内の新しいウィンドウにデータベースサーバメッセージを表示します。
- uq [データベースサーバオプション \[496 ページ\]](#)  
データベースサーバをシェルモードで起動しますが、出力を表示しません。
- ut [データベースサーバオプション \[496 ページ\]](#)

テンポラリファイルをタッチします。

[-ux データベースサーバオプション \[497 ページ\]](#)

Linux で、サーバ起動オプションウィンドウを開くか、データベースサーバメッセージウィンドウを表示します (X-Window Server を使用)。

[-v データベースサーバオプション \[498 ページ\]](#)

ソフトウェアのバージョンを表示します。

[-vss データベースサーバオプション \[498 ページ\]](#)

ボリュームシャドウコピーサービス (VSS) を有効または無効にします。

[-wc データベースサーバオプション \[499 ページ\]](#)

チェックサムがデフォルトでデータベースによって有効にされていない場合、このデータベースサーバのすべてのデータベースに対する書き込み操作でチェックサムを有効にするかどうかを制御します。

[-x データベースサーバオプション \[500 ページ\]](#)

サーバ側のネットワーク通信プロトコルを指定します。

[-xa データベースサーバオプション \[502 ページ\]](#)

監視サーバに対するデータベース名と認証文字列のカンマ区切りのリストを指定します。

[-xd データベースサーバオプション \[504 ページ\]](#)

データベースサーバがデフォルトのデータベースサーバにならないようにします。

[-xf データベースサーバオプション \[505 ページ\]](#)

データベースミラーリングシステムに関するステータス情報の管理に使用されるファイルのロケーションを指定します。

[-xm データベースサーバオプション \[506 ページ\]](#)

データベースサーバが新しい IP アドレスをチェックする頻度を指定します。

[-xs データベースサーバオプション \[507 ページ\]](#)

サーバ側の Web サービス通信プロトコルを指定します。

[-z データベースサーバオプション \[509 ページ\]](#)

診断通信メッセージなどのメッセージをトラブルシューティングのために表示します。

[-ze データベースサーバオプション \[510 ページ\]](#)

データベースサーバメッセージウィンドウにデータベース環境変数を表示します。

[-zl データベースサーバオプション \[511 ページ\]](#)

サーバ上のデータベースへの各接続について、最後に作成された SQL 文の取得を有効にします。

[-zn データベースサーバオプション \[512 ページ\]](#)

保持する要求ログファイルのコピー数を指定します。

[-zo データベースサーバオプション \[513 ページ\]](#)

通常のログファイルとは別のファイルに、要求ログ情報をリダイレクトします。

[-zoc データベースサーバオプション \[514 ページ\]](#)

Web サービスクライアント情報をファイルにリダイレクトします。

[-zp データベースサーバオプション \[514 ページ\]](#)

クエリオブティマイザが最近使用したプランの取得をオンにします。

[-zr データベースサーバオプション \[515 ページ\]](#)

操作の要求ロギングを有効にします。

-zs データベースサーバオプション [517 ページ]

要求ログのサイズを制限します。

-zt データベースサーバオプション [518 ページ]

要求タイミング情報のロギングをオンにします。

## 関連情報

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

### 1.4.1.1 @data データベースサーバオプション

指定した環境変数または設定ファイルからオプションを読み込みます。

#### 構文

```
dbsrv17 @data ...
```

## 適用対象

すべてのオペレーティングシステム。

このオプションは、言語選択ユーティリティ (dblang) とサーババックグラウンド起動ユーティリティ (dbspawn) を除くすべてのデータベースユーティリティでもサポートされます。

## 備考

このオプションを使用して、指定された環境変数または設定ファイルからコマンドラインオプションを読み出します。指定された名前と同じ名前の環境変数と設定ファイルが両方存在する場合は、環境変数が使用されます。

設定ファイルには、改行を含めたり、あらゆるオプションの設定を格納したりできます。

設定ファイルの情報を (パスワードが含まれるなどの理由で) 保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容を暗号化してください。

@data パラメータはコマンドの任意の位置に指定でき、ファイルに含まれるパラメータがその位置に挿入されます。複数のファイルを指定可能で、ファイル指定子をコマンドラインオプションで使用できます。

## 例

次の例では、**myserver** という名前のデータベースサーバが起動され、サンプルデータベースがロードされます。データベースサーバは、総物理メモリ量の 40% である静的キャッシュサイズに自動的に調整されません。

```
-c 4096
-n myservers
"c:¥mydatabase.db"
```

この設定ファイルを `c:¥config.txt` として保存すると、コマンドで次のように使用できます。

```
dbsrv17 @c:¥config.txt
```

次の設定ファイルにはコメントが含まれています。

```
#This is the server name:
-n MyServer
#These are the protocols:
-x tcpip
#This is the database file
my.db
```

次の文は、データベースサーバをキャッシュサイズ 4 MB で起動し、mydatabase.db というデータベースをロードするオプションを格納する環境変数を設定します。

```
SET envvar=-c 4096 "c:¥mydatabase.db";
```

次のコマンドは、**envvar** という環境変数を使用してデータベースサーバを起動します。

```
dbsrv17 @envvar
```

## 関連情報

[設定ファイル \[1066 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

## 1.4.1.2 -? データベースサーバオプション

使用法を表示します。

### 構文

```
dbsrv17 -?
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションを指定すると、各サーバオプションの簡単な説明が表示されます。

### 1.4.1.3 -al データベースサーバオプション

指定したユーザに対して標準のユーザ認証を許可します。

#### 構文

```
dbsrv17 -al userid[ ;userid ... ] ...
```

UNIX では、複数のパラメータを指定するときに引用符が必要です。

## 適用対象

すべてのオペレーティングシステム。

## 備考

セミコロンで区切られたユーザ ID リストで指定されたユーザに対して、標準認証の使用を許可します。最大 5 つのユーザ ID を指定できます。

login\_mode オプションに標準認証が含まれておらず、ユーザが代わりにの方法を使用して認証できない場合に、このオプションが便利です。完全な DBA 権限を持つユーザは、常に標準認証を使用して認証できます。

## 関連情報

[login\\_mode オプション \[717 ページ\]](#)

## 1.4.1.4 -b データベースサーバオプション

バルクオペレーションモードを使用します。

### 構文

```
dbsrv17 -b ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

このオプションは、Interactive SQL INPUT 文を使用して、大量のデータをデータベースにロードする場合に役立ちます。

LOAD TABLE を使用してデータをバルクロードする場合は、-b オプションは使用しないでください。

このオプションを使用した場合、データベースサーバでは、1つのアプリケーションで1つの接続しか確立できません。ロールバックログは保存されますが、トランザクションログは保存されません。マルチユーザロック機構はオフになっています。

-b オプションでデータをロードした後、初めてデータベースサーバを起動する場合は、新しいトランザクションログファイルを使用してください。

バルクオペレーションモードにしても、トリガは起動不可になりません。

## 1.4.1.5 -c データベースサーバオプション

データベースページや他のデータベースサーバ情報をキャッシュするために予約される初期メモリを設定します。

### 構文

```
dbsrv17 -c { size[ k | m | g | p ] } ...
```

### 適用対象

すべてのオペレーティングシステム。

## 備考

データベースサーバキャッシュで利用できるキャッシュメモリの量は、パフォーマンスを制御する主要な要因の 1 つになります。初期のキャッシュメモリ量は、`-c` サーバオプションを使用して設定できます。データベースサーバ用のキャッシュメモリが大きいほど、パフォーマンスは向上します。

`size` には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ `k`、`m`、`g` のいずれかを使用してください。

単位 `p` は、物理システムメモリの総量と、サポートされる最大キャッシュサイズのうち、いずれか小さい方のパーセンテージを表します。サポートされる最大キャッシュサイズは、オペレーティングシステムによって異なります。例:

- 2.5 GB - Windows 32 ビット Advanced Server、Enterprise Server、Datacenter Server
- 3.5 GB - Windows x64 Edition 上の 32 ビットデータベースサーバ
- 1.5 GB - その他すべての 32 ビット Windows オペレーティングシステム
- 64 ビットのデータベースサーバの場合、キャッシュサイズは無制限と見なされる

`p` を使用すると、引数はパーセンテージを表します。`p` の代わりに `%` も使用できますが、Windows バッチファイルや Windows の一部の代替コマンドシェルでは、`%` 文字を 2 つ指定して `%` 文字をエスケープする必要があります。たとえば、バッチファイル内で、初期キャッシュサイズを物理システムメモリの 50 % に設定するには、次のようなコマンドを使用します。

```
dbspawn dbsrv17 -c 50%% ...
```

`-c` オプションを指定しないと、初期キャッシュの割り当てサイズが次のように計算されます。

### Windows

式は次のとおりです。

```
max( 2 MB, min( dbsize, 0.25*TotalPhysicalMemory ) );
```

`dbsize` は起動したデータベースファイルの合計サイズです。`TotalPhysicalMemory` は、コンピュータの物理メモリの合計量です。

### UNIX

最小で 8 MB です。

```
max( 8 MB, min( 0.1*( physical-memory + available-swap ) , database-size*1.1 )
```

## i 注記

初期キャッシュサイズまたは最小キャッシュサイズを最大キャッシュサイズの 1/8 よりも小さな値に設定しようとする、初期キャッシュサイズおよび最小キャッシュサイズは最大キャッシュサイズに比例して自動的に増加します。

動的なキャッシュサイズの変更を無効 (`-ca` オプション) にすると、使用されるキャッシュサイズが使用可能なメモリのサイズによって制限されることがあります。

データベースサーバメッセージウィンドウには起動時のキャッシュサイズが表示されます。また、次の文を使用して現在のキャッシュサイズを取得することもできます。

```
SELECT PROPERTY( 'CurrentCacheSize' );
```

## 例

次の例は、キャッシュサイズ 3 MB で始まる **myserver** という名前のデータベースサーバを起動し、サンプルデータベースをロードします。

```
dbsrv17 -c 3m -n myserver "%SQLANYSAMPl7%¥demo.db"
```

## 関連情報

[動的キャッシュサイズ決定 \[1373 ページ\]](#)

[動的キャッシュサイズ決定 \(UNIX\) \[1374 ページ\]](#)

[ヒント: パフォーマンス向上のためのキャッシュの使用 \[1370 ページ\]](#)

[-ca データベースサーバオプション \[387 ページ\]](#)

[-cc データベースサーバオプション \[388 ページ\]](#)

[-ch データベースサーバオプション \[392 ページ\]](#)

[-chx データベースサーバオプション \[393 ページ\]](#)

[-cl データベースサーバオプション \[395 ページ\]](#)

[-cr データベースサーバオプション \[397 ページ\]](#)

[-cs データベースサーバオプション \[398 ページ\]](#)

[-cv データベースサーバオプション \[399 ページ\]](#)

## 1.4.1.6 -ca データベースサーバオプション

静的キャッシュサイズを強制的に適用します。

### 構文

```
dbsrv17 -ca 0 ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

-ca 0 オプションを指定すると、自動キャッシュサイズチューニングを無効にすることができます。-ca 0 オプションの設定がない場合、データベースサーバは自動的にキャッシュサイズを増加します。このオプションを指定した場合、動的メモリが足りなくなったことを示すエラーがデータベースサーバで発生しても、キャッシュサイズは引き続き調整されます。

このサーバオプションは、-ca 0 という形式でのみ使用してください。

### 例

```
dbsrv17 -c 40P -ca 0 -n myserver "%SQLANYAMP17%¥demo.db"
```

## 関連情報

[動的キャッシュサイズ決定 \[1373 ページ\]](#)

[-c データベースサーバオプション \[385 ページ\]](#)

[-cc データベースサーバオプション \[388 ページ\]](#)

[-ch データベースサーバオプション \[392 ページ\]](#)

[-chx データベースサーバオプション \[393 ページ\]](#)

[-cl データベースサーバオプション \[395 ページ\]](#)

[-cr データベースサーバオプション \[397 ページ\]](#)

[-cs データベースサーバオプション \[398 ページ\]](#)

[-cv データベースサーバオプション \[399 ページ\]](#)

### 1.4.1.7 -cc データベースサーバオプション

次回にデータベースが起動されるときに、キャッシュウォーミングに使用するデータベースページに関する情報を収集します。

### 構文

```
dbsrv17 -cc{ + | - } ...
```

## デフォルト

デフォルトでは、ページ収集はオンになっています。

## 適用対象

すべてのオペレーティングシステム。

## 指定可能な値

- cc+ ページ収集をオンにします。
- cc- ページ収集をオフにします。

## 備考

収集がオンになると、データベースサーバは要求された各データベースページを追跡し続けます。ページの収集は、最大ページ数が収集されるか、データベースが停止されるか、または収集率が最小値を下回った場合に終了します。収集する最大ページ数を設定したり、収集率の値を指定したりすることはできません (この値は、キャッシュサイズとデータベースサイズに基づいています)。いったん収集が停止すると、要求されたページに関する情報はデータベースに記録されるので、それらのページは次回データベースが -cr オプションで開始されるときに、キャッシュの準備に使用できます。参照されたページの収集は、デフォルトではオンになっています。

## 関連情報

[キャッシュウォーミング \[1375 ページ\]](#)

- c データベースサーバオプション [385 ページ]
- ca データベースサーバオプション [387 ページ]
- ch データベースサーバオプション [392 ページ]
- chx データベースサーバオプション [393 ページ]
- cl データベースサーバオプション [395 ページ]
- cr データベースサーバオプション [397 ページ]
- cs データベースサーバオプション [398 ページ]
- cv データベースサーバオプション [399 ページ]

### 1.4.1.8 -cdb データベースサーバオプション

コックピットを起動します。

#### 構文

```
dbsrv17 -cdb "option [;...]"
```

```
option:  
DBF={ AUTO | cockpit-database-filename }  
START={ YES | NO }
```

## 適用対象

リモートデータアクセス機能がサポートされるすべてのオペレーティングシステム。

## 指定可能な値

### DBF option

コックピット設定 (アラートしきい値など) とアラート履歴が保存されるデータベースファイルを指定します。ファイルが存在しない場合は、作成されます。ファイルのフルパスを指定します。相対パスを使用する場合、そのパスは現在の作業フォルダと相対関係となります (または、ディスクサンドボックスが有効化されている場合、相対パスは、メインデータベースファイルがあるディレクトリに対する相対パスとみなされます)。

このファイルは、データベースサーバの動作中、または sa\_server\_option システムプロシージャを使用してコックピットの別のデータベースファイルが指定されるまで、コックピットのデフォルトデータベースファイルとなります。

または、テンポラリデータベースを使用してコックピットを実行するには、AUTO を指定します。コックピットが停止すると、コックピットのテンポラリデータベースが削除されるため、コックピット内で施された変更は保存されません。この値がデフォルトです。

### START option

データベースサーバの起動時に、コックピットを起動します。この動作は DBF オプションを指定したときのデフォルト動作です。

## 備考

コックピットに接続する必要がある場合に必ず接続できるよう、データベースサーバの起動時にコックピットを起動します。sa\_server\_option システムプロシージャを使用して、実行中のサーバ上でコックピットを起動および停止したり、コックピットの設定を変更したりすることもできます。コックピットは、設定を格納するデータベースファイルをモニタおよび使用すると同じデータベースサーバ上で実行します。

HTTPS を使用してコックピットと Web ブラウザ間の通信を暗号化するには、HTTPS 接続リスナ (-xs サーバオプション) を起動します。HTTPS 接続リスナを起動しない場合、コックピットは保護されていない HTTP 接続リスナを起動します。保護されていない HTTP 接続リスナを使用する場合は、第三者がコックピットと Web ブラウザ間の通信を閲覧して、機密情報 (コックピットへの接続に使用するデータベースユーザ名やパスワードなど) を容易に入手することができます。

xs データベースサーバを指定して HTTPS 接続リスナを作成するときは、DBN="SQLACockpit" と指定することによって、コックピットへの接続リスナを制限できます。

コックピットでは、データベースサーバが、HTTP 接続または HTTPS 接続に加えて、ローカルの共有メモリ接続を受け入れる必要があります。データベースサーバで `-ec` TLS データベースサーバオプションが使用される場合は、`-es` データベースサーバオプションを指定します。

コックピットにアクセスするための URL を取得するには、`CockpitURL` データベースサーバプロパティを使用します。例:

```
SELECT property( 'CockpitURL' );
```

現在のコックピット設定を取得するには、`CockpitDB` データベースサーバプロパティを使用します。

## 例

- 次のコマンドは、サンプルデータベースとコックピットを起動します。`-xs` データベースサーバオプションを使用して、コックピット、サンプルデータベース、およびデータベースサーバ上で動作するその他のデータベースで使用できる HTTPS 接続リスナを作成します。

```
dbsrv17 -cdb "DBF=C:¥temp¥cockpitsettings.db" -xs "HTTPS (PORT=443; IDENTITY=C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥Certificates¥rsaserver.id; IDENTITY_PASSWORD=test)" "C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

- 次のコマンドは、サンプルデータベースとコックピットを起動します。DBN パラメータを `-xs` データベースサーバオプションとともに指定して、コックピットのみで使用できる HTTPS 接続リスナを作成します。

```
dbsrv17 -cdb "DBF=C:¥temp¥cockpitsettings.db" -xs "HTTPS (PORT=443; DBN=SQLACockpit; IDENTITY=C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥Certificates¥rsaserver.id; IDENTITY_PASSWORD=test)" "C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

- 次のコマンドは、サンプルデータベースとコックピットを起動します。`-xs` データベースサーバオプションを使用して、HTTPS 接続リスナを作成します。また、このコマンドは TLS 暗号化を使用してデータベースサーバとクライアント間の TCP/IP 通信を暗号化し、一方でローカルの共有メモリ接続を可能にします。

```
dbsrv17 -cdb "DBF=C:¥temp¥cockpitsettings.db" -xs "HTTPS (PORT=443; DBN=SQLACockpit; IDENTITY=C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥Certificates¥rsaserver.id; IDENTITY_PASSWORD=test)" -ec "TLS (IDENTITY=C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥Certificates¥rsaserver.id; IDENTITY_PASSWORD=test)" -es "C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

- 次のコマンドは、コックピットのデフォルトデータベースファイルを設定しますが、コックピットを起動しません。

```
-cdb "START=NO; DBF=cockpitsettings"
```

後で、ファイル名を指定しないでコックピットを起動すると、コックピットのデフォルトデータベースファイルが使用されます。たとえば、`CALL sa_server_option( 'CockpitDB', 'START=YES' );` のように記述します。

## 警告

これらの例では、テストや開発を想定したサンプル ID ファイルを使用しています。運用システムでは使用しないでください。代わりに、アプリケーションを配備する前に独自の証明書で置換してください。

## 関連情報

[SQL Anywhere コックピット \[1349 ページ\]](#)

[コックピットの起動とコックピットへの接続 \(sa\\_server\\_option システムプロシージャ\) \[1353 ページ\]](#)

[-xs データベースサーバオプション \[507 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[コックピットのセキュリティ \[1359 ページ\]](#)

### 1.4.1.9 -ch データベースサーバオプション

自動キャッシュ増加機能を利用した際の最大キャッシュサイズ上限値を設定します。

#### 構文

```
dbsrv17 -ch { size[ k | m | g | p ] } ...
```

#### 適用対象

すべてのオペレーティングシステム。

#### 備考

このオプションは、データベースサーバが自動キャッシュ増加機能を実行するときに使用できる、キャッシュサイズの上限を設定します。デフォルトでは、上限値の概算は、最大キャッシュサイズとコンピュータの物理メモリ総量の 90% のうち、いずれか少ない方になります。

`size` には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ `k`、`m`、`g` のいずれかを使用してください。

単位 `p` は、物理システムメモリと、サポートされる最大キャッシュサイズのうち、いずれか小さい方のパーセンテージを表します。サポートされる最大キャッシュサイズは、オペレーティングシステムによって異なります。例:

- 2.5 GB - Windows 32 ビット Advanced Server、Enterprise Server、Datacenter Server
- 3.5 GB - Windows x64 Edition 上の 32 ビットデータベースサーバ
- 1.5 GB - その他すべての 32 ビット Windows オペレーティングシステム
- 64 ビットのデータベースサーバの場合、キャッシュサイズは無制限と見なされる

`p` を使用すると、引数はパーセンテージを表します。`p` の代わりに `%` も使用できますが、Windows オペレーティングシステムでは `%` 文字をエスケープしないと使用できません。初期キャッシュサイズを物理システムメモリの 50% に設定するには、次のコマンドを実行します。

```
dbsrv17 -ch 50%% ...
```

-ch を使用して 64 MB 未満のキャッシュサイズを指定した場合は、データベースサーバによって最大キャッシュが 64 MB に調整されます。64 MB 未満の最大キャッシュサイズを指定することはおすすめませんが、-chx オプションを使用して指定することは可能です。

#### 例

次の例は、silver という名前のデータベースサーバを最大キャッシュサイズ 64 MB で起動し、サンプルデータベースをロードします。

```
dbsrv17 -ch 64m -n silver "%SQLANYXSAMP17%¥demo.db"
```

## 関連情報

[動的キャッシュサイズ決定 \[1373 ページ\]](#)

[-c データベースサーバオプション \[385 ページ\]](#)

[-ca データベースサーバオプション \[387 ページ\]](#)

[-cc データベースサーバオプション \[388 ページ\]](#)

[-chx データベースサーバオプション \[393 ページ\]](#)

[-cl データベースサーバオプション \[395 ページ\]](#)

[-cr データベースサーバオプション \[397 ページ\]](#)

[-cs データベースサーバオプション \[398 ページ\]](#)

[-cv データベースサーバオプション \[399 ページ\]](#)

### 1.4.1.10 -chx データベースサーバオプション

自動キャッシュ増加機能の上限となる最大キャッシュサイズを設定します。

#### 構文

```
dbsrv17 -chx { size[ k | m | g | p ] } ...
```

## デフォルト

最低でも 512 MB のアドレス領域が、キャッシュ以外の使用に確保されます。

## 適用対象

すべてのオペレーティングシステム。

## 備考

### 警告

-chx オプションを使用して 32 ビットサーバで大きなキャッシュサイズを設定する場合は、必ずテストを行って、アドレス領域をさらに使用しても、サーバアドレス領域を使用する下記のような他のコンポーネントに影響を及ぼさないことを確認してください。

- サーバがロードする必要のある DLL
- リモートテーブルアクセス用の ODBC ドライバ
- ネットワークパケットバッファ
- 外部ストアプロシージャ
- キャッシュ以外のメモリ割り付け

サイズの大きいキャッシュが必要な場合は、64 ビット版のオペレーティングシステムで 64 ビット版のデータベースサーバを使用してください。

最低でも 512 MB のアドレス領域をキャッシュ以外の使用に確保するため、通常、32 ビットのデータベースサーバではキャッシュサイズが制限されます。確保される最大キャッシュサイズからキャッシュ以外に使用するアドレス領域を差し引いたサイズを指定する場合は、-chx オプションを使用します。サイズの大きいキャッシュを使用すると、データベースサーバが不安定になる可能性があります。このオプションの使用には十分注意してください。

サポートされる最大キャッシュサイズは、オペレーティングシステムによって異なります。例:

- 2.5 GB - Windows 32 ビット Advanced Server、Enterprise Server、Datacenter Server
- 3.5 GB - Windows x64 Edition 上の 32 ビットデータベースサーバ
- 1.5 GB - その他すべての 32 ビット Windows オペレーティングシステム
- 64 ビットのデータベースサーバの場合、キャッシュサイズは無制限と見なされる

size には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ k、m、g のいずれかを使用してください。

単位 p は、物理システムメモリと、最大キャッシュサイズのうち、いずれか小さい方のパーセンテージを表します。p の代わりに % も使用できますが、Windows ではほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。

-chx オプションを使用して 64 MB 未満の最大キャッシュサイズを指定することもできますが、そうしたサイズの最大キャッシュの使用はおすすめできません。

## 関連情報

[ヒント: パフォーマンス向上のためのキャッシュの使用 \[1370 ページ\]](#)

- c データベースサーバオプション [385 ページ]
- ca データベースサーバオプション [387 ページ]
- cc データベースサーバオプション [388 ページ]
- ch データベースサーバオプション [392 ページ]
- cl データベースサーバオプション [395 ページ]
- cr データベースサーバオプション [397 ページ]
- cs データベースサーバオプション [398 ページ]
- cv データベースサーバオプション [399 ページ]

### 1.4.1.11 -cl データベースサーバオプション

キャッシュサイズの動的な調整の下限値として、最小キャッシュサイズを設定します。

#### 構文

```
dbsrv17 -cl { size[ k | m | g | p ] } ...
```

#### デフォルト

Windows では 2 MB

UNIX では 8 MB

#### 適用対象

すべてのオペレーティングシステム。

#### 備考

このオプションは、キャッシュの下限値を設定します。-c を指定し、-cl を指定しないと、最小キャッシュサイズは初期キャッシュサイズ (-c 設定) に設定されます。-c と-cl をどちらも指定しないと、必要に応じてキャッシュを縮小できるように、最小キャッシュサイズは低い定数値に設定されます。Windows プラットフォームでは、この値は 2 MB になります。

size には、メモリ容量をバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ k、m、g のいずれかを使用してください。

単位 p は、物理システムメモリと、サポートされる最大キャッシュサイズのうち、いずれか小さい方のパーセンテージを表します。サポートされる最大キャッシュサイズは、オペレーティングシステムによって異なります。例:

- 2.5 GB - Windows 32 ビット Advanced Server、Enterprise Server、Datacenter Server
- 3.5 GB - Windows x64 Edition 上の 32 ビットデータベースサーバ
- 1.5 GB - その他すべての 32 ビットシステム
- 64 ビットのデータベースサーバの場合、キャッシュサイズは無制限と見なされる

p を使用すると、引数はパーセンテージを表します。p の代わりに % も使用できますが、Windows ではほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。

たとえば、初期キャッシュサイズを物理システムメモリの 50% に設定するには、次のコマンドを実行できます。

```
dbsrv17 -cl 50%% ...
```

### i 注記

初期キャッシュサイズまたは最小キャッシュサイズを最大キャッシュサイズの 8 分の 1 より小さく設定しようとすると、サポートされるキャッシュサイズに合わせて自動的にキャッシュサイズが増やされます。

### 例

次の例は、silver という名前のデータベースサーバを最小キャッシュサイズ 5 MB で起動し、データベースファイル example.db をロードします。

```
dbsrv17 -cl 5m -n silver "c:¥example.db"
```

## 関連情報

[動的キャッシュサイズ決定 \[1373 ページ\]](#)

[-c データベースサーバオプション \[385 ページ\]](#)

[-ca データベースサーバオプション \[387 ページ\]](#)

[-cc データベースサーバオプション \[388 ページ\]](#)

[-ch データベースサーバオプション \[392 ページ\]](#)

[-chx データベースサーバオプション \[393 ページ\]](#)

[-cr データベースサーバオプション \[397 ページ\]](#)

[-cs データベースサーバオプション \[398 ページ\]](#)

[-cv データベースサーバオプション \[399 ページ\]](#)

## 1.4.1.12 -cp データベースサーバオプション

クラスの検索先となる一連の追加ディレクトリまたは JAR ファイルを指定します。

### 構文

```
dbsrv17 -cp location[ ;location ... ] ...
```

UNIX では、複数のパラメータを指定する場合に二重引用符が必要です。

## 適用対象

すべてのオペレーティングシステム。

## 備考

データベース内の Java で使用しているすべてのクラスと JAR ファイルはデータベース内にインストールする必要があります。クラスと JAR ファイルをデータベース内に保存すると、データベースを簡単に別のコンピュータやオペレーティングシステムに移動できます。また、クラスと JAR ファイルをデータベース内にインストールすると、データベースサーバのクラスローダでクラスとリソースをデータベースからフェッチできるという利点もあります。その結果、データベース内の Java を使用している各接続で、これらのクラスの独自のインスタンスと、クラス内の静的変数の独自のコピーを使用できます。

ただし、クラスまたは JAR ファイルをシステムクラスローダによってロードする必要がある場合は、`java_class_path` データベースオプションまたは `-cp` データベースサーバオプションで指定できます。どちらのオプションを指定しても、Java VM の起動用にデータベースサーバによって構築されるクラスパスにクラスと JAR ファイルが追加されます。`java_class_path` データベースオプションは、サーバが複数のデータベースを実行していて、それぞれのデータベースにシステムクラスローダでロードする必要のある異なるセットの JAR とディレクトリがあるときに便利です。`-cp` データベースサーバオプションは、サーバ上のすべてのデータベースが同一のクラスまたは JAR ファイルを必要とするときに便利です。

## 関連情報

[java\\_class\\_path option \[713 ページ\]](#)

### 1.4.1.13 -cr データベースサーバオプション

データベースが最後に実行されたときに収集した情報を使用して、キャッシュとデータベースページを再ロード（準備）します。

#### 構文

```
dbsrv17 -cr{ + | - } ...
```

## デフォルト

デフォルトでは、キャッシュウォーミングはオンになっています。

## 適用対象

すべてのオペレーティングシステム。

## 指定可能な値

- cr+ キャッシュウォーミングをオンにします。
- cr- キャッシュウォーミングをオフにします。

## 備考

データベースが最後に起動したときに参照されたページを使用して、データベースサーバにキャッシュを準備するよう指示できます (ページ収集は -cc オプションを使用してオンにします)。キャッシュウォーミングは、デフォルトではオンになっています。データベースが起動されると、サーバはデータベースをチェックして、データベースが最後に起動されたときに要求されたページの収集があるかどうかを確認します。データベースにこの情報が含まれている場合、前に参照したページがキャッシュにロードされます。

データベースが最後に起動されたときに参照されたページのキャッシュを準備することで、同じクエリまたは似たようなクエリがデータベースの起動のたびに実行されるような場合に、パフォーマンスを改善できます。

## 関連情報

[キャッシュウォーミング \[1375 ページ\]](#)

[-cc データベースサーバオプション \[388 ページ\]](#)

[-cl データベースサーバオプション \[395 ページ\]](#)

[-cs データベースサーバオプション \[398 ページ\]](#)

[-cv データベースサーバオプション \[399 ページ\]](#)

### 1.4.1.14 -cs データベースサーバオプション

データベースサーバメッセージウィンドウに、動的なキャッシュサイズ設定に関する統計情報を表示します。

#### 構文

```
dbsrv17 -CS ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

トラブルシューティング目的で、キャッシュサイズのチューニング方法を決めるためにデータベースサーバで使用される統計が、データベースサーバメッセージウィンドウに表示されます。

## 関連情報

[キャッシュウォーミング \[1375 ページ\]](#)

[-c データベースサーバオプション \[385 ページ\]](#)

[-ca データベースサーバオプション \[387 ページ\]](#)

[-cc データベースサーバオプション \[388 ページ\]](#)

[-ch データベースサーバオプション \[392 ページ\]](#)

[-chx データベースサーバオプション \[393 ページ\]](#)

[-cl データベースサーバオプション \[395 ページ\]](#)

[-cr データベースサーバオプション \[397 ページ\]](#)

[-cv データベースサーバオプション \[399 ページ\]](#)

### 1.4.1.15 -cv データベースサーバオプション

データベースサーバメッセージウィンドウでのキャッシュウォーミングに関するメッセージの表示を制御します。

#### 構文

```
dbsrv17 -CV{ + | - } ...
```

## デフォルト

キャッシュウォーミングメッセージは表示されません。

## 適用対象

すべてのオペレーティングシステム。

## 備考

-cv+ が指定されると、次のいずれかのキャッシュウォーミングアクティビティが発生した場合に、データベースサーバメッセージウィンドウにメッセージが表示されます。

- 要求されたページの収集が開始または停止する (-cc サーバオプションで制御)
- ページの再ロードが開始または停止する (-cr サーバオプションで制御)

### 例

次のコマンドは、データベースページの収集とページのロードをオンにして `mydatabase.db` というデータベースを起動し、これらのアクティビティに関するメッセージをデータベースサーバメッセージウィンドウに表示します。

```
dbsrv17 -cc+ -cr+ -cv+ mydatabase.db
```

## 関連情報

[キャッシュウォーミング \[1375 ページ\]](#)

[-c データベースサーバオプション \[385 ページ\]](#)

[-ca データベースサーバオプション \[387 ページ\]](#)

[-cc データベースサーバオプション \[388 ページ\]](#)

[-ch データベースサーバオプション \[392 ページ\]](#)

[-cl データベースサーバオプション \[395 ページ\]](#)

[-cr データベースサーバオプション \[397 ページ\]](#)

[-cs データベースサーバオプション \[398 ページ\]](#)

### 1.4.1.16 -dt データベースサーバオプション

テンポラリファイルを保存するディレクトリを指定します。

#### 構文

```
dbsrv17 -dt temp-file-dir ...
```

## 適用対象

すべてのオペレーティングシステム (UNIX 上の共有メモリ接続を除く)

## 備考

データベースサーバでは、次の 2 種類のテンポラリファイルが作成されます。

データベースサーバ関連のテンポラリファイル (すべてのプラットフォームで作成)

-dt オプションでは、データベースサーバ関連のテンポラリファイルを保存するディレクトリを指定できます。このオプションを指定しないでデータベースサーバを起動すると、データベースサーバは以下の環境変数をこのとおりの順序で調べ、テンポラリファイルの保存先とするディレクトリを決定します。

### Windows/UNIX

1. SATMP
2. TMP
3. TMPDIR
4. TEMP

これらの環境変数がいずれも定義されていない場合、テンポラリファイルは、Windows の場合は現在のディレクトリ、UNIX の場合は /tmp ディレクトリに作成されます。

通信関連のテンポラリファイル (クライアントとデータベースサーバの両方に対して UNIX のみで作成)

UNIX では、通信関連のテンポラリファイルは -dt で指定されたディレクトリには保存されません。その代わりに、データベースサーバは以下の環境変数をこのままの順序で調べ、テンポラリファイルの保存先にするディレクトリを決定します。

### Unix

1. SATMP
2. TMP
3. TMPDIR
4. TEMP

これらの環境変数がいずれも定義されていない場合、テンポラリファイルは UNIX では /tmp ディレクトリに作成されます。

UNIX では、共有メモリを介して接続する場合、クライアントとデータベースサーバの両方で SATMP を同じ値に設定する必要があります。

## 例

データベースサーバ関連のテンポラリファイルの場所を特定するには、DB\_PROPERTY システム関数で TempFileName プロパティを使用します。

```
SELECT DB_PROPERTY ( 'TempFileName' );
```

## 関連情報

[データベースファイルのタイプ \[268 ページ\]](#)

[SATMP 環境変数 \[553 ページ\]](#)

[temp\\_space\\_limit\\_check オプション \[807 ページ\]](#)

[トラブルシューティング: データベースサーバの起動 \[974 ページ\]](#)

### 1.4.1.17 -ec データベースサーバオプション

トランスポートレイヤセキュリティまたは単純難読化を使用して、すべてのクライアントとの間で転送される通信プロトコルパケット (DBLIB や ODBC など) をエンコードします。TDS 接続はこのオプションを指定してもエンコードされません。

#### 構文

```
dbsrv17 -ec encoding-option [,...]
```

Specify at least one of the supported options in a comma-separated list.

encoding-option :

```
{ NONE  
  | SIMPLE  
  | TLS (  
    [ FIPS={ ON | OFF } ; ]  
    IDENTITY=server-identity-filename;  
    IDENTITY_PASSWORD=password  
    ALLOW_EXPIRED_CERTS={ ON | OFF } )
```

UNIX では、-ec オプションの値を指定するときに二重引用符が必要です。

```
dbsrv17 -ec "encoding-option [,...]"
```

#### 指定可能な値

##### NONE

データベースサーバは、暗号されていない接続を受け入れます。リモート接続の場合もありますし、共有メモリを使用したローカル接続の場合もあります。

##### SIMPLE

データベースサーバは、単純難読化を使用してエンコードされた接続を受け入れます。データベースサーバは、エンコードされていない接続も受け入れ、その接続を単純難読化に格上げします。

SIMPLE と NONE の両方のパラメータが指定された場合、データベースサーバはエンコードされていない接続を受け入れますが、その接続を格上げしません。

##### TLS

データベースサーバは、トランスポートレイヤセキュリティの RSA 暗号化アルゴリズムを使用して暗号化された接続を受け入れます。TLS パラメータは次の引数を受け取ります。

## FIPS

FIPS 認定の TLS の場合は、FIPS=ON と指定します。FIPS 認定の TLS は別のライブラリを使用しますが、認定されていない TLS と互換性があります。

FIPS 認定 RSA 暗号化には別途ライセンスが必要です。

**IDENTITY=server-identity-filename**

サーバ ID 証明書のパスとファイル名を指定します。RSA 暗号化アルゴリズムを使用して証明書を生成してください。

**IDENTITY\_PASSWORD=password**

ID ファイルに含まれるプライベートキーのパスワードです。このパスワードは、サーバ ID 証明書の作成時に定義されたものです。

**ALLOW\_EXPIRED\_CERTS**

ON に設定すると、データベースサーバは、期限切れのファイルまたはまだ有効でないファイルを特定して受け入れます。デフォルトは OFF です。

## デフォルト

デフォルトは、NONE, SIMPLE です。

SIMPLE と NONE の両方のパラメータが指定された場合、データベースサーバは、単純難読化を使用してエンコードされた接続を受け入れます。また、エンコードされていない接続も受け入れますが、その接続を格上げしません。

## 適用対象

すべてのオペレーティングシステムに適用されます。

## 備考

このオプションは、クライアントアプリケーションとデータベースサーバ間の通信パケットを安全化する場合に使用します。デフォルトでは、通信パケットは暗号化されないため、セキュリティに潜在的なリスクがあります。ネットワークパケットのセキュリティが心配な場合は、-ec オプションを使用します。

-ec オプションを指定すると、データベースサーバは、指定された暗号化タイプの 1 つを使用して暗号化される接続のみを受け入れます。カンマ区切りリストで、少なくとも 1 つのサポートされているパラメータを指定してください。

単純難読化の場合、サーバの検証や強力な暗号化などの TLS の機能は提供されません。

-ec NONE オプションを指定すると、リモートおよびローカル接続の両方で、暗号化されていない接続を受け入れるようデータベースサーバに指示することになります。データベースサーバがリモート接続ではエンコードされた接続のみを受け入れ、ローカル (共有メモリ) 接続ではエンコードされていない接続も受け入れるようにするには、-es データベースサーバオプションとともに -ec TLS または -ec SIMPLE を指定する必要があります。

クライアントとデータベースサーバで暗号化の設定が一致していることが必要です。設定が異なっていると、次の場合を除き、接続は失敗します。

- データベースサーバに対して `-ec SIMPLE` を指定し、`-ec NONE` を指定しなかった場合、Encryption 接続パラメータを指定しない接続および `Encryption=NONE` を指定する接続は許可され、単純難読化が使用されるよう自動的に格上げされます。
- 認定されていない TLS 暗号化をデータベースサーバ側で指定し、FIPS 認定の TLS 暗号化をクライアント側で指定している場合、またはその逆の場合、接続は成功します。この場合、Encryption 接続プロパティはデータベースサーバ側で指定された値を返します。

`-ec` オプションは、TDS 通信プロトコル接続には適用されません。TDS パケットは暗号化されません。TDS プロトコルを介した接続は、jConnect を使用する Java アプリケーションを含みますが、`-ec` オプションの使用に関係なく常に受け入れられ、暗号化されることはありません。暗号化されていない TDS 接続を受け入れないようにするためには、TDS プロトコルオプションを `NO` に設定する必要があります。

暗号化がパフォーマンスに及ぼす影響はごくわずかです。

`dbrsa17.dll` (UNIX では `libdbrsa17_r.so`) ファイルには、暗号化と復号化に使用される TLS コードがあります。  
`dbfips17.dll` (Linux では `libdbfips17_r.so`) ファイルには、TLS アルゴリズムの FIPS 認定バージョンがあります。データベースサーバに接続するときに、適切なファイルが見つからなかったり、エラーが発生したりすると、データベースサーバメッセージウィンドウにメッセージが表示されます。指定されたタイプの暗号化を開始できない場合、データベースサーバは起動しません。

## 例

次の例は、暗号化されない接続と単純難読化を使用する接続を許可します。

Windows の場合:

```
dbsrv17 -ec NONE,SIMPLE -x tcpip mydemo.db
```

UNIX の場合:

```
dbsrv17 -ec "NONE,SIMPLE" -x tcpip mydemo.db
```

次の例では、TLS 暗号化と ID 証明書 `rsaserver.id` を使用するデータベースサーバを起動します。

Windows の場合:

```
dbsrv17 -ec TLS (IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip mydemo.db
```

UNIX の場合:

```
dbsrv17 -ec "TLS (IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip mydemo.db
```

次の例では、FIPS 認定 TLS 暗号アルゴリズムと ID 証明書 `rsaserver.id` を使用するデータベースサーバを起動します。`rsaserver.id` が期限切れの場合でも、データベースサーバによって受け入れられます。

```
dbsrv17 -ec  
TLS (FIPS=ON;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test;ALLOW_EXPIRED_CERTS=ON) -  
x tcpip mydemo.db
```

次の例では、同じサーバに接続し、テキスト `rsa_tls_fips` をコマンドプロンプトに表示します。

```
dbisql -c
"UID=DBA;PWD=passwd;ENC=TLS (TRUSTED_CERTIFICATE=rsaroot.crt;SKIP_CERTIFICATE_NAME_
CHECK=ON);Server=mydemo;LINKS=TCPIP" select connection_property('encryption')
```

### i 注記

ここで参照されているサンプル証明書は、`samples¥certificates` フォルダにインストールされています。

## 関連情報

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

[トランスポートレイヤセキュリティを指定したデータベースサーバ \[1654 ページ\]](#)

[デジタル証明書 \[1644 ページ\]](#)

[TDS プロトコルオプション \(サーバ側のみ\) \[231 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

[-ek データベースオプション \[527 ページ\]](#)

[-ep データベースサーバオプション \[406 ページ\]](#)

[-es データベースサーバオプション \[407 ページ\]](#)

[DatabaseKey \(DBKEY\) 接続パラメータ \[67 ページ\]](#)

[FIPS プロトコルオプション \[195 ページ\]](#)

[-fips データベースサーバオプション \[411 ページ\]](#)

[allow\\_expired\\_certs プロトコルオプション \[181 ページ\]](#)

### 1.4.1.18 -edi データベースサーバオプション

データベース独立性をオンにします。

#### 構文

```
dbsrv17 -edi ...
```

## デフォルト

デフォルトでは、データベース独立性の有効化はオフになっています。

## 適用対象

すべてのオペレーティングシステムに適用されます。

## 備考

このオプションを指定すると、すべてのデータベースがデータベースサーバで実行されている唯一のデータベースであるかのように動作します。1つのデータベースサーバへの接続が、同じデータベースサーバ上で実行されている他のデータベースのプロパティや接続情報のクエリを実行することはできません。接続が、同じデータベースサーバ上で実行されている他のデータベースを停止したり起動したりすることはできません。

### i 注記

現在の接続のデータベース独立性を一時的にオフにするために `database_isolation` セキュリティ機能を使用します。

データベース独立性をオンにすると、アプリケーションは、該当するユーティリティユーザ ID とパスワードを使用してユーティリティデータベースに引き続き接続し、ユーティリティアクションを実行することができます。ただし、ユーティリティデータベースへの接続中のデータベースの起動や停止が制限されます。アプリケーションはデータベースの起動や停止に `database_isolation` セキュリティ機能を使用する必要があります。

## 関連情報

[-sf データベースサーバオプション \[471 ページ\]](#)

### 1.4.1.19 -ep データベースサーバオプション

強い暗号で保護されたデータベースを起動する際、ユーザに暗号化キーの入力を求めます。

#### 構文

```
dbsrv17 -ep ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

-ep オプションを指定すると、データベースサーバは、コマンドラインから起動されるときに暗号化キーの入力が必要なデータベースについて、ユーザが暗号化キーを入力するためのウィンドウを表示します。このサーバオプションでは、クリアテキストで暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。

データベースサーバでこのオプションを指定した場合、次の条件がすべて該当する場合にユーザは暗号化キーの入力を要求されます。

- -ep オプションが指定されている
- データベースを起動するために暗号化キーが必要である
- データベースサーバが Windows のサービスまたはデーモンではない (UNIX)

クライアントアプリケーションとデータベースサーバ間の通信パケットの安全性を保つために、-ec サーバオプションとトランスポートレイヤセキュリティを使用します。

### 例

myencrypted.db データベースが起動すると、ユーザは暗号化キーの入力を要求されます。

```
dbsrv17 -ep -x tcpip myencrypted.db
```

## 関連情報

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

[トランスポートレイヤセキュリティを指定したデータベースサーバ \[1654 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

[-ek データベースオプション \[527 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

[DatabaseKey \(DBKEY\) 接続パラメータ \[67 ページ\]](#)

### 1.4.1.20 -es データベースサーバオプション

共有メモリを経由した暗号化されていない接続を許可します。

#### 構文

```
dbsrv17 -ec encryption-options -es ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、`-ec` オプションとともに指定された場合のみ有効です。`-es` オプションは、共有メモリを経由した、暗号化されていない Command Sequence ローカル接続を受け入れるよう、データベースサーバに指示します。これに対して、`-ec NONE` オプションを指定すると、リモートおよびローカル接続の両方で、暗号化されていない Command Sequence 接続を受け入れるようデータベースサーバに指示することになります。

TCP/IP を介した接続では、`-ec` オプションで指定された暗号化タイプを使用する必要があります。`-es` オプションは、リモートクライアントからのデータベースアクセスには暗号化された接続を使用し、パフォーマンス上の理由から、ローカルコンピュータからのデータベースアクセスには暗号化されていない接続を使用できるようにしたい場合に便利です。

### 例

- たとえば、SQL Anywhere コックピットは共有メモリ接続を必要とします。TLS で暗号化されたデータベースサーバ上でコックピットを実行するには、`-ec TLS(...)` `-es` を使用してデータベースサーバを起動します。
- 次の例では、単純難読化を使用する接続と、共有メモリを経由した暗号化されない接続を指定します。

```
dbsrv17 -ec SIMPLE -es -x tcpip c:¥mydemo.db
```

## 関連情報

[トランスポートレイヤセキュリティを指定したデータベースサーバ \[1654 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

### 1.4.1.21 -f データベースサーバオプション

トランザクションログが失われた後で、データベースサーバを強制的に起動します。

### 構文

```
dbsrv17 -f ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

### 警告

このオプションは、リカバリの場合にだけ使用されます。

トランザクションログが存在しない場合、データベースサーバはデータベースのチェックポイントリカバリを実行して終了します。実行は継続されません。その後は、`-f` オプションを指定せずにデータベースサーバを再起動し、通常の作業を再開できます。

データベースと同じディレクトリにトランザクションログが存在する場合、データベースサーバはチェックポイントリカバリとトランザクションログを使ったリカバリを実行して終了します。実行は継続されません。その後は、`-f` オプションを指定せずにデータベースサーバを再起動し、通常の作業を再開できます。

サーバの起動時にキャッシュサイズを指定すると、リカバリ時間を短縮できます。

### 例

次のコマンドを入力すると、データベースサーバがデータベース `mydatabase.db` を起動し、リカバリを実行します。

```
dbsrv17 mydatabase.db -f
```

## 関連情報

[特殊モード \[327 ページ\]](#)

[データベースのバックアップとリカバリ \[894 ページ\]](#)

### 1.4.1.22 `-fc` データベースサーバオプション

ファイルシステムフルのコールバック関数がある DLL (UNIX では共有オブジェクト) のファイル名を指定します。

### 構文

```
dbsrv17 -fc filename ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションを使用すると、ファイルシステムフルの状態が発生したときに、ユーザに通知して、必要に応じて修正措置をとることができます。-fc オプションを使用した場合、起動時にデータベースサーバは指定の DLL の読み込みとコールバック関数のエントリポイントの解決を試みます。DLL とエントリポイントの両方を検出できない場合、データベースサーバはエラーを返し、停止します。ユーザが提供する DLL は、コールバックを使用して指定のバッチファイル (UNIX の場合はシェルスクリプト) を呼び出し、診断または修正措置をとることができます。また、コールバック関数自体でもそのようなアクションを実行できます。

`%SQLANYSAMPI7%`¥SQLAnywhere¥DiskFull にディスクフルのコールバック関数のサンプルがあります。

データベースサーバは、他の DLL やファイルを検索する場合と同じロケーションでコールバック関数 DLL を検索します。

データベースサーバは、ディスクフルの状態を検出すると、コールバック関数 (指定されている場合) を呼び出し、それに以下の情報を渡します。

- 条件がトリガされた DB 領域の名前
- 失敗した操作からのオペレーティングシステム固有のエラーコード

xp\_out\_of\_disk からのリターンコードにより、原因となった操作を中止するか再実行するかが決定されます。0 以外の値が返された場合は、操作は中止され、それ以外の場合は操作が再実行されます。0 が返され、ファイルシステムの操作が失敗するたびに、コールバック関数が繰り返し呼び出されます。

Microsoft Windows プラットフォームでは、データベースサーバの起動時に、データベースサーバメッセージウィンドウが表示される設定で (-qi も -qw も指定されていない)、コールバック DLL が指定されていない場合は、ディスクフルの状態が発生するとウィンドウが表示されます。このウィンドウには、DB 領域名とエラーコードが含まれているので、ユーザはディスクフルの状態の原因となった操作を再実行するか中止するかを決定できます。

その他すべてのオペレーティングシステムでは、-fc が指定されていない場合にディスクフルの状態になると、致命的なエラーが発生します。

データベースファイル、トランザクションログファイル、テンポラリファイルなどを格納しているデバイスの使用可能なディスク領域を管理するシステムイベントを作成して、ディスク領域の不足を管理者に警告できます。

### 例

データベースサーバは、起動時に `diskfull.dll` DLL をロードしようとします。

```
dbsrv17 -fc diskfull.dll -n my_server
```

## 関連情報

[SQL Anywhere のファイルを配置する方法 \[562 ページ\]](#)

[システムイベント \[954 ページ\]](#)

[max\\_temp\\_space オプション \[734 ページ\]](#)

[temp\\_space\\_limit\\_check オプション \[807 ページ\]](#)

### 1.4.1.23 -fips データベースサーバオプション

データベースや通信の強力な暗号化を行うため、データベースサーバが FIPS 認定の暗号化を使用するよう設定します。

#### 構文

```
dbsrv17 -fips ...
```

#### 適用対象

Windows および Linux。

#### 備考

このオプションは、データベースの強力な暗号化、クライアント/サーバのトランスポートレイヤセキュリティ、Web サービスのトランスポートレイヤセキュリティに適用されます。

FIPS 認定の暗号化には別途ライセンスが必要です。

データベースの強力な暗号化では、-fips オプションを指定すると、CREATE DATABASE 文の ALGORITHM 句で AES と AES256 が指定されていても、新しいデータベースではそれらに相当する FIPS 認定の暗号化が使用されます。

-fips を指定してデータベースサーバを起動した場合、AES、AES256、AES\_FIPS、AES256\_FIPS のいずれかの暗号化方式で暗号化されたデータベースを実行できますが、単純難読化方式でエンコードされたデータベースは実行できません。-fips が指定されているときは、暗号化されていないデータベースをサーバ上で開始することもできます。

AES\_FIPS または AES256\_FIPS で暗号化したデータベースを実行するために使用するコンピュータには、SQL Anywhere セキュリティオプションをインストールしてください。

トランスポートレイヤセキュリティでは、-fips オプションを指定すると、RSA が指定されていても、サーバは FIPS 認定のアルゴリズムを使用します。

Web サービス用のトランスポートレイヤセキュリティでは、-fips オプションを指定すると、HTTPS\_FIPS の代わりに HTTPS が指定されていても、サーバは FIPS 認定の HTTPS FIPS を使用します。

-fips を指定すると、ENCRYPT 関数と HASH 関数では FIPS 認定の暗号化アルゴリズムが使用され、パスワードハッシングではアルゴリズムとして SHA-256 ではなく SHA-256 FIPS が使用されます。

-fips データベースサーバオプションを指定してから、xp\_startsmtp システムプロシージャを使用して電子メールセッションを開始すると、データベースサーバは常に PLAIN 認証を使用して、SMTP サーバに対する ID とパスワードを認証します。-fips データベースサーバオプションは CRAM\_MD5 認証をサポートしません。

## 関連情報

[単純難読化と強力な暗号化 \[1615 ページ\]](#)

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

[SQL Anywhere Web サービス暗号化 \[1658 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

### 1.4.1.24 -ga データベースサーバオプション

最後の非 HTTP クライアント接続を切断した後、データベースをアンロードします。

#### 構文

```
dbsrv17 -ga ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

ネットワークサーバでこのオプションを指定すると、最後の非 HTTP クライアント接続が切断された後に各データベースがアンロードされます。最後の非 HTTP クライアント接続が切断された後に各データベースをアンロードし、最後のデータベースが停止したときにはサーバも停止します。

データベースへの唯一の接続が HTTP 接続で、自動的に停止するようにデータベースが設定されている場合、HTTP 接続が切断したときにデータベースはアンロードされません。また、-ga オプションを指定したデータベースに HTTP 接続と Command Sequence 接続または TDS 接続がある場合は、最後の Command Sequence 接続または TDS 接続が切断したときにデータベースが自動的に停止します。このときに HTTP 接続がまだあった場合は切断されます。

## 関連情報

[AutoStop \(ASTOP\) 接続パラメータ \[51 ページ\]](#)

### 1.4.1.25 -gb データベースサーバオプション

サーバプロセスの優先度クラスを設定します。

#### 構文

```
(Windows only) dbsrv17 -gb { IDLE | NORMAL | HIGH | MAXIMUM } ...
```

```
(Unix only) dbsrv17 -gb level ...
```

#### 指定可能な値

##### Windows

Windows の場合、一般的に使用される設定は、NORMAL および HIGH です。値 IDLE は万全を期すために提供されています。値 MAXIMUM はコンピュータの実行を妨げる可能性があります。

##### UNIX

UNIX の場合、`level` は -20 ~ 19 の整数です。UNIX のデフォルト値は、親プロセスの `nice` 値と同じです。`level` 値を下げると、より適切なスケジューリング優先度になります。`-gb` オプションには、`nice` 値の設定に関するすべての制限が適用されます。たとえば、ほとんどの UNIX プラットフォームの場合、プロセスの優先度レベルを下げる (たとえば、0 から -1 に変更する) ことができるのは root ユーザだけです。

#### 適用対象

すべてのオペレーティングシステム。

### 1.4.1.26 -gc データベースサーバオプション

チェックポイント間の最大間隔を設定します。

#### 構文

```
dbsrv17 -gc minutes ...
```

## デフォルト

60 分

## 指定可能な値

### minutes

デフォルト値は、checkpoint\_time データベースオプションの設定値 (デフォルトは 60 分) です。値 0 を入力すると、デフォルト値の 60 が使用されます。

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、各データベースでチェックポイントを行わずに、データベースサーバを実行する期待最大時間を分数で設定するために使用します。

通常、チェックポイントは、指定する時間より頻繁に発生します。

## 関連情報

[チェックポイントログ \[294 ページ\]](#)

[データベースサーバがチェックポイントのタイミングを決定する方法 \[293 ページ\]](#)

[checkpoint\\_time オプション \[667 ページ\]](#)

## 1.4.1.27 -gd データベースサーバオプション

実行中のデータベースサーバ上でデータベースを起動または停止するために必要な権限を設定します。

### 構文

```
dbsrv17 -gd { DBA | ALL | NONE } ...
```

## 指定可能な値

### DBA

SERVER OPERATOR システム権限を持つユーザのみがデータベースを起動または停止できます。

### ALL

すべてのユーザがデータベースを起動または停止できます。リモートクライアントからアクセスできるネットワークサーバへの設定はおすすめしません。

### NONE

データベースサーバ自体が起動しているか停止している場合を除き、データベースの開始と停止は許可されません。

## デフォルト

ネットワークデータベースサーバの場合、デフォルトの設定は **DBA** です。大文字と小文字の両方の構文を使用できます。

パーソナルデータベースサーバの場合、デフォルトの設定は **ALL** です。

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、ユーザが新しいデータベースファイルをデータベースサーバにロードするために、または実行中のデータベースサーバでデータベースを停止するために必要な権限のレベルを指定します。

オプションを DBA に設定した場合、クライアントアプリケーションは、データベースを開始または停止させるために同一のサーバ上で動作している別のデータベースへの既存の接続を使用する必要があります。DatabaseFile (DBF) 接続パラメータを使用して、まだ動作中でないデータベースを起動することはできません。

StartDBPermission サーバプロパティを使用して、-gd オプションの設定を取得できます。

```
SELECT PROPERTY ( 'StartDBPermission' );
```

データベースサーバを停止するための権限は、-gk オプションで指定します。

### 例

ネットワークデータベースサーバで -gd オプションを使用する手順は、次のとおりです。

1. ネットワークデータベースサーバを起動します。

```
dbsrv17 -su passwd -gd DBA -n my_server
```

2. Interactive SQL からユーティリティデータベースに接続します。

```
dbisql -c "UID=DBA;PWD=passwd;DBN=utility_db"
```

3. データベースを起動します。

```
START DATABASE 'demo.db';
```

4. データベースに接続します。

```
CONNECT USING 'DBN=demo;UID=DBA;PWD=sql';
```

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[-gk データベースサーバオプション \[417 ページ\]](#)

### 1.4.1.28 -ge データベースサーバオプション

外部関数のスタックサイズを設定します。

#### 構文

```
dbsrv17 -ge integer ...
```

#### デフォルト

32 KB

#### 適用対象

Windows

#### 備考

外部関数を実行するスレッドのスタックサイズをバイト数で設定します。

## 関連情報

[スレッド動作 \[333 ページ\]](#)

### 1.4.1.29 -gf データベースサーバオプション

サーバがトリガを起動できないようにします。

#### 構文

```
dbsrv17 -gf ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

-gf サーバオプションは、参照整合性トリガ (カスケード更新および削除など) を含めて、トリガの起動を無効にするようサーバに指示します。

## 関連情報

[fire\\_triggers オプション \[702 ページ\]](#)

### 1.4.1.30 -gk データベースサーバオプション

データベースサーバの停止に必要な権限を設定します。

#### 構文

```
dbsrv17 -gk { DBA | ALL | NONE } ...
```

## 指定可能な値

### DBA

SERVER OPERATOR システム権限を持つユーザのみがデータベースサーバを停止できます。

この値はパーソナルサーバの場合のデフォルトです。

### ALL

データベースサーバを停止するのに権限が必要ありません。

この値はパーソナルサーバの場合のデフォルトです。

### NONE

データベースサーバを停止できません。

## 適用対象

すべてのオペレーティングシステム。

## 備考

-gd データベースサーバオプションは、dbstop ユーティリティに適用されるほか、次の文にも適用されます。

- ALTER DATABASE `dbname` FORCE START 文
- STOP DATABASE 文

## 関連情報

[サーバ停止ユーティリティ \(dbstop\) \[1163 ページ\]](#)

### 1.4.1.31 -gl データベースサーバオプション

LOAD TABLE 文を使用したデータのロードに必要な権限と、UNLOAD 文または UNLOAD TABLE 文を使用したデータのアンロードに必要な権限を設定します。

#### 構文

```
dbsrv17 -gl { DBA | ALL | NONE } ...
```

## 指定可能な値

### DBA

データのアンロードには、SELECT ANY TABLE システム権限が必要です。

データのロードには、LOAD ANY TABLE、ALTER ANY TABLE、または ALTER ANY OBJECT システム権限が必要です。

### ALL

データのアンロードには、テーブルの所有者であるか、テーブルに対する SELECT 権限を持っているか、または SELECT ANY TABLE システム権限を持っていることが必要です。

データのロードには、次のいずれかの条件に該当する必要があります。

- テーブルの所有者である。
- テーブルに対する LOAD 権限を持っている。
- LOAD ANY TABLE システム権限を持っている。
- ALTER ANY TABLE システム権限を持っている。
- ALTER ANY OBJECT システム権限を持っている。

### NONE

データのロードやアンロードはできません。

## デフォルト

DBA はネットワークサーバのデフォルト値です。

Windows では、パーソナルデータベースサーバのデフォルト設定は **ALL** です。パーソナルデータベースサーバは現在のコンピュータで実行されるため、ユーザはすでにファイルシステムにアクセス可能だからです。

UNIX では、パーソナルデータベースサーバとネットワークデータベースサーバのどちらの場合も、デフォルト設定は **DBA** です。

## 適用対象

すべてのオペレーティングシステム。

## 備考

UNLOAD TABLE や UNLOAD 文は、データベースサーバコンピュータのファイルにデータを配置します。LOAD TABLE 文は、データベースサーバコンピュータからファイルを読み込みます。

これらの文を使用してファイルシステムへのアクセスを制御する場合、-gl サーバオプションを使用して、これらの文を使用するのに必要な権限を制御できます。

大文字と小文字の両方の構文を使用できます。

## 1.4.1.32 -gm データベースサーバオプション

データベースサーバへの同時接続の数を制限します。

### 構文

```
dbsrv17 -gm integer ...
```

### デフォルト

パーソナルデータベースサーバ (dbeng17) の接続制限およびデフォルト値は 10 個の接続です。そのうち、3 個の接続のみで標準接続を受け入れることができます。

ネットワークデータベースサーバのデフォルト値はライセンス契約に応じて異なります。

### 適用対象

すべてのオペレーティングシステム。

### 備考

データベースサーバの接続制限を定義します。

ライセンス契約で許可される数より大きい値、またはメモリ制約を超えた値をここで設定した場合、その値は無効です。

システム権限 DROP CONNECTION または SERVER OPERATOR を持つユーザがデータベースサーバに接続して他の接続を削除できるように、データサーバは接続制限より 1 つ余分に接続を許可します。

データベースサーバの HTTP/HTTPS 接続の制限に達すると、データベースサーバは HTTP/HTTPS 接続試行をキューに入れます。HTTP/HTTPS キューに接続が入っている間、HTTP/HTTPS 接続の代わりに標準接続を行うことはできません。標準接続用にデータベース接続の一部が予約されている場合以外は、標準接続を行うとしているユーザは無限に待機することとなります。データベースの接続の一部を標準接続用に予約するには、reserved\_connections データベースオプションを指定します。

MaxConnections プロトコルオプションを指定することで、データベースサーバに適用する HTTP/HTTPS 接続の最大数を制限することもできます。

## 関連情報

[データベースおよびデータベースサーバに接続できる限界数 \[237 ページ\]](#)

[データベース接続の制限 \[239 ページ\]](#)

[max\\_connections オプション \[725 ページ\]](#)

[MaxConnections \(MAXCONN\) プロトコルオプション \[216 ページ\]](#)

### 1.4.1.33 -gn データベースサーバオプション

データベースサーバのマルチプログラミングレベルを設定します。

#### 構文

```
dbsrv17 -gn integer ...
```

#### デフォルト

20 個のアクティブなタスク

#### 適用対象

すべてのオペレーティングシステム。

#### 備考

-gn サーバオプションは、データベースサーバの初期マルチプログラミングレベルを設定します。

デフォルトでは、サーバの負荷に応じて、マルチプログラミングレベルがネットワークデータベースサーバで自動的に調整されます。ネットワークデータベースサーバの起動後に、sa\_server\_option システムプロシージャと CurrentMultiprogrammingLevel パラメータを使用して、マルチプログラミングの設定を手動で上書きできます。

#### i 注記

大きな integer 値を設定することによって、32 ビットプラットフォームで利用可能な最大キャッシュサイズを減らすことができます。

## 関連情報

[スレッド \[330 ページ\]](#)

[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[-gna データベースサーバオプション \[422 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gni データベースサーバオプション \[424 ページ\]](#)

### 1.4.1.34 -gna データベースサーバオプション

データベースサーバのマルチプログラミングレベルの自動チューニングを制御します。

#### 構文

```
dbsrv17 -gna { 0 | 1 }
```

#### デフォルト

1 (有効)

#### 適用対象

すべてのオペレーティングシステム。

ネットワークデータベースサーバ (dbsrv17) のみ。

#### 備考

このオプションは、データベースサーバのマルチプログラミングレベルの動的チューニングを有効または無効にします。sa\_server\_option システムプロシージャを使用すると、データベースサーバの動作中にマルチプログラミングレベルを手動で変更できます。

デフォルトでは、データベースサーバのマルチプログラミングレベルの自動チューニングは有効になっています。

## 関連情報

[スレッド \[330 ページ\]](#)

[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[max\\_query\\_tasks オプション \[731 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gnl データベースサーバオプション \[424 ページ\]](#)

### 1.4.1.35 -gnh データベースサーバオプション

最大マルチプログラミングレベル (ネットワークデータベースサーバが同時に実行できるタスクの最大数) を設定します。

#### 構文

```
dbsrv17 -gnh integer ...
```

#### デフォルト

-gn データベースサーバオプション値の 4 倍

#### 適用対象

すべてのオペレーティングシステム。

ネットワークデータベースサーバ (dbsrv17) のみ。

#### 備考

このオプションは、ネットワークデータベースサーバの最大マルチプログラミングレベルを設定します。ネットワークデータベースサーバが同時に実行できるタスク (ユーザ要求とシステム要求) の最大数を制限します。ネットワークデータベースサーバが最大数を超える要求を受け取った場合、新しい要求は実行中のタスクが完了するまで待つこととなります。同時に実行できるアクティブタスクの数は、ネットワークデータベースサーバのスレッドの数と、使用されている論理プロセッサの数によって異なります。

未スケジュールの要求とアクティブな要求を加算した総数は、-gm ネットワークデータベースサーバオプションで制限されます。これにより、ネットワークデータベースサーバへの接続数を制限します。

-gnh 値の設定が大きすぎると、システムのアドレス領域の大部分がスタック領域に消費されるため、エラーになる場合があります。

ネットワークデータベースサーバのカーネルは、スケジューリングの単位としてタスクを使用します。ユーザ要求を実行するには、最低 1 つのタスクが必要です。ただし、1 つの要求に伴って追加のタスクがスケジューリングされることがあります。たとえば、外部のプロシージャや関数 (Java、Perl など) の実行を伴う要求では、ネットワークデータベースサーバに対してデータベース要求が発行されます。

クエリ内並列処理が関連する場合は、同時に実行される各アクセスプランコンポーネントがタスクになります。これらのタスクは、別個の要求であるかのように -gnh 制限で考慮されます。ただし、クエリ内並列処理で作成されたタスクは、アクティブな要求とアクティブでない要求の数を追跡するデータベースプロパティには反映されません。

### 警告

-gss によって指定したサイズのスタックが、各データベースサーバワーカに割り付けられます。

最大ワーカ数はネットワークサーバの -gnh オプションで指定します。

パーソナルサーバの場合、最大ワーカ数は -gn オプションで指定します。

-gss に大きい値を設定し、かつワーカの最大数に大きな値を設定すると、データベースサーバが起動しないか、キャッシュサイズが大幅に制限される場合があります。たとえば、ネットワークデータベースサーバの起動時に -gss オプションで 16 M を指定し、-gnh オプションで 100 を指定すると、スタック用に 1.6 GB のサーバアドレス領域が割り当てられます。

## 関連情報

[スレッド \[330 ページ\]](#)

[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[max\\_query\\_tasks オプション \[731 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gna データベースサーバオプション \[422 ページ\]](#)

[-gnl データベースサーバオプション \[424 ページ\]](#)

[-gss データベースサーバオプション \[429 ページ\]](#)

### 1.4.1.36 -gnl データベースサーバオプション

データベースサーバのマルチプログラミングレベルの最小値を設定します。

#### 構文

```
dbsrv17 -gnl integer ...
```

## デフォルト

-gtc サーバオプション値とコンピュータの論理 CPU 数の小さい方の値。

## 適用対象

すべてのオペレーティングシステム。

ネットワークデータベースサーバ (dbsrv17) のみ。

## 備考

-gnl の値を設定すると、現在のマルチプログラミングレベルをこの値より低く設定できないようになります。負荷の高い操作の後にアイドル期間が続く場合は、-gnl の値を大きくすると、負荷の高い処理が再度発生したときにデータベースサーバのパフォーマンスを向上するのに効果的です。

## 関連情報

[スレッド \[330 ページ\]](#)

[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

[max\\_query\\_tasks オプション \[731 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gna データベースサーバオプション \[422 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

### 1.4.1.37 -gns データベースサーバオプション

データベースサーバメッセージログにマルチプログラミングレベルの自動調整の統計を表示します。

#### 構文

```
dbsrv17 -gns ...
```

## デフォルト

統計は表示されません。

## 適用対象

すべてのオペレーティングシステム。

ネットワークデータベースサーバ (dbsrv17) のみ。

## 備考

マルチプログラミングレベルの統計は、-gns オプションが指定されている場合にのみデータベースサーバメッセージログに記録されます。

データベースサーバの起動後に、sa\_server\_option システムプロシージャを使用して、マルチプログラミングレベルの統計のレポート設定を変更できます。

マルチプログラミングレベルの統計が現在レポートされているかどうかを確認するには、次のクエリを実行します。

```
SELECT PROPERTY ( 'AutoMultiProgrammingLevelStatistics' );
```

マルチプログラミングレベルの統計は、次の例のように、データベースサーバメッセージログに記録されます。

```
MPL[curr=20 prev=20] concurrency[4] throughput[curr=2698 prev=3270]
```

上記の例の意味は次のとおりです。

統計情報	説明
curr=20	最新の時間間隔におけるワーカの数。
prev=20	一つ前の時間間隔におけるワーカの数。
concurrency[4]	最新の時間間隔における同時 (並列) 要求数の平均値。
throughput[curr=2698]	最新の時間間隔の間に処理された要求数。
throughput[prev=3270]	一つ前の時間間隔の間に処理された要求数。

## 関連情報

[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[データベースサーバのロギング \[322 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gna データベースサーバオプション \[422 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gni データベースサーバオプション \[424 ページ\]](#)

### 1.4.1.38 -gp データベースサーバオプション

許可されるデータベースページサイズの最大値を設定します。

#### 構文

```
dbsrv17 -gp { 2048 | 4096 | 8192 | 16384 | 32768 } ...
```

#### デフォルト

4096 (データベースをロードしないでデータベースサーバを起動した場合)。

#### 適用対象

すべてのオペレーティングシステム。

#### 備考

サーバのページサイズよりも大きいページサイズのデータベースファイルはロードできません。このオプションは、サーバのページサイズをバイト数で明示的に設定します。

デフォルトでは、サーバページサイズは、コマンドラインで指定されたデータベースの最大ページサイズと同じ大きさです。

2048 ページサイズは廃止予定です。

いずれのプラットフォームでも、このオプションを指定せずにデータベースがロードされていない状態でサーバを起動した場合、デフォルト値は 4096 になります。

#### 関連情報

[最大ページサイズの考慮事項 \[319 ページ\]](#)

## 1.4.1.39 -gr データベースサーバオプション

システム障害からのリカバリに要する最長時間を分数で設定します。

### 構文

```
dbsrv17 -gr minutes ...
```

### デフォルト

recovery\_time データベースオプションの設定 (デフォルトでは 2 分)。

### 適用対象

すべてのオペレーティングシステム。

### 備考

データベースサーバが複数のデータベースで実行されている場合、このオプションで書き換えられないかぎり、最初に起動されたデータベースで指定されているリカバリ時間が使用されます。

-gr オプションで指定した値は、データベースサーバにチェックポイントを実行する頻度を指示します。たとえば、-gr を 5 に設定した場合、データベースサーバは頻繁にチェックポイントを実行して、リカバリが 5 分以上かからないようにします。ただし、リカバリが必要な場合は、-gr で指定した時間より長くかかったとしても、完了するまで実行されます。

リカバリ時間には、データベースのリカバリ予想時間とチェックポイント予想時間が含まれます。

### 関連情報

[データベースサーバがチェックポイントのタイミングを決定する方法 \[293 ページ\]](#)

[recovery\\_time オプション \[768 ページ\]](#)

## 1.4.1.40 -gss データベースサーバオプション

データベースサーバでワーカごとのスタックサイズを設定します。

### 構文

```
dbsrv17 -gss { integer[ k | m ] } ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

-gss オプションによって、メモリが限られている環境で、データベースサーバのメモリ使用量を節約できます。

*size* には、使用するメモリ容量をバイト単位で指定します。キロバイトまたはメガバイトの単位を指定するには、それぞれ *k*、*m* を使用します。

### 警告

-gss によって指定したサイズのスタックが、各データベースサーバワーカに割り付けられます。最大ワーカ数はネットワークサーバの -gnh オプションで指定します。

パーソナルサーバの場合、最大ワーカ数は -gn オプションで指定します。

-gss に大きい値を設定し、かつワーカの最大数に大きな値を設定すると、データベースサーバが起動しないか、キャッシュサイズが大幅に制限される場合があります。たとえば、ネットワークデータベースサーバの起動時に -gss オプションで 16 M を指定し、-gnh オプションで 100 を指定すると、スタック用に 1.6 GB のサーバアドレス領域が割り当てられます。

ワーカごとのスタックサイズの最小値、最大値、デフォルトは次のとおりです。

プラットフォーム	最小	デフォルト	最大
UNIX (32 ビット)	512 KB	512 KB	4 MB
UNIX (64 ビット)	1 MB	1 MB	8 MB
Windows (32 ビット)	64 KB	1 MB	16 MB
Windows (64 ビット)	64 KB	4 MB	256 MB

### 関連情報

[スレッド \[330 ページ\]](#)

## 1.4.1.41 -gt データベースサーバオプション

使用できる物理プロセッサの最大数を設定します (ライセンスされたプロセッサの数を上限とする)。このオプションは、マルチプロセッサシステムでのみ役立ちます。

### 構文

```
dbsrv17 -gt num-processors ...
```

### 指定可能な値

#### num-processors

1 と、次に示す項目の最小値との間の整数で値を設定できます。

- コンピュータ上で利用可能な物理プロセッサの数
- サーバがライセンスを受けている CPU の最大数 (CPU がライセンスされている場合)

-gt オプションに範囲外の値を指定すると、最小または最大の整数値が設定されます。

パーソナルデータベースサーバ (dbeng17) では、サーバは -gt オプションの値として 1 を使用します。

### 適用対象

すべてのオペレーティングシステム。Windows、Linux、Solaris を除く、すべてのオペレーティングシステムに対する設定は無視されます。

### 備考

パーシットライセンスでは、ネットワークデータベースサーバは、コンピュータで使用可能なすべての CPU を使用します。CPU ベースのライセンスの場合、ネットワークデータベースサーバはライセンスを受けたプロセッサ数のみ使用可能です。ネットワークデータベースサーバが使用できる CPU 数は、SQL Anywhere のエディションによって制限される場合もあります。

パーソナルデータベースサーバは常に単一のプロセッサに限定されています。

物理プロセッサは、コンピュータの CPU であり、パッケージまたはダイと呼ばれることもあります。物理プロセッサがハイパースレッドをサポートする場合や、マルチプロセッサ (多くの場合、"マルチコアプロセッサ" と呼ばれます) として設定されている場合は、追加の論理プロセッサが存在します。オペレーティングシステムは、論理プロセッサ上でスレッドをスケジュールします。

-gt オプションの値を指定すると、データベースサーバはアフィニティマスクを調整し (ハードウェアプラットフォームでサポートされている場合)、指定した物理プロセッサの数のみを使用して実行するようにデータベースサーバを制限します。データベースサーバが  $n$  個のプロセッサのライセンスを受けている場合、デフォルトでは、サーバは  $n$  個の物理プロセッサにおいて、すべての論理プロセッサ (ハイパースレッドとコア) 上で実行します。-gtc オプションを使用すると、この動作にさらに制限を加えることができます。

-gt オプションは、-gta オプションとは一緒に使用できません。

## 関連情報

[スレッド \[330 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

[-gta データベースサーバオプション \[431 ページ\]](#)

### 1.4.1.42 -gta データベースサーバオプション

データベースサーバが使用できる論理プロセッサを設定します。

#### 構文

```
dbsrv17 -gta logical-processors-to-use,...
```

UNIX では、複数のパラメータを指定するときに引用符が必要です。

## 指定可能な値

### logical-processors-to-use

プロセッサの番号、範囲、またはその両方のカンマ区切りのリストを受け入れます。範囲の下限の終了ポイントを省略すると、ゼロと見なされます。範囲の上限の終了ポイントを省略すると、オペレーティングシステムが認識している最大の CPU であると見なされます。すべてのプロセッサを指定するには、`-` または `0-` と入力します。

## 適用対象

Windows および Linux。

## 備考

このオプションを設定することは、ProcessorAffinity システムプロシージャを実行することと同義になります。

-gta オプションを使用すると、データベースサーバが使用する論理プロセッサを指定できます。

次の場合には、指定されたすべての論理プロセッサをデータベースサーバが使用しない場合があります。

- 指定された論理プロセッサの中に、存在しないもの、またはオフラインのものがある場合。
- ライセンスで使用が許可されない場合。

-gta オプションで指定した論理プロセッサのセットがライセンスの制限を超えている場合は、-gta オプションで指定された論理プロセッサのうち、ライセンスの制限までの (ただし、制限を超えない) 最も小さい番号のものがサーバで使用されます。データベースサーバは、-gta オプションでリストされた順番に論理プロセッサを選択しません。サーバは、ライセンスの制限および指定した -gta オプションの範囲内で同時実行性を最大化しようとはしません。

-gta オプションで指定した論理プロセッサのセットと一致するオンラインプロセッサがない場合は、-gta オプションを指定していない場合と同様にサーバが動作し、プロセッサ 0 から、ライセンスされた数までの論理プロセッサを使用します。

-gta オプションを -gt または -gtc オプションと一緒に使用することはできません。

データベースサーバの実行中に、sa\_server\_option システムプロシージャの ProcessorAffinity オプションを使用して、データベースサーバが使用する論理プロセッサを変更できます。

### 例

次の例では、論理プロセッサ 3、5、6、7、または 9 の一部または全部を使用するようデータベースサーバに求めます。

```
dbsrv17 -gta 3,5-7,9
```

## 関連情報

[-gt データベースサーバオプション \[430 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

### 1.4.1.43 -gtc データベースサーバオプション

データベースサーバで許可される、プロセッサの最大同時実行性を制御します。

#### 構文

```
dbsrv17 -gtc logical-processors-to-use ...
```

## 適用対象

Intel 互換の x86 と x64 プラットフォームで実行される Linux、Solaris、Windows オペレーティングシステム。

## 備考

データベースサーバを起動すると、検出された物理プロセッサと論理プロセッサの数がデータベースサーバメッセージウィンドウに表示されます。

物理プロセッサは、コンピュータの CPU であり、パッケージまたはダイと呼ばれることもあります。物理プロセッサがハイパースレッドをサポートする場合や、マルチプロセッサ (多くの場合、"マルチコアプロセッサ" と呼ばれます) として設定されている場合は、追加の論理プロセッサが存在します。オペレーティングシステムは、論理プロセッサ上でスレッドをスケジュールします。

-gtc オプションでは、データベースサーバが使用する論理プロセッサの数を指定できます。このオプションを指定することで、サーバの起動時に作成されるデータベースサーバスレッドの数を制限します。これは、データベースサーバで同時に実行できるアクティブタスクの数を制限することになります。デフォルトでは、作成されるスレッドの数は 1 + ライセンスを受けているすべての物理プロセッサ上にある論理プロセッサの数です。

デフォルトでは、ライセンスされた各物理プロセッサにおいて、すべての論理プロセッサ (コアまたはハイパースレッド) の同時使用が可能です。

たとえば、2 個の CPU を搭載し、各 CPU に 4 個のコアがあり、コアあたり 2 スレッドを使用できるシステムで、1 CPU ライセンスを使用する場合、ネットワークデータベースサーバでは、一方の CPU で 8 スレッドを同時に実行し、他方の CPU で 0 スレッドとすることができます。-gtc オプションが指定されている場合で、使用される論理プロセッサの数が、ライセンスされている物理プロセッサで利用できる総数よりも少なくなると、データベースサーバはラウンドロビン方式に基づいて論理プロセッサを割り付けます。-gtc オプションに対して暗黙的に 1 を指定すると、クエリ内並列処理 (クエリの並列処理) は無効になります。また、max\_query\_tasks オプションを使用して、クエリ内並列処理を明示的に制限したり無効にしたりすることもできます。

-gtc オプションは、-gta オプションとは一緒に使用できません。

パーソナルデータベースサーバでは、1 つの CPU 上のコアは 4 つに制限されています。

### 例

Windows ベースの SMP コンピュータを例に挙げて考えてみます。ここでは、システムが 4 プロセッサ構成で、各物理プロセッサにコアが 2 つある (合計 8 つの論理プロセッサがある) ことを前提としています。物理プロセッサは文字で、論理プロセッサ (この例では、コア) は数字でそれぞれ区別します。このシステムには 4 プロセッサユニットとして A0、A1、B0、B1、C0、C1、D0、D1 が存在することになります。

シナリオ	ネットワークデータベースサーバの設定
シングル CPU ライセンス、または -gt に 1 が指定されている	<ul style="list-style-type: none"><li>• -gt 1</li><li>• -gtc 2</li><li>• -gnh 20</li></ul> スレッドは A0 または A1 で実行できます。

シナリオ	ネットワークデータベースサーバの設定
CPU のライセンスに制限がなく、-gtc に 5 が指定されている	<ul style="list-style-type: none"> <li>• -gt 4</li> <li>• -gtc 5</li> <li>• -gnh 20</li> </ul> スレッドは A0、A1、B0、C0、D0 で実行できます。
データベースサーバに 3 CPU ライセンスがあり、-gtc に 5 が指定されている	<ul style="list-style-type: none"> <li>• -gt 3</li> <li>• -gtc 5</li> <li>• -gnh 20</li> </ul> スレッドは A0、A1、B0、B1、C0 で実行できます。
CPU のライセンスに制限がなく、-gtc に 1 が指定されている	<ul style="list-style-type: none"> <li>• -gt 4</li> <li>• -gtc 1</li> <li>• -gnh 20</li> </ul> スレッドは A0 でのみ実行できます。

## 関連情報

[スレッド \[330 ページ\]](#)

[max\\_query\\_tasks オプション \[731 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gt データベースサーバオプション \[430 ページ\]](#)

[-gta データベースサーバオプション \[431 ページ\]](#)

### 1.4.1.44 -gtp データベースサーバオプション

トポロジを考慮したスケジューリングをオンまたはオフにします。このオプションは、マルチプロセッサシステムでのみ役立ちます。

#### 構文

```
dbsrv17 -gtp { 0 | 1 } ...
```

#### 指定可能な値

0

トポロジを考慮したスケジューリングをオフにします。

1

トポロジを考慮したスケジューリングをオンにします。

## デフォルト

1

## 適用対象

Microsoft Windows および Linux。

## 備考

トポロジを考慮したスケジューリングをオンにした場合、タスクのスケジューリングは、利用可能な各コアごとにスレッドを1つずつ使用してから、コアの2番目のスレッドを使用するという方法で行われます。

## 関連情報

[スレッド \[330 ページ\]](#)

[-gnh データベースサーバオプション \[423 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

[-gta データベースサーバオプション \[431 ページ\]](#)

### 1.4.1.45 -gu データベースサーバオプション

データベースファイル管理文 (データベースの作成や削除などの文) の実行に必要な権限を設定します。

#### 構文

```
dbsrv17 -gu { DBA | ALL | NONE | utility_db } ...
```

## 指定可能な値

-gu オプション	効果	適用対象
<code>DBA</code>	SERVER OPERATOR システム権限を持つユーザのみがファイル管理文を実行できます。	ユーティリティデータベースを含むすべてのデータベース
<code>ALL</code>	このオプションは推奨されなくなりました。誰でもファイル管理文を実行できます。	ユーティリティデータベースを含むすべてのデータベース
<code>NONE</code>	ファイル管理文の実行は許可されません。	ユーティリティデータベースを含むすべてのデータベース
<code>utility_db</code>	ユーティリティデータベースに接続できるユーザのみがファイル管理文を実行できます。	ユーティリティデータベースのみ

## デフォルト

DBA

## 適用対象

すべてのオペレーティングシステム。

## 備考

次のデータベースファイル管理文を実行できるユーザが制限を受けます。

- ALTER DATABASE dbfile ALTER TRANSACTION LOG
- CREATE DATABASE 文
- CREATE DECRYPTED DATABASE 文
- CREATE DECRYPTED FILE 文
- CREATE ENCRYPTED DATABASE 文
- CREATE ENCRYPTED FILE 文
- DROP DATABASE 文
- RESTORE DATABASE 文。

`utility_db` を指定した場合、これらの文はユーティリティデータベースからのみ実行できます。DBA を指定した場合、これらの文は SERVER OPERATOR システム権限を持つユーザのみ実行できます。none を指定した場合は、どのユーザもこれらの文を実行できません。

### 例

ファイル管理文の使用を防ぐため、-gu オプションの none 権限レベルを使用してデータベースサーバを起動します。次のコマンドは、データベースサーバを起動し、TestSrv という名前を付けます。このコマンドによって mytestdb.db データベ

ースがロードされますが、どのユーザも、そのサーバを使用してデータベースを作成または削除したり、他のファイル管理文を実行したりすることはできません。これは、ユーザのリソース作成権の有無や、ユーティリティデータベースをロードして接続できるかどうかには関係ありません。

```
dbsrv17 -n TestSrv -gu none c:¥mytestdb.db
```

ユーティリティデータベースのパスワードを知っているユーザだけにファイル管理文の実行を許可するには、次のコマンドを実行してサーバを起動します。

```
dbsrv17 -n TestSrv -su passwd -gu utility_db
```

次のコマンドは、Interactive SQL をクライアントアプリケーションとして起動し、TestSrv という名前のサーバに接続し、ユーティリティデータベースをロードして、ユーザを接続させます。

```
dbisql -c "UID=DBA;PWD=passwd;DBN=utility_db;Host=host1;Server=TestSrv"
```

上記のコマンドが正常に実行されると、ユーザがユーティリティデータベースに接続し、ファイル管理文を実行できます。

## 関連情報

[ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.4.1.46 -im データベースサーバオプション

データベースサーバをインメモリで実行し、ディスクへの書き込みをなくしたり軽減したりします。検証オプションを除き、インメモリモードには別途ライセンスが必要です。

#### 構文

```
dbsrv17 -im { NONE | C | NW | V } ...
```

## 指定可能な値

### NONE

インメモリモードは使用されません。これはサーバのデフォルト値です。

### C (Checkpoint only)

チェックポイント専用モードで実行すると、データベースサーバでトランザクションログが使用されないため、コミットした最新のトランザクションにはリカバリできません。ただし、チェックポイントログは有効になっているため、データベースを最新のチェックポイントにリカバリすることができます。通常、トランザクションログを使用せずにデータベースを実行しても、データベースサーバはコミット操作ごとにチェックポイントの設定を実行するため、パフォーマンスに影響します。ただし、デ

データベースサーバをチェックポイント専用モードで実行すると、データベースサーバはコミット操作ごとにチェックポイントの設定を実行しません。

このモードは、パフォーマンスの向上が必要で、最新のチェックポイントの実行後にコミットされたトランザクションが失われても構わないアプリケーションで使用してください。

チェックポイント専用モードでの実行時には、次の制限が適用されます。

1. トランザクションログは生成されません。
2. テンポラリファイルは生成されません。
3. チェックポイントは要求に応じて設定可能で、さらにデータベースサーバの標準のチェックポイント頻度で設定されません。
4. ダーティページはチェックポイントでのみディスクにフラッシュされません。

#### NW (Never write)

非書き込みモードで実行すると、コミットされたトランザクションはディスク上のデータベースファイルに書き込まれません。データベースが停止またはクラッシュするとすべての変更は失われるため、データベースファイルは常に元の状態に保たれます。新しい DB 領域を作成して使用できますが、これらはディスクに書き込まれません。非書き込みモードでバックアップを作成しても、システム DB 領域の変更内容がファイルに書き込まれないので、意味がありません。

非書き込みモードでの実行時には、次の制限が適用されます。

1. DB 領域の拡張または新規作成の要求は許可されますが、変更内容はデータベースファイルに反映されません。
2. トランザクションログは生成されません。
3. チェックポイントログは生成されません。
4. テンポラリファイルは生成されません。
5. ダーティデータベースページはディスクにフラッシュされません。
6. 元のデータベースファイルは変更されません。

#### V (Validation)

データベースのバックアップコピーを変更せずに検証するには、このモードを使用します。このモードには別途ライセンスは必要ありません。

検証モードで実行した場合、データベースファイルとトランザクションログは常に元の状態のままとなります。リカバリ操作は実行されますが、ディスク上のデータベースファイルには書き込まれません。データベースが停止またはクラッシュすると、すべての変更は失われます。

-a、-ad、-fなどのフラグが指定されていても、リカバリ後にデータベースが停止されることはありません。

検証モードでの実行時には、次の制限が適用されます。

1. 新たな DB 領域の拡張、作成、使用の要求は許可されません。
2. データベース変更の要求は許可されません。
3. データベースサーバは実行を続けます。
4. クライアントの接続は読み込み専用となります。
5. トランザクションログは生成されません。
6. チェックポイントログは生成されません。
7. テンポラリファイルは生成されます。
8. ダーティデータベースページはディスクにフラッシュされません。
9. 元のデータベースファイルとトランザクションログは変更されません。

## 適用対象

すべてのオペレーティングシステム。

検証オプションを除き、インメモリモードには別途ライセンスが必要です。

## 備考

この機能は、キャッシュ内のすべてのデータベースファイルを保持するのに十分な大量のメモリを使用できるシステムで使用する場合に最も効果的です。

非書き込みモードの場合 (-im nw)、元のデータベースファイルに変更内容が書き込まれないため、現在のデータベースの永続コピーが必要な場合は、dbunload ユーティリティまたは UNLOAD TABLE 文を使用してください。SQL クエリを使用して変更内容を取得することもできますが、その場合は、手動で変更内容をデータベースファイルに書き込む必要があります。

インメモリモードによるパフォーマンス向上は、アプリケーションの負荷と I/O サブシステムの処理速度によって異なります。パフォーマンスの大幅な向上は、大量データの挿入や更新を行うアプリケーションや、コミットとチェックポイントを頻繁に実行するアプリケーションで顕著に見られます。

多くの場合、インメモリモードでのパフォーマンスは、トランザクションにグローバルテンポラリテーブルを使用した場合のパフォーマンスと同等か、それ以上に改善されます。データベースへの問い合わせが大部分を占めるアプリケーションの場合は、パフォーマンスの向上がほとんど期待できません。一般に、インメモリモードの使用によるパフォーマンスの向上は、予想されるデータベースファイル全体を保持するのに十分な容量がキャッシュに事前に割り当てられている場合に最も顕著に現れます。こうすることで、アプリケーションの実行中にキャッシュを逐次増分させることによって生じるオーバヘッドを軽減できます。

### 警告

非書き込みモードまたは検証モードの場合、ページがキャッシュからフラッシュされないため、データベース内のデータ量が増えすぎると、使用できるキャッシュを使い果たしてしまう場合があります。このような場合、データベースサーバはエラーを発行して、要求の処理を停止します。このような理由から、非書き込みモードの使用には注意が必要です。また、使用する場合は、アプリケーションによる使用が予想されるページの作業セット全体を保持するのに十分なキャッシュを常に確保しておく必要があります。

検証モードを使用する場合は、リカバリによって変更されるページをすべて保持するのに十分なキャッシュを確保しておく必要があります。

チェックポイント専用モードではチェックポイントが引き続き発生するため、非書き込みモードや検証モードの場合に比べ、サーバでキャッシュを使い果たしてしまうリスクは少なくなります。

LOAD TABLE 文と一部の ALTER TABLE 文では、障害の影響を部分的に取り消すため、またはエラーからのリカバリのためにチェックポイントログが使用されます。非書き込みモードでは、チェックポイントログが作成されず、一部の文が失敗するか、エラーが発生しても、文の影響を部分的に取り消すことはできません。不正または不完全なデータがテーブルに残る場合があります。

## 関連情報

[チェックポイントログ \[294 ページ\]](#)

## 1.4.1.47 -k データベースサーバオプション

Windows パフォーマンスモニタ統計とステートメントパフォーマンスサマリ統計の収集を制御します。

### 構文

```
dbsrv17 -k ...
```

### デフォルト

Windows パフォーマンスモニタ統計とステートメントパフォーマンスサマリが収集されます。

### 適用対象

すべてのオペレーティングシステム。

### 備考

データベースサーバの起動時に -k を指定すると、Windows パフォーマンスモニタ統計値は収集されません。-k オプションは、クエリ最適化が使用するカラム統計には影響しません。Windows パフォーマンスモニタ統計とステートメントパフォーマンスデータの収集に関する設定は、sa\_server\_option システムプロシージャを使用しても変更できます。

また、このオプションは、ステートメントパフォーマンスサマリの収集を無効化します。単一のデータベースでステートメントパフォーマンスサマリ収集を無効にするには、sa\_db\_option システムプロシージャの CollectStmtPerfStats オプションを設定します。このオプションを設定するには、MONITOR 権限が必要です。

ステートメントパフォーマンスデータが収集されるのは、データベースサーバとデータベースの両方でデータ収集が有効化されている場合のみです。

このオプションは、データベースサーバをマルチプロセッサコンピュータで実行しており、テストによってパフォーマンスの向上が確認できた場合のみ使用してください。通常の負荷レベルでは、パフォーマンスの向上はわずかであるため、このオプションの使用は推奨できません。パフォーマンスカウンタを無効にすると、パフォーマンス上の問題が発生した場合、分析に必要な情報を取得できません。

## 関連情報

ヒント: 速度が遅い文の原因の特定 [1377 ページ]

-ks データベースサーバオプション [445 ページ]

-ksc データベースサーバオプション [446 ページ]

-ksd データベースサーバオプション [447 ページ]

### 1.4.1.48 -kl データベースサーバオプション

Kerberos GSS-API ライブラリ (UNIX の場合は共有オブジェクト) のファイル名を指定し、データベースサーバへの Kerberos 認証による接続を有効にします。

#### 構文

```
dbsrv17 -kl GSS-API-library-file ...
```

#### 適用対象

すべてのオペレーティングシステム。

#### 備考

このオプションでは、Kerberos GSS-API のロケーションと名前を指定します。このオプションが必要となるのは、Kerberos クライアントでデフォルトと異なる Kerberos GSS-API ライブラリファイル名が使用されているか、データベースサーバを実行しているコンピュータに複数の GSS-API ライブラリがインストールされている場合だけです。Kerberos クライアントのインストールと設定が完了し、データベースサーバが SSPI を使用できない状態であることが必要です。

このオプションを指定すると、データベースサーバに対する Kerberos 認証が有効になります。

#### 例

次のコマンドは、Kerberos 認証に `libgssapi_krb5.so` 共有オブジェクトを使用するデータベースサーバを起動します。

```
dbsrv17 -kl libgssapi_krb5.so -n my_server_princ /opt/myapp/kerberos.db
```

## 関連情報

[Kerberos ユーザ認証 \[149 ページ\]](#)

[-kr データベースサーバオプション \(廃止予定\) \[443 ページ\]](#)

[-krb データベースサーバオプション \[444 ページ\]](#)

[Kerberos \(KRB\) 接続パラメータ \[87 ページ\]](#)

### 1.4.1.49 -kp データベースサーバオプション

Kerberos サーバプリンシパルを指定し、データベースサーバへの Kerberos 認証接続を有効にします。

#### 構文

```
dbsrv17 -kp server-principal ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、データベースサーバによって使用される Kerberos サーバプリンシパルを指定し、Kerberos 認証を有効にします。デフォルトでは、データベースサーバが Kerberos 認証に使用するプリンシパルは `server-name@default-realm` です。`default-realm` は Kerberos クライアントに設定されたデフォルト領域です。別のサーバプリンシパルを使用する場合はこのオプションを指定します。Kerberos サーバプリンシパルの業界フォーマットは、`server-name/hostname@realm` です。

OpenClient または jConnect の Kerberos で認証された接続がデータベースサーバに対して作成された場合、サーバプリンシパルは、アプリケーションによっても指定される必要があります。jConnect のマニュアルの `SERVICE_PRINCIPAL_NAME` を参照してください。

-kp オプションが指定されている場合は、-kr オプションは指定できません。

#### 例

次のコマンドは、Kerberos ログインを受け入れて認証にサーバプリンシパル `myserver/mycomputer.example.com@EXAMPLE.COM` を使用するデータベースサーバを起動します。

```
dbsrv17 -kp myserver/mycomputer.example.com@EXAMPLE.COM -n myserver C:¥kerberos.db
```

## 関連情報

[Kerberos ユーザ認証 \[149 ページ\]](#)

[-kl データベースサーバオプション \[441 ページ\]](#)

[-krb データベースサーバオプション \[444 ページ\]](#)

[-kr データベースサーバオプション \(廃止予定\) \[443 ページ\]](#)

[Kerberos \(KRB\) 接続パラメータ \[87 ページ\]](#)

### 1.4.1.50 -kr データベースサーバオプション (廃止予定)

Kerberos サーバプリンシパルの領域を指定し、データベースサーバへの Kerberos 認証による接続を有効にします。

#### 注記

-kr オプションを使用する方法は廃止されます。-kp オプションを使用して Kerberos サーバプリンシパルを指定してください。-kp を指定するときに、サーバプリンシパルは、データベースサーバを実行しているコンピュータの Kerberos keytab ファイルに抽出されている必要があります。

#### 構文

```
dbsrv17 -kr server-realm ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションでは、Kerberos サーバプリンシパルの領域を指定します。通常、データベースサーバが Kerberos 認証に使用するプリンシパルは `server-name@default-realm` です。`default-realm` は Kerberos クライアントに設定されたデフォルト領域です。このオプションは、サーバプリンシパルの領域をデフォルト領域以外に変更する場合に使用します。このオプションを指定すると、サーバプリンシパルは `server-name@server-realm` となります。

このオプションを指定すると、データベースサーバに対する Kerberos 認証が有効になります。

-kp オプションが指定されている場合は、-kr オプションは指定できません。

## 例

次のコマンドは、Kerberos ログインを受け入れて認証にプリンシパル `my_server_princ@MYREALM` を使用するデータベースサーバを起動します。

```
dbsrv17 -kr MYREALM -n my_server_princ C:¥kerberos.db
```

## 関連情報

[Kerberos ユーザ認証 \[149 ページ\]](#)

[-kl データベースサーバオプション \[441 ページ\]](#)

[-kp データベースサーバオプション \[442 ページ\]](#)

[-krb データベースサーバオプション \[444 ページ\]](#)

[Kerberos \(KRB\) 接続パラメータ \[87 ページ\]](#)

## 1.4.1.51 -krb データベースサーバオプション

データベースサーバへの Kerberos 認証接続を有効にします。

### 構文

```
dbsrv17 -krb ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、データベースサーバに対する Kerberos ユーザ認証を有効にします。データベースサーバが Kerberos を使用してクライアントを認証するためには、`-krb`、`-kl`、`-kr` のうち 1 つ以上を指定する必要があります。

Kerberos ユーザ認証を使用するためには、クライアントコンピュータとデータベースサーバコンピュータの両方に Kerberos クライアントがインストールされ、設定が完了している必要があります。さらに、Kerberos KDC にプリンシパル `server-name@REALM` があり、プリンシパル `server-name@REALM` のキータブがデータベースサーバコンピュータ上の keytab ファイルに抽出されていることも必要です。この準備が完了していない場合、`-krb` オプションを指定すると、データベースサーバは起動しません。

### 注記

データベースサーバ名には、/、¥、@ を使用できません。また、Kerberos ではマルチバイト文字を使用したデータベースサーバ名は使用できません。

Kerberos ログインを許可するためには login\_mode データベースオプションを設定する必要があります。また、GRANT KERBEROS LOGIN 文を使用して、Kerberos クライアントプリンシパルをデータベースユーザ ID にマッピングすることも必要です。

### 例

次のコマンドは、データベースサーバの Kerberos プリンシパル my\_server\_princ@MYREALM に対し、データベースサーバ my\_server\_princ を起動します。

```
dbsrv17 -krb -n my_server_princ C:¥kerberos.db
```

## 関連情報

[Kerberos ユーザ認証 \[149 ページ\]](#)

[-kl データベースサーバオプション \[441 ページ\]](#)

[-kr データベースサーバオプション \(廃止予定\) \[443 ページ\]](#)

[Kerberos \(KRB\) 接続パラメータ \[87 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

## 1.4.1.52 -ks データベースサーバオプション

Windows パフォーマンスモニタで、データベースサーバからカウンタ値を収集するために使用される共有メモリの作成を無効にします。

### 構文

```
dbsrv17 -ks 0 ...
```

## 適用対象

Windows

## 備考

このオプションを指定すると、Windows パフォーマンスモニタには現在のデータベースサーバのデータベースサーバ名、データベース名、接続統計は表示されません。

## 関連情報

[ヒント: 速度が遅い文の原因の特定 \[1377 ページ\]](#)

[-k データベースサーバオプション \[440 ページ\]](#)

[-ksc データベースサーバオプション \[446 ページ\]](#)

[-ksd データベースサーバオプション \[447 ページ\]](#)

### 1.4.1.53 -ksc データベースサーバオプション

Windows パフォーマンスモニタでモニタできる接続の最大数を指定します。

#### 構文

```
dbsrv17 -ksc integer ...
```

## デフォルト

10

## 適用対象

Windows

## 関連情報

[ヒント: 速度が遅い文の原因の特定 \[1377 ページ\]](#)

[-k データベースサーバオプション \[440 ページ\]](#)

[-ks データベースサーバオプション \[445 ページ\]](#)

[-ksd データベースサーバオプション \[447 ページ\]](#)

## 1.4.1.54 -ksd データベースサーバオプション

Windows パフォーマンスモニタでモニタできるデータベースの最大数を指定します。

### 構文

```
dbsrv17 -ksd integer ...
```

### デフォルト

2

### 適用対象

Windows

### 関連情報

[ヒント: 速度が遅い文の原因の特定 \[1377 ページ\]](#)

[-k データベースサーバオプション \[440 ページ\]](#)

[-ks データベースサーバオプション \[445 ページ\]](#)

[-ksc データベースサーバオプション \[446 ページ\]](#)

## 1.4.1.55 -m データベースサーバオプション

チェックポイントの実行後にトランザクションログをトランケートします。

### 構文

```
dbsrv17 -m ...
```

### 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションを指定すると、シャットダウン時、またはサーバでスケジュールされたチェックポイントの結果としてチェックポイントが実行されたときに、トランザクションログをトランケートします。

### 警告

このオプションを選択すると、データベースファイルを含むデバイスのメディア障害に対して無防備な状態になります。

このオプションを指定すると、トランザクションログの肥大化が自動的に制限されます。チェックポイントの頻度は、`checkpoint_time` と `recovery_time` オプションによって制御できます (また、コマンドラインでも設定できます)。

`-m` オプションは、高速な応答時間を必要とする大容量のトランザクションを処理する場合や、リカバリやレプリケーションがトランザクションログの内容に依存しない場合に、トランザクションログのサイズを制限するのに役立ちます。`-m` オプションは、各 COMMIT の後にチェックポイントが必要で、その結果パフォーマンスが低下するような場合に、トランザクションログなしで稼働する場合の代替策となります。`-m` オプションを指定すると、データベースファイルを含むデバイスのメディア障害に対して無防備な状態になります。`-m` オプションを使用する前に、トランザクションログを管理する他の代替策 (BACKUP DATABASE 文やイベントの使用など) を検討してください。

データベースファイルの断片化を防ぐためには、トランザクションログをデータベースそのものとは別のデバイスまたはパーティションに保管してください。

このオプションを指定すると、チェックポイントの実行中に他の操作は行われません。

### 警告

レプリケートされるデータベースまたは同期されるデータベースでは、`-m` オプションを使わないでください。

## 関連情報

[トランザクションログ \[279 ページ\]](#)

[チェックポイントログ \[294 ページ\]](#)

[-m データベースオプション \[528 ページ\]](#)

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

[checkpoint\\_time オプション \[667 ページ\]](#)

[recovery\\_time オプション \[768 ページ\]](#)

## 1.4.1.56 -n データベースサーバオプション

データベースサーバの名前を設定します。

### 構文

```
dbsrv17 -n server-name database-filename ...
```

## デフォルト

データベースサーバで起動された最初のデータベースファイルの名前 (パスと拡張子を取り除いた名前)。

## 適用対象

すべてのオペレーティングシステム。

## 備考

データベースサーバは起動するとき、そのコンピュータのデフォルトのデータベースサーバになろうとします。デフォルトのサーバがない場合、最初に起動したデータベースサーバが、デフォルトのデータベースサーバになります。そのコンピュータで、データベースサーバ名を明示的に指定しないで共有メモリに接続しようとする、デフォルトのサーバに接続されます。

### i 注記

配備されたアプリケーションで使用されているデータベースサーバには `-xd` オプションを使用します。また、すべてのクライアントは、`ServerName (Server)` 接続パラメータを使用して、接続先のデータベースサーバ名を明示的に指定します。このようにすると、コンピュータで複数のデータベースサーバが実行されているときに、データベースが正しいデータベースサーバに接続します。

サーバ名に対して文字セット変換は実行されません。クライアントとデータベースサーバとで文字セットが異なる場合、サーバ名に拡張文字が使用されているとサーバが見つからないことがあります。クライアントとサーバを異なるオペレーティングシステムやロケールで実行している場合は、サーバ名に 7 ビットの ASCII 文字を使用してください。

データベースサーバの名前は、有効な識別子であることが必要です。データベースサーバのロングネームは、プロトコルごとに異なる長さにトランケートされます。データベースサーバ名には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロン、フォワードスラッシュ (/)、またはバックスラッシュ (\) を含む値
- 長さが 250 バイトを超える値
- UNIX 上で実行するときにスペースを含む値

### i 注記

Windows と UNIX では、データベースサーバがバージョン 10.0.0 以降で、名前が次の長さを超えている場合、バージョン 9.0.2 以前のクライアントから接続することはできません。

- Windows 共有メモリの場合は、40 バイト
- UNIX 共有メモリの場合は、31 バイト
- TCP/IP の場合は 40 バイト

サーバ名は、クライアントアプリケーション接続文字列かプロファイルの ServerName (Server) 接続パラメータで使用する名前を指定します。共有メモリ環境では、-xd を指定しないかぎり、サーバ名が指定されておらず少なくとも1つのデータベースサーバがコンピュータで実行されている場合に使用されるデフォルトのデータベースサーバがあります。

同じ名前でも複数のデータベースサーバを実行することはおすすめしません。

### 注記

2つの -n オプションがあります。-n オプションは指定する位置によって意味が異なります。データベースファイル名の前に指定すると、サーバオプションとしてサーバ名を指定します。データベースファイル名の後に指定すると、データベースオプションとしてデータベース名を指定します。

### 例

次の例では、データベースサーバがファイル %SQLANYSAMPI7%\demo.db に対して起動されます。サーバ名は DemoServer です。

```
dbsrv17 -n DemoServer %SQLANYSAMPI7%\demo.db
```

-n オプションが指定されない場合、サーバ名は demo です。

次の例では、データベースサーバオプション -n とデータベースオプション -n を使用して、データベースサーバ名 SERV とデータベース名 DATA を指定します。

```
dbsrv17 -n SERV sales.db -n DATA
```

## 関連情報

[データベースサーバ名とデータベース名 \[320 ページ\]](#)

[接続文字列と文字セット \[582 ページ\]](#)

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

[-xd データベースサーバオプション \[504 ページ\]](#)

[-n データベースオプション \[529 ページ\]](#)

## 1.4.1.57 -ncs データベースサーバオプション

NCS (Native Component Supportability) ライブラリを使用した、SAP Solution Manager へのステータスイベント送信を有効にします。

### 構文

```
dbsrv17 -ncs ...
```

## 適用対象

Windows および x64 Linux。

## 備考

NCS 設定ファイルの場所は、環境変数 NCS\_CONFIG の設定によって特定されます。この環境変数が設定されていない場合、NCS ライブラリは、デフォルトのフォルダ (Linux の場合は `/usr/sap/ncs/config`、Windows の場合は `c:\usr\%sap%\ncs\config`)、続いて現在の作業ディレクトリを検索します。NCS 設定ファイルには、SAP Solution Manager への接続に必要な設定が保存されています。

### 例

```
dbsrv17 -ncs
```

## 関連情報

[Solution Manager \[1300 ページ\]](#)

[-ncsd データベースサーバオプション \[451 ページ\]](#)

## 1.4.1.58 -ncsd データベースサーバオプション

NCS (Native Component Supportability) ライブラリを使用した、SAP Solution Manager へのステータスイベント送信を有効にし、使用する NCS 設定ファイルの場所を指定します。

### 構文

```
dbsrv17 -ncsd filename ...
```

## 適用対象

Windows および x64 Linux。

## 備考

`filename` は、NCS 設定ファイルのパスおよびファイル名を指定します。NCS 設定ファイルには、SAP Solution Manager への接続に必要な設定が含まれています。

### 例

```
dbsrv17 -ncsd c:¥ncs¥ncs.config
```

## 関連情報

[Solution Manager \[1300 ページ\]](#)

[-ncs データベースサーバオプション \[450 ページ\]](#)

## 1.4.1.59 -o データベースサーバオプション

すべてのデータベースサーバメッセージをデータベースサーバメッセージログファイルに出力します。

### 構文

```
dbsrv17 -o filename ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

情報メッセージ、エラー、警告、MESSAGE 文の出力を含むすべてのデータベースサーバメッセージを、指定したファイルとデータベースサーバメッセージウィンドウに出力します。`-o` オプションとともに `-qi` オプションを指定すると、すべてのメッセージがデータベースサーバメッセージログファイルだけに出力されます。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する可能性があるため、ファイル名の最後には `.log` を付けないでください。

データベースサーバメッセージログのファイル名を検索するには、次のコマンドを実行します。

```
SELECT PROPERTY ( 'ConsoleLogFile' );
```

## 関連情報

[データベースサーバのロギング \[322 ページ\]](#)

[-oe データベースサーバオプション \[453 ページ\]](#)

[-on データベースサーバオプション \[454 ページ\]](#)

[-os データベースサーバオプション \[455 ページ\]](#)

[-ot データベースサーバオプション \[456 ページ\]](#)

[-qi データベースサーバオプション \[462 ページ\]](#)

### 1.4.1.60 -oe データベースサーバオプション

起動エラー、致命的なエラー、アサーションをロギングするファイルの名前を指定します。

#### 構文

```
dbsrv17 -oe filename ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

出力ログファイルの各行の先頭には、日付と時刻が記録されます。起動エラーには、次のようなエラーがあります。

- ファイルを開くことも読むこともできません:database-filename。
- その名前のデータベースサーバはすでに起動しています。

致命的なエラーとアサーションは、-oe が指定されているかどうかにかかわらず、Windows アプリケーションイベントログまたは UNIX システムログに記録されます。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する可能性があるため、ファイル名の最後には .log を付けないでください。

## 関連情報

[-o データベースサーバオプション \[452 ページ\]](#)

[-on データベースサーバオプション \[454 ページ\]](#)

[-os データベースサーバオプション \[455 ページ\]](#)

[-ot データベースサーバオプション \[456 ページ\]](#)

[-qi データベースサーバオプション \[462 ページ\]](#)

## 1.4.1.61 -on データベースサーバオプション

データベースサーバメッセージログの最大サイズを指定します。ログファイルがこのサイズに達すると、現在のファイルが拡張子 `.old` の付いた名前に変更され、新しいファイルが作成されます。

### 構文

```
dbsrv17 -on { size[ k | m | g ] } ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

`size` には、データベースサーバメッセージログの最大サイズをバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ `k`、`m`、`g` のいずれかを使用してください。最小のサイズ制限は 10 KB です。デフォルトでは、最大サイズは無制限となります。

データベースサーバメッセージログが指定されたサイズに達すると、データベースサーバによって現在のファイルが拡張子 `.old` の付いた名前に変更され、元の名前を持つ新しいファイルが開始されます。

### i 注記

`.old` データベースサーバメッセージログファイルがすでに存在する場合は、そのファイルが上書きされます。古いデータベースサーバメッセージログファイルを削除しないようにするには、代わりに `-os` オプションを使用します。

このオプションは、`-os` オプションと同時に使用できません。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する可能性があるため、データベースサーバメッセージログファイル名の最後には `.log` を付けしないでください。

### 関連情報

[データベースサーバのロギング \[322 ページ\]](#)

- o データベースサーバオプション [452 ページ]
- oe データベースサーバオプション [453 ページ]
- os データベースサーバオプション [455 ページ]
- ot データベースサーバオプション [456 ページ]

## 1.4.1.62 -os データベースサーバオプション

データベースサーバメッセージログファイルの最大サイズを指定します (このサイズに達するとログのファイル名が変更されます)。

### 構文

```
dbsrv17 -os { size[ k | m | g ] } ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

`size` には、データベースサーバメッセージのログを取るファイルの最大サイズをバイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ `k`、`m`、`g` のいずれかを使用してください。最小のサイズ制限は 10 KB です。デフォルトでは、最大サイズは無制限となります。

データベースサーバは、出力メッセージをデータベースサーバメッセージログファイルに書き込む前に、現在のファイルサイズを確認します。新しいログメッセージを書き込むと、指定されたファイルサイズを超える場合は、データベースサーバメッセージログのファイル名が `yyymmddxx.slg` に変更されます。`yyymmdd` は、そのファイルが作成された年、月、日を表します。`xx` は 00 で始まる番号で、1 ずつ増えていきます。

このオプションによって、データベースサーバメッセージログファイルが古いことを確認して削除し、ディスク領域を解放できます。

このオプションは、`-on` オプションとは一緒に使用できません。

トランザクションログを使用した操作を実行するユーティリティで問題が発生する可能性があるため、データベースサーバメッセージログファイル名の最後には `.log` を付けないでください。

### 関連情報

[データベースサーバのロギング \[322 ページ\]](#)

- o データベースサーバオプション [452 ページ]
- oe データベースサーバオプション [453 ページ]
- on データベースサーバオプション [454 ページ]
- ot データベースサーバオプション [456 ページ]

### 1.4.1.63 -ot データベースサーバオプション

データベースサーバのメッセージログファイルをトランケートし、出力メッセージを追加します。

#### 構文

```
dbsrv17 -ot logfile ...
```

#### 適用対象

すべてのオペレーティングシステム。

#### 備考

このオプションの機能は、データベースサーバメッセージログファイルのトランケート後にメッセージが書き込まれる点を除き、-o オプションと同じです。データベースサーバメッセージログのファイル名を検索するには、次のコマンドを実行します。

```
SELECT PROPERTY ( 'ConsoleLogFile' );
```

トランザクションログを使用した操作を実行するユーティリティで問題が発生する可能性があるため、データベースサーバメッセージログファイル名の最後には .log を付けしないでください。

#### 関連情報

- データベースサーバのロギング [322 ページ]
- o データベースサーバオプション [452 ページ]
- oe データベースサーバオプション [453 ページ]
- on データベースサーバオプション [454 ページ]
- os データベースサーバオプション [455 ページ]

## 1.4.1.64 -p データベースサーバオプション

通信パケットの最大サイズを設定します。

### 構文

```
dbsrv17 -p integer ...
```

### デフォルト

7300 バイト (すべてのオペレーティングシステム)

### 適用対象

すべてのオペレーティングシステム。

### 備考

最小値は 1000 バイトで、最大値は 65535 バイトです。

接続の通信バッファサイズを変更するには、CommBufferSize (CBSIZE) 接続パラメータを設定します。

### 関連情報

[-pc データベースサーバオプション \[457 ページ\]](#)

[-pt データベースサーバオプション \[461 ページ\]](#)

[CommBufferSize \(CBSIZE\) 接続パラメータ \[55 ページ\]](#)

## 1.4.1.65 -pc データベースサーバオプション

同一コンピュータ接続以外のすべての接続のパケットを圧縮します。

### 構文

```
dbsrv17 -pc ...
```

## 適用対象

すべてのオペレーティングシステム。  
ネットワークデータベースサーバ (dbsrv17) のみ。

## 備考

クライアントとサーバの間で送信されるパケットは、-pc オプションを使用して圧縮できます。状況によっては、接続を圧縮することでパフォーマンスが向上する場合があります。大幅に圧縮可能なデータの大規模なデータ転送では、圧縮率が高くなります。クライアントの接続パラメータに COMPRESS=NO を指定すると、特定のクライアントについてこのオプションを上書きできます。

デフォルトでは、接続は圧縮されません。-pc オプションを指定すると、同一コンピュータ接続、Web サービス接続、および TDS 接続を除くすべての接続のパケットが圧縮されます。TDS 接続 (jConnect を含む) では、通信圧縮はサポートされません。

どの通信リンクを使用している場合でも、同一コンピュータ接続は -pc オプションまたは COMPRESS=YES 接続パラメータを使用しても圧縮されません。

## 関連情報

[通信の圧縮設定 \[175 ページ\]](#)

[-p データベースサーバオプション \[457 ページ\]](#)

[-pt データベースサーバオプション \[461 ページ\]](#)

[Compress \(COMP\) 接続パラメータ \[59 ページ\]](#)

### 1.4.1.66 -pf データベースサーバオプション

指定されたファイルにプロセス ID を書き込みます。

#### 構文

```
dbsrv17 -pf filename
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

指定されたファイルはプロセスの起動時に作成され、プロセスの終了時に削除されます。作成時にファイルが存在するとき、ファイルの最初の行に1つの番号があり、その番号がシステムで動作中のプロセスを参照していない場合は、既存のファイルが上書きされます。

プロセスは、指定されたファイルに書き込むのに十分な権限を持つ必要があります。ファイルを作成できない場合、プロセスはエラーメッセージで終了します。

### 例

```
dbsrv17 "%SQLANY17%/dmo.db" -pf processid
```

## 関連情報

[Broadcast Repeater ユーティリティ \(dbns17\) \[1077 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

## 1.4.1.67 -phi データベースサーバオプション

指定したデータベースサーバプロパティの履歴追跡を設定します。

### 構文

```
dbsrv17 -phi { OFF | ON | NONE | property-name, ... }
```

UNIX では、複数のパラメータを指定する場合に二重引用符が必要です。

## デフォルト

ON

## 適用対象

すべてのオペレーティングシステム。

## 指定可能な値

**ON** PropertyHistoryList が有効な場合、プロパティのデフォルトリストが追跡されます。デフォルトは ON です。

**OFF** PropertyHistoryList が無効な場合、データベースサーバのプロパティの追跡が無効になります。

**NONE** このプロパティを NONE に設定すると、プロパティの追跡が有効になりますが、プロパティはデータベースサーバにより追跡されません。データベースにより要求されたプロパティのみが追跡されます。

データベースサーバプロパティのカンマ区切りのリスト

追跡するデータベースサーバプロパティのカンマ区切りのリストを指定します。

## 備考

-phl データベースサーバオプションを使用すると、サーバが追跡するデータベースサーバプロパティ (ある場合) を指定できません。このオプションを設定しないか ON に設定すると、データベースサーバでは PropertyHistoryList データベースサーバプロパティのデフォルト設定が使用されます。

## 関連情報

[-sf データベースサーバオプション \[471 ページ\]](#)

[-phs データベースサーバオプション \[460 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[データベースプロパティの一覧 \[874 ページ\]](#)

[システム権限 \[1502 ページ\]](#)

[すべての追跡可能なデータベースサーバプロパティ値の表示 \[889 ページ\]](#)

## 1.4.1.68 -phs データベースサーバオプション

プロパティ履歴の追跡に使用するメモリの最大容量を、割当時間またはバイトで設定します。

### 構文

```
dbsrv17 -phs { [HH:]MM:SS | size [ K | M ] | max | default }
```

## デフォルト

*default*

## 適用対象

すべてのオペレーティングシステム。

## 備考

-phs データベースサーバオプションは、プロパティ履歴の追跡に使用可能な最大サイズを指定します。サイズは、プロパティごとの割当時間またはメモリのバイトで測定されます。また、`size` パラメータは、起動時にキャッシュを増加させなくて済むように、データベースサーバの初期キャッシュサイズを増加させます。このように初期キャッシュサイズを増やすことにより、起動時間が長くなるのを防ぐことができます。このオプションを設定しなかった場合は、データベースサーバで `PropertyHistorySize` データベースサーバオプションの `default` 設定が使用されます。

-phs データベースサーバオプションの値を指定しても、-phi データベースサーバオプションを OFF に設定すると、プロパティ履歴は収集されません。-phs データベースサーバプロパティを 0 または時間ベースの値に設定すると、プロパティ履歴の追跡に領域が割り当てられません。

`max` を指定すると、プロパティ履歴を追跡するための最大サイズは、キャッシュの 2% または 256 MB のどちらか小さい方に設定されます。

`default` を指定すると、プロパティ履歴を追跡するためのサイズは、10 分またはキャッシュの最大 2% に設定されます。

### i 注記

指定されたサイズでは、追跡される各データベースサーバプロパティの最小データ量 (1024 バイト、または約 30 秒) をデータベースサーバが格納できない場合は、エラーが返されます。

## 関連情報

[-sf データベースサーバオプション \[471 ページ\]](#)

[-phi データベースサーバオプション \[459 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[データベースプロパティの一覧 \[874 ページ\]](#)

[システム権限 \[1502 ページ\]](#)

## 1.4.1.69 -pt データベースサーバオプション

TCP/IP パケットの圧縮サイズの制限値を増減します。

### 構文

```
dbsrv17 -pt size ...
```

## デフォルト

120 バイト

## 適用対象

すべてのオペレーティングシステム。  
ネットワークデータベースサーバ (dbsrv17) のみ。

## 備考

このパラメータには、圧縮が適用されるパケットサイズの最小バイト数を表す整数値を指定します。80 未満の値はおすすめしません。

状況によっては、圧縮のスレッシュホールドを変更すると、パケットの転送速度が上昇する場合のみパケットを圧縮できるようになり、圧縮された接続のパフォーマンスが向上することがあります。ほとんどの場合、デフォルト設定が適しています。

クライアントとサーバで圧縮スレッシュホールドの設定が異なる場合は、クライアントの設定が適用されます。

## 関連情報

[通信の圧縮設定 \[175 ページ\]](#)

[-p データベースサーバオプション \[457 ページ\]](#)

[-pc データベースサーバオプション \[457 ページ\]](#)

[CompressionThreshold \(COMPTH\) 接続パラメータ \[60 ページ\]](#)

### 1.4.1.70 -qi データベースサーバオプション

データベースサーバのシステムトレイアイコンと、データベースサーバのメッセージウィンドウを表示するかどうかを制御します。

#### 構文

```
dbsrv17 -qi ...
```

## 適用対象

Windows

## 備考

このオプションは、起動エラーウィンドウを除いて、サーバの実行中に視覚的な表示が出ないようにします。-o または -oe で指定したログファイルのいずれか（または両方）を使用してエラーを診断できます。

## 関連情報

- [-qn データベースサーバオプション \[463 ページ\]](#)
- [-qp データベースサーバオプション \[464 ページ\]](#)
- [-qs データベースサーバオプション \[465 ページ\]](#)
- [-qw データベースサーバオプション \[466 ページ\]](#)
- [-o データベースサーバオプション \[452 ページ\]](#)
- [-oe データベースサーバオプション \[453 ページ\]](#)

### 1.4.1.71 -qn データベースサーバオプション

起動時にデータベースサーバメッセージウィンドウを最小化しないことを指定します。

#### 構文

```
dbsrv17 -qn ...
```

## 適用対象

Windows

Linux (X-Window Server が使用されている場合)

## 備考

デフォルトでは、データベースサーバが起動すると、データベースサーバメッセージウィンドウは自動的に最小化されます。このオプションを指定すると、データベースサーバが起動しても、データベースサーバメッセージウィンドウは最小化されません。

アプリケーションがアクティブでなく、`-qn` が指定されているときにアプリケーションがデータベースサーバを自動的に起動する場合、データベースサーバメッセージウィンドウはバックグラウンドで表示されることがあります。

Linux では、`-qn` オプションとともに `-ux` オプション (X-Window Server の使用を指定) を指定する必要があります。

### 例

次のコマンドは、Linux または Solaris 上でデータベースサーバを起動し、データベースサーバメッセージウィンドウを表示して、データベースサーバの起動が完了した後でデータベースサーバメッセージウィンドウを最小化しません。

```
dbsrv17 -ux -qn sample.db
```

## 関連情報

[-ux データベースサーバオプション \[497 ページ\]](#)

[-qi データベースサーバオプション \[462 ページ\]](#)

[-qp データベースサーバオプション \[464 ページ\]](#)

[-qs データベースサーバオプション \[465 ページ\]](#)

[-qw データベースサーバオプション \[466 ページ\]](#)

## 1.4.1.72 -qp データベースサーバオプション

データベースサーバメッセージウィンドウに、パフォーマンスに関するメッセージを表示しないように指定します。

### 構文

```
dbsrv17 -qp ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

データベースサーバメッセージウィンドウにパフォーマンスに関するメッセージを表示しないようにします。表示されなくなるメッセージには次のものがあります。

- テーブル "table-name" にユニークなインデックスまたはプライマリキーがありません。
- データベースファイル "mydatabase.db" は nnn のディスクフラグメントで構成されています。

## 関連情報

[-qi データベースサーバオプション \[462 ページ\]](#)

[-qn データベースサーバオプション \[463 ページ\]](#)

[-qs データベースサーバオプション \[465 ページ\]](#)

[-qw データベースサーバオプション \[466 ページ\]](#)

### 1.4.1.73 -qs データベースサーバオプション

起動エラーウィンドウを表示しないようにします。

#### 構文

```
dbsrv17 -qs ...
```

## 適用対象

Windows

## 備考

このオプションを指定すると、起動エラーウィンドウが表示されなくなります。起動エラーには、たとえばデータベースサーバでデータベースファイルを開いたり、読み取ったりできない場合や、指定された名前の別のデータベースサーバがすでに実行中なのでデータベースサーバが起動しない場合などがあります。

Windows プラットフォームでは、サーバが自動的に起動しない場合、これらのエラーがウィンドウに表示されるので、これらをクリアしてからサーバを停止する必要があります。これらのウィンドウは -qs オプションを使用すると表示されません。

言語 DLL のロード時のエラーの場合、このエラーは -o または -oe で指定されたログには記録されず、Windows アプリケーションイベントログに記録されます。

-qs がコマンドラインで指定されていても、@data 拡張で指定されていない場合、使用法エラーは表示されません。

## 関連情報

- [-qi データベースサーバオプション \[462 ページ\]](#)
- [-qn データベースサーバオプション \[463 ページ\]](#)
- [-qp データベースサーバオプション \[464 ページ\]](#)
- [-qw データベースサーバオプション \[466 ページ\]](#)
- [-o データベースサーバオプション \[452 ページ\]](#)
- [-oe データベースサーバオプション \[453 ページ\]](#)

### 1.4.1.74 -qw データベースサーバオプション

データベースサーバメッセージウィンドウを表示しないことを指定します。

#### 構文

```
dbsrv17 -qw ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションを指定すると、データベースサーバメッセージウィンドウが表示されなくなります。Windows プラットフォームでは、データベースサーバシステムトレイアイコンは表示されます。-o または -oe で指定したログファイルのいずれか (または両方) を使用してエラーを診断できます。

## 関連情報

- [-qi データベースサーバオプション \[462 ページ\]](#)
- [-qn データベースサーバオプション \[463 ページ\]](#)
- [-qp データベースサーバオプション \[464 ページ\]](#)
- [-qs データベースサーバオプション \[465 ページ\]](#)

## 1.4.1.75 -r データベースサーバオプション

データベースサーバ上で起動されるすべてのデータベースを、強制的に読み込み専用にします。

データベースへの変更はできません。つまり、データベースサーバはデータベースファイルやトランザクションログファイルを変更しません。

### 構文

```
dbsrv17 -r ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

コマンドラインでどのデータベース名よりも前にオプションを指定した場合、テンポラリファイルを除くすべてのデータベースファイルが読み込み専用モードで開きます。あるデータベース名の後ろに -r オプションを指定した場合、そのデータベースだけが読み込み専用になります。テンポラリテーブルに変更を加えることはできますが、トランザクションログとロールバックログが無効化されているため、ROLLBACK を実行しても効果はありません。

変更が不可能なデータベースファイルの例として、CD-ROM によって配布されるデータベースがあります。このようなデータベースには読み込み専用モードでアクセスできます。

INSERT 文や DELETE 文などを使ってデータベースを変更しようとすると、SQLSTATE\_READ\_ONLY\_DATABASE エラーが発生します。

リカバリの必要なデータベースは、データベースサーバがインメモリ検証モードで実行されていない限り、読み込み専用モードで起動することはできません。オンラインバックアップを使って作成したデータベースファイルは、バックアップの開始時に開いているトランザクションがあると、読み込み専用モードで起動できません。これは、バックアップコピーの開始時に、これらのトランザクションがリカバリを必要とするためです。

監査がオンであるデータベースを読み込み専用モードで起動することはできません。

バックアップデータベースの妥当性を検証する場合は、変更できないように、読み込み専用モードでデータベースを実行してください。または、インメモリ検証モードまたは非書き込みモードでデータベースサーバを起動することもできます。

### 例

次のコマンドは、2つのデータベースを読み込み専用モードで起動します。

```
dbsrv17 -r database1.db database2.db
```

次のコマンドは、2つのうち最初のデータベースだけを読み込み専用モードで開きます。

```
dbsrv17 database1.db -r database2.db
```

## 関連情報

特殊モード [327 ページ]

データベースの検証 (SQL Central) [944 ページ]

-r データベースオプション [531 ページ]

auditing オプション [656 ページ]

### 1.4.1.76 -s データベースサーバオプション

Syslog メッセージのユーザ ID を設定します。

#### 構文

```
dbsrv17 -s { none | user | daemon | localn } ...
```

## 適用対象

Unix

## 備考

Syslog 機能へのメッセージに使用されるシステムユーザ ID を設定します。フォアグラウンドで起動しているデータベースサーバのデフォルトは user で、バックグラウンドで起動しているサーバ (dbspawn で起動した場合、クライアントが自動的に起動した場合、-ud データベースサーバオプションで起動した場合など) のデフォルトは daemon です。

none を指定すると、Syslog メッセージはログに記録されません。引数 localn を使用すると、機能識別子を使用してメッセージをファイルヘリダイレクトできます。n に 0 ~ 7 の数字を指定できます。詳細については、UNIX Syslog(3) の man ページを参照してください。

### 1.4.1.77 -sb データベースサーバオプション

ブロードキャストへのデータベースサーバの反応を指定します。

#### 構文

```
dbsrv17 -sb { 0 | 1 } ...
```

## 適用対象

すべてのオペレーティングシステム。TCP/IP のみ。

## 備考

データベースサーバは、次の 3 種類のブロードキャストに応答できるように、1 つまたは複数の UDP リスナを起動します。

1. このサーバを探しているクライアント接続ブロードキャスト。
2. データベースサーバ列挙ブロードキャスト (dblocate ユーティリティからのブロードキャストや管理ツールの**サーバ検索ウィザード (接続ウィンドウ)**からのブロードキャストなど)。
3. 同じ名前を持つすべてのデータベースサーバを探している別のデータベースサーバから送信されたブロードキャスト。

Embedded SQL 接続では、dblocate ユーティリティと db\_locate\_servers 関数によって、すべてのデータベースサーバを検索する際にブロードキャストパケットがローカルネットワーク上に送出されます。データベースサーバ内の UDP リスナは、その後、データベースサーバへの接続方法について、情報付きで送信側に応答します。

-sb オプションは、データベースサーバ内の UDP リスナの動作を制御します。

-sb オプションを指定しなかった場合、データベースサーバは 3 種類すべての要求に応答します。

-sb に 0 を指定した場合、データベースサーバは (3) にのみ応答します。このため、クライアントがデータベースサーバに接続するときに、Host 接続パラメータまたは HOST プロトコルオプションの使用が強制されます。さらに、サーバがデフォルトポート (2638) 以外のポートを使用している場合、クライアントもサーバのポートを指定する必要があります。データベースサーバがデータベースサーバ列挙ブロードキャストに応答しないため、データベースサーバは dblocate の出力に含まれません。

-sb に 1 を指定した場合、データベースサーバは (1) と (3) に応答します。データベースサーバがデータベースサーバ列挙ブロードキャストに応答しないため、データベースサーバは dblocate の出力に含まれません。

## 関連情報

[BroadcastListener \(BLISTENER\) プロトコルオプション \(サーバ側のみ\) \[184 ページ\]](#)

[サーバ列挙ユーティリティ \(dblocate\) \[1145 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[Host \(IP\) プロトコルオプション \(クライアント側のみ\) \[197 ページ\]](#)

## 1.4.1.78 -sbx データベースサーバオプション

データベースサーバ上で起動されたデータベースのうちディスクサンドボックス設定が明示的に行われていないすべてのデータベースについて、デフォルトのディスクサンドボックス設定を行います。

### 構文

```
dbsrv17 -sbx{ + | - } ...
```

### デフォルト

Off

### 適用対象

すべてのオペレーティングシステム。

### 備考

このオプションは、データベースサーバ上で起動されたデータベースのうちディスクサンドボックス設定が明示的に行われていないすべてのデータベースについて、デフォルトのディスクサンドボックス設定を指定します。データベースに対するディスクサンドボックス設定を指定する -sbx データベースオプションは、-sbx データベースサーバオプションよりも優先されます。

このオプションを有効にすると、データベースサーバで実行されているすべてのデータベースファイルの読み込み/書き込み操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されます。

データベースサーバの実行中にデータベースサーバのディスクサンドボックス設定を変更するには、sa\_server\_option システムプロシージャで DiskSandbox オプションを使用します。sa\_server\_option システムプロシージャは、すでにデータベースサーバで起動されているデータベースや、ディスクサンドボックスが明示的に設定されたデータベースには影響を与えません。

### 関連情報

[-sbx データベースオプション \[534 ページ\]](#)

[disk\\_sandbox オプション \[696 ページ\]](#)

## 1.4.1.79 -sclr データベースサーバオプション

データベースサーバで実行されているすべてのデータベースに対してデータベースサーバが1つの CLR 外部環境を使用するように指定します。

### 構文

```
dbsrv17 -sclr { 35 | 40 | 45 }
```

### デフォルト

データベースサーバがデータベースごとに1つの CLR 外部環境を使用します。

### 適用対象

Windows オペレーティングシステム

### 備考

このオプションは、データベースサーバ上のすべてのデータベースに対してデータベースサーバが1つの CLR 外部環境を使用するかどうかを指定します。このオプションの値は、起動するバージョンを指定します。sa\_server\_option システムプロシージャを呼び出し、SingleCLRInstanceVersion データベースサーバオプションを CLR バージョンに設定することによって、すべてのデータベースに対して1つの CLR 外部環境を使用するようにデータベースサーバを設定することもできます。

### 例

次のコマンドは、myserver1という名称で、データベースサーバで実行されているすべてのデータベースに対して1つの CLR 外部環境を使用するデータベースサーバを起動します。

```
dbsrv17 -n mysserver1 -sclr 35
```

## 1.4.1.80 -sf データベースサーバオプション

現在のデータベースサーバで実行中のデータベースで使用できる機能にユーザがアクセスできるかどうかを制御します。

セキュアでない機能には、適切なシステムロールまたは権限を持つすべてのユーザがアクセスできます。セキュアな機能には、セキュアな機能を使用する権限を付与されたユーザ（適切なセキュリティ機能キーを指定することにより）と、適切なシステムロールまたは権限を持つユーザのみアクセスできます。

## 構文

```
dbsrv17 -sf feature-list ...
```

```
feature-list :  
feature-name | feature-set[,[-]feature-name | feature-set] ...
```

## パラメータ

### none

どの機能もセキュリティで保護されないことを指定します。

### manage\_server

この機能セットを使用すると、ユーザはすべてのデータベースサーバ関連機能にアクセスできなくなります。この機能セットには、次の機能が含まれています。

#### processor\_affinity

ユーザは、データベースサーバのプロセッサのアフィニティ (使用する論理プロセッサの数) を変更できなくなります。

**manage\_cockpitdb** ユーザは、コックピットの有効化や無効化、コックピットのデフォルトファイルの変更ができなくなります。すなわち、manage\_cockpitdb ではユーザは CockpitDB オプションを指定して sa\_server\_option を実行することができなくなります。

#### manage\_listeners

ユーザは、sp\_start\_listener システムプロシージャまたは sp\_stop\_listener システムプロシージャを使用して、接続リスナを起動または停止することができなくなります。

**manage\_property\_history** ユーザは、データベースサーバプロパティ値の追跡を有効化して設定することができなくなります。

### manage\_security

この機能セットを使用すると、ユーザはデータベースサーバのセキュリティの管理を許可する機能にアクセスできなくなります。デフォルトでは、これらの機能はセキュリティで保護されています。

#### manage\_features

ユーザは、データベースサーバ上でセキュリティで保護できる機能のリストを変更できなくなります。

#### manage\_keys

セキュリティ機能キーの作成、変更、削除、またはリストができなくなります。

manage\_keys 機能にアクセスできるものの manage\_features 機能にアクセスできないユーザは、そのユーザに割り当てられているよりも多くの機能を持つキーを定義することができなくなります。

#### manage\_disk\_sandbox

ユーザは、sa\_server\_option システムプロシージャまたは sa\_db\_option システムプロシージャを使用してディスクサンドボックス設定を一時的に変更することができなくなります。manage\_disk\_sandbox 機能は、すべてのデータベースまたはすべてのユーザに対してオフにすることはできません。sp\_use\_secure\_feature\_key システムプロシージャを使用して、個々の接続に対してのみオフにすることができます。

### server\_security

この機能セットを使用すると、ユーザはセキュリティ設定を一時的にバイパスできる機能にアクセスできなくなります。デフォルトでは、`trace_system_event` を除く次の機能がセキュリティで保護されます。

#### **disk\_sandbox**

ユーザは、メインデータベースファイルがあるディレクトリ以外の場所でデータベースファイルの読み込み/書き込み操作を実行できなくなります。

#### **trace\_system\_event**

ユーザは、ユーザ定義のトレースイベントを作成できなくなります。

**database\_isolation** 現在の接続に対して、データベース独立性を一時的にオフにすることができます。

#### **all**

この機能セットを使用すると、ユーザは以下のグループにアクセスできなくなります。

#### **client**

この機能セットを使用すると、ユーザはクライアント関連入出力へのアクセスを許可するすべての機能にアクセスできなくなります。この機能は、クライアントコンピューティング環境へのアクセスを制御します。この機能セットには、次の機能が含まれています。

##### **read\_client\_file**

クライアントファイルの読み込みを可能にする文が使用できなくなります。たとえば、`READ_CLIENT_FILE` 関数や `LOAD TABLE` 文がこれに該当します。

##### **write\_client\_file**

クライアントファイルへの書き込みを可能にする文が使用できなくなります。たとえば、`UNLOAD` 文や `WRITE_CLIENT_FILE` 関数がこれに該当します。

#### **remote**

この機能セットを使用すると、リモートアクセスまたはリモートプロセスとの通信を許可するすべての機能にアクセスできなくなります。この機能セットには、次の機能が含まれています。

##### **remote\_data\_access**

プロキシテーブルなどのリモートデータアクセスサービスがすべて使用できなくなります。

##### **send\_email**

`xp_sendmail` などの電子メールシステムプロシージャが使用できなくなります。

##### **send\_udp**

`sa_send_udp` システムプロシージャを使用して指定したアドレスに UDP パケットを送信する機能が使用できなくなります。

##### **web\_service\_client**

Web サービスクライアントのストアドプロシージャコール (HTTP 要求を発行するストアドプロシージャ) が使用できなくなります。

#### **local**

この機能セットを使用すると、ユーザはすべてのローカル関連機能にアクセスできなくなります。この機能は、サーバコンピューティング環境へのアクセスを制御します。この機能セットには `local_call`、`local_db`、`local_io`、および `local_log` の各機能サブセットが含まれています。

##### **local\_call**

この機能セットを使用すると、ユーザはデータベースサーバの直接的な一部ではなく、データベースサーバによって制御されていないコードを実行できるすべての機能にアクセスできなくなります。この機能セットには、次の機能が含まれています。

#### **cmdshell**

xp\_cmdshell プロシージャが使用できなくなります。

#### **external\_procedure**

ユーザ定義の外部ストアードプロシージャが使用できなくなります。

#### **external\_library\_full\_text**

ユーザ定義の外部単語区切りライブラリが使用できなくなります。

#### **java**

Java プロシージャなどの Java 関連機能が使用できなくなります。

### **local\_db**

この機能セットを使用すると、ユーザはすべてのデータベースファイル関連機能にアクセスできなくなります。この機能セットには、次の機能が含まれています。

#### **backup**

BACKUP DATABASE 文と、BACKUP DATABASE 文を使用したサーバ側バックアップの実行機能が使用できなくなります。dbbackup ユーティリティによるクライアント側バックアップは引き続き実行できます。

#### **restore**

RESTORE DATABASE 文が使用できなくなります。

#### **database**

CREATE DATABASE 文、ALTER DATABASE 文、DROP DATABASE 文が使用できなくなります。

また、CREATE ENCRYPTED FILE 文、CREATE DECRYPTED FILE 文、CREATE ENCRYPTED DATABASE 文、CREATE DECRYPTED DATABASE 文も使用できなくなります。

#### **dbspace**

CREATE DBSPACE 文、ALTER DBSPACE 文、DROP DBSPACE 文が使用できなくなります。

### **local\_env**

この機能セットを使用すると、すべての環境変数関連機能にアクセスできなくなります。この機能セットには、次の機能が含まれています。

#### **getenv**

ユーザは、どの環境変数の値の読み込みもできなくなります。

### **local\_io**

この機能セットを使用すると、ユーザはファイルとその内容への直接アクセスを許可するすべての機能にアクセスできなくなります。この機能セットには、次の機能が含まれています。

#### **create\_trace\_file**

イベントトレースターゲットを作成する文が使用できなくなります。

#### **read\_file**

ローカルファイルの読み込みを可能にする文が使用できなくなります。たとえば、xp\_read\_file システムプロシージャ、LOAD TABLE 文、OPENSTRING( FILE ...) です。

## **write\_file**

ローカルファイルへの書き込みを可能にする文が使用できなくなります。たとえば、UNLOAD 文や xp\_write\_file システムプロシージャがこれに該当します。

## **delete\_file**

ローカルファイルの削除を可能にするすべての文が使用できなくなります。たとえば、この機能をセキュリティで保護することで、-x オプションや -xo オプションを指定した場合に dbbackup ユーティリティの実行が失敗します。

## **directory**

ディレクトリクラスプロキシテーブルが使用できなくなります。remote\_data\_access が無効の場合、この機能は無効になります。

## **file\_directory\_functions**

この機能セットを使用すると、ユーザは以下の個々の機能すべてにアクセスできなくなります。

### **sp\_list\_directory**

sp\_list\_directory システムプロシージャが使用できなくなります。

### **sp\_create\_directory**

sp\_create\_directory システムプロシージャが使用できなくなります。

### **sp\_copy\_directory**

sp\_copy\_directory システムプロシージャが使用できなくなります。

### **sp\_move\_directory**

sp\_move\_directory システムプロシージャが使用できなくなります。

### **sp\_delete\_directory**

sp\_delete\_directory システムプロシージャが使用できなくなります。

### **sp\_copy\_file**

sp\_copy\_file システムプロシージャが使用できなくなります。

### **sp\_move\_file**

sp\_move\_file システムプロシージャが使用できなくなります。

### **sp\_delete\_file**

sp\_delete\_file システムプロシージャが使用できなくなります。

### **sp\_disk\_info**

sp\_disk\_info システムプロシージャが使用できなくなります。

## **local\_log**

ユーザは、結果としてディスク上のファイルにデータを直接作成したり書き込んだりするすべてのロギング機能にアクセスできなくなります。この機能セットには、次の機能が含まれています。

### **request\_log**

要求ログのファイル名を変更する機能と、その最大サイズまたは最大ファイル数を増やす機能が使用できなくなります。データベースサーバの起動コマンドには、要求ログファイルとそのファイルの最大サイズを指定できます。ただし、それらをデータベースサーバの起動後に変更することはできません。要求ログの機能

が無効になっていても、要求ロギングのオンとオフを切り替えたり、要求ログファイルの最大ファイルサイズや最大ファイル数を減らしたりすることは引き続き可能です。

#### console\_log

sa\_server\_option システムプロシージャの ConsoleLogFile オプションを使用してデータベースサーバメッセージログのファイル名を変更する機能が使用できなくなります。また、この機能をセキュリティで保護すると、sa\_server\_option システムプロシージャの ConsoleLogMaxSize オプションを使用してデータベースサーバメッセージログファイルの最大サイズを増やす機能が使用できなくなります。データベースサーバの起動時には、サーバログファイルとそのサイズを指定できます。

#### webclient\_log

sa\_server\_option システムプロシージャの WebClientLogFile オプションを使用して Web サービスクライアントログのファイル名を変更する機能が使用できなくなります。データベースサーバの起動時に Web サービスクライアントログファイルを指定できます。

## デフォルト

none

## 適用対象

すべてのオペレーティングシステム。

## 備考

データベースサーバの所有者はこのオプションを使用して、データベースサーバで実行中のデータベースで使用できる機能にユーザがアクセスできるかどうかを制御できます。-sk データベースサーバオプションを使用すると、データベースサーバの所有者は、-sf データベースサーバオプションにより保護された機能にユーザがアクセスできるようにする SYSTEM セキュリティ機能キーを作成することができます。

-sk データベースサーバオプションを指定せずにデータベースサーバを起動した場合、-sf データベースサーバオプションにより指定された機能といくつかのデフォルト機能がセキュリティで保護され、データベースサーバの実行中はセキュリティで保護される機能を変更することはできません。後になって、システムストアプロシージャを使用して SYSTEM セキュリティ機能キーを作成することはできません。データベースサーバをシャットダウンし、再起動時に -sk データベースサーバオプションを指定する必要があります。

`feature-list` は、データベースサーバで保護する機能名または機能セットをカンマで区切って示したリストです。機能をセキュリティで保護すると、管理者以外のすべてのデータベースユーザがアクセスできなくなります。機能セットを指定すると、そのセットに含まれているすべての機能が保護されます。機能セット内の機能の全部ではなく一部を保護する場合は、個別の機能名を指定します。

機能を保護する (アクセスできなくする) ことを指定するには `feature-name` を使用し、機能の保護を解除する (すべてのデータベースユーザがアクセスできるようにする) ことを指定するには `-feature-name` または `feature-name-` を使用します。たとえば、次のコマンドは、DB 領域機能にのみすべてのユーザがアクセスできることを指定します。

## i 注記

機能セットの部分機能のうち、デフォルトで保護されている機能について、コマンドラインから保護を解除することはできません。たとえば、次のコマンドは動作しません。

```
-sf manage_security,-manage_keys
```

デフォルトで保護される機能セットのサブ機能を使用するには、MANAGE\_KEYS と -sk オプションにより指定されたパスワードを含む SYSTEM セキュリティ機能キーを指定して、sp\_use\_secure\_feature\_key システムプロシージャを呼び出します。

```
CALL sp_use_secure_feature_key( 'system' , 'letmeinweyou' );
```

## 例

次のコマンドは、dbspace 機能を除く、すべてのローカルデータアクセス機能を保護して、secure\_server という名前のデータベースサーバを起動します。

```
dbsrv17 -n secure_server -sf all,-dbspace
```

次のコマンドは、web\_service\_client 機能を除く、すべてのリモートデータアクセス機能を保護して、secure\_server という名前のデータベースサーバを起動します。-sk オプションで指定したキーを、後で sp\_use\_secure\_feature\_key システムプロシージャで使用することで、現在の接続のすべてのユーザがそれらの機能にアクセスできるようになります。

```
dbsrv17 -n secure_server -sf remote,-web_service_client -sk j978kls12
```

secure\_server データベースサーバで実行中のデータベースに接続しているユーザが、sp\_use\_secure\_feature\_key システムプロシージャを使用して、-sk で指定された値と同じ値を authorization\_key パラメータに設定した場合は、その接続からリモートデータアクセス機能にアクセスできます。

```
CALL sp_use_secure_feature_key ( 'MyKey' , 'j978kls12' );
```

次のコマンドは、ローカルデータベース機能を除くすべての機能を保護します。

```
dbsrv17 -n secure_server -sf all,-local_db
```

## 関連情報

[機能の保護 \[1605 ページ\]](#)

[イベントのトレース \[964 ページ\]](#)

[要求ロギング \[1430 ページ\]](#)

[SQL Anywhere Minimum Hardware Requirements \(SQL Anywhere の最小ハードウェア要件\)](#) 

[セキュリティ機能キーの作成 \[1607 ページ\]](#)

[セキュリティ機能キーの作成 \[1607 ページ\]](#)

[-sk データベースサーバオプション \[479 ページ\]](#)

[-sbx データベースサーバオプション \[470 ページ\]](#)

[-sbx データベースオプション \[534 ページ\]](#)

[-zoc データベースサーバオプション \[514 ページ\]](#)

### 1.4.1.81 -sjvm データベースサーバオプション

データベースサーバで実行されているすべてのデータベースに対してデータベースサーバが1つの Java VMを使用するように指定します。

#### 構文

```
dbsrv17 -sjvm { ON | OFF }
```

#### デフォルト

OFF

#### 適用対象

すべてのオペレーティングシステム (Linux ARM を除く)。

#### 備考

このオプションは、データベースサーバ上のすべてのデータベースに対してデータベースサーバが1つの Java VMを使用するかどうかを指定します。sa\_server\_option システムプロシージャを呼び出し、UseSingleJVMInstance データベースサーバオプションを ON に設定することによって、すべてのデータベースに対して1つの Java VMを使用するようにデータベースサーバを設定することもできます。

#### 例

次のコマンドは、myserver1 という名称で、データベースサーバで実行されているすべてのデータベースに対して1つの Java VMを使用するデータベースサーバを起動します。

```
dbsrv17 -n myserver1 -sjvm ON
```

## 1.4.1.82 -sk データベースサーバオプション

SYSTEM セキュリティ機能キーを作成し、認証コードを設定します。これにより、データベースサーバの保護された機能にアクセスできるようになります。

### 構文

```
dbsrv17 -sk auth_code ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

-sf オプションを使用してデータベースサーバの機能を保護するとき、-sk オプションも含めることにより、SYSTEM セキュリティ機能を作成して認証コードを設定することができます。この認証コードは、接続のセキュリティ機能にアクセスできるようにするため、sp\_use\_secure\_feature\_key システムプロシージャとともに使用できます。この場合、sa\_server\_option システムプロシージャを使用して、データベースサーバ上で実行されているすべてのデータベースを保護する機能または機能セットに変更を加えることもできます。

認証コードは、6 文字以上の空でない文字列にする必要があります、二重引用符、制御文字 (0x20 未満のすべての文字)、またはバックslashを含めることはできません。

sp\_use\_secure\_feature\_key システムプロシージャとともに指定された認証コードが、-sk で指定した値と一致しない場合、エラーにはならず、-sf で指定した機能が接続に対して引き続き保護されます。

-sf を指定しないで -sk を指定した場合は、manage\_security 機能セットに含まれる機能など、デフォルトの機能のみが保護されますが、データベースサーバの実行中に SYSTEM セキュリティ機能キーを使用してそれらの機能にアクセスし、セキュリティ機能の設定を変更することができます。

### 例

次のコマンドは、バックアップ機能といくつかのデフォルト機能を保護した状態で、secure\_server という名前のデータベースサーバを起動します。後で、-sk オプションで指定した認証コードを使用し、特定の接続でそれらの機能にアクセスします。

```
dbsrv17 -n secure_server -sf backup -sk j978kls12
```

sp\_use\_secure\_feature\_key システムプロシージャを使用するときは、キー名に SYSTEM を指定して、-sk オプションで指定された認証コードを指定します。これにより、バックアップの実行、secure\_server データベースサーバで保護された機能の変更、他のキーの作成が可能になります。

```
CALL sp_use_secure_feature_key( 'SYSTEM' , 'j978kls12' );
```

次の文を実行すると、secure\_server で実行中のデータベースに対してすべての機能を保護できます。

```
CALL sa_server_option( 'SecureFeatures', 'all' );
```

選択された機能への他のユーザによるアクセスを許可する他のキーを作成できます (キー名と認証コードを開示している場合)。

```
CALL sp_create_secure_feature_key ( 'client_access' , 'client_auth_code' ,  
'client' );
```

## 関連情報

[セキュリティ機能キーの作成 \[1607 ページ\]](#)

[セキュリティ機能キーの作成 \[1607 ページ\]](#)

[-sf データベースサーバオプション \[471 ページ\]](#)

### 1.4.1.83 -su データベースサーバオプション

ユーティリティデータベース (utility\_db) のユーザ ID とパスワードを設定します。または、ユーティリティデータベースへの接続を無効にします。

#### 構文

```
dbsrv17 -su { user-ID,password | password | none } ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、ユーティリティデータベースの最初のパスワードと、必要に応じてユーザ ID を指定します。パスワードは必ず 6 文字以上とします。大文字と小文字を区別します。カンマを含めることはできません。ユーティリティデータベースへのすべての接続を無効にするには、ユーザ ID とパスワードの代わりに *none* と指定します。*user-ID* を指定しなかった場合、データベースサーバではデフォルト値 DBA が使用されます。

- パーソナルデータベースサーバを使用していて、-su オプションを指定していない場合には、データベースサーバはユーザ ID DBA と任意のパスワードでユーティリティデータベースに接続することを許可します。パーソナルデータベースサーバ

バに接続できるユーザであればファイルシステムに直接アクセスできるので、パスワードによるユーザの選別は行われません。

- ネットワークデータベースサーバを使用していて、-su オプションを指定していない場合には、ユーティリティデータベースへの接続は許可されません。
- ネットワークデータベースサーバ用のユーティリティデータベースに接続できれば、データベースサーバは実行中だけでもデータベースに接続できないという場合に便利です。  
たとえば、ミラーリングシステムでは、ユーティリティデータベースに接続してデータベースサーバをシャットダウンしたり、必要に応じてミラーサーバを強制的にプライマリサーバにしたりできるようになります。

utility\_db に接続しているときに、CREATE USER `user-ID` IDENTIFIED BY `new-password` 文を実行して、ユーティリティデータベースのユーザ `user-ID` のパスワードを変更できます。utility\_db データベースへの接続を無効にするには、REVOKE CONNECT FROM `user-ID` 文を使用します。すべての SQL 文がユーティリティデータベースに対してサポートされているわけではありません。

ユーティリティデータベースのパスワードがコマンドライン上でクリアテキストで入力されるのを避けるには、dbfhide ユーティリティを使用して `password` を含むファイルの内容を暗号化し、暗号化されたファイルをコマンドライン上で参照します。

### 例

次のコマンドは、ユーティリティデータベースへのすべての接続を無効にします。

```
dbsrv17 -su none c:¥inventory.db
```

次の例では、ユーティリティデータベースのユーザ ID とパスワードを含む `util_db_uid_pwd.cfg` ファイルが `dbfhide` によって暗号化され、ファイル名が `util_db_uid_pwd_hide.cfg` に変更されます。

```
dbfhide util_db_uid_pwd.cfg util_db_uid_pwd_hide.cfg
```

その後、次のように、`util_db_pwd_hide.cfg` ファイルを使用して、ユーティリティデータベースのユーザ ID とパスワードを指定できます。

```
dbsrv17 -su @util_db_uid_pwd_hide.cfg -n my_server c:¥inventory.db
```

## 関連情報

[ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

[ユーティリティデータベースへの接続 \(接続ウィンドウの場合\) \[298 ページ\]](#)

[ミラーリングシステムのデータベースサーバの停止 \(dbstop ユーティリティ\) \[1684 ページ\]](#)

[トラブルシューティング: プライマリサーバが起動できない \[1694 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

## 1.4.1.84 -tdsl データベースサーバオプション

TDS ログインモードを設定します。

### 構文

```
dbsrv17 -tdsl { all | RSA | RSANonce }...
```

### デフォルト

*all*

### 適用対象

すべてのオペレーティングシステム。

### 備考

このオプションは、データベースサーバでサポートされる TDS ログイン要求のタイプを制限します。

モードのタイプ	説明
all	暗号化されたパスワードと暗号化されていないパスワード (臨時のパスワードを含む) が TDS ログイン要求で許可されます。
RSA	RSA 暗号化されたパスワード (臨時のパスワードを含む) のみが TDS ログイン要求で許可されます。
RSANonce	RSA 暗号化された臨時のパスワードのみが TDS ログイン要求で許可されます。

臨時ではない RSA をサポートする TDS アプリケーションからログイン要求を行うと、ログイン要求用の新しい暗号化キーのセットがデータベースサーバで生成されます。新しい暗号化キーの生成には時間がかかることがあります。

臨時の RSA をサポートする TDS アプリケーションからログイン要求を行うと、RSA 暗号化キーのセットがデータベースサーバで再利用されます。これらの暗号化キーは 24 時間ごとに再生成されます。RSA 暗号化キーを再利用することによって、パフォーマンスを向上させると同時に、データベースサーバをリプレイ攻撃から守ることができます。

jConnect と Open Client のどちらも、臨時の RSA ログイン要求と臨時ではない RSA ログイン要求をサポートしています。

## 1.4.1.85 -ti データベースサーバオプション

非アクティブな接続を切断します。

### 構文

```
dbsrv17 -ti minutes ...
```

### デフォルト

240 (4 時間)

### 適用対象

すべてのオペレーティングシステム。

### 備考

`minutes` で指定された期間、要求を送信しなかった接続を切断します。最大値は 32767 です。データベーストランザクション中のクライアントコンピュータは、トランザクションが終了するか、接続が切断されるまでロックされます。`-ti` オプションを指定すると、非アクティブ接続が切断され、ロックが解除されます。

値を 0 に設定すると、非アクティブ接続は検査されず、接続が切断されません。IdleTimeout (IDLE) 接続パラメータが使用されていない場合、TCP/IP 接続のアイドルタイムアウト値は `-ti` データベースサーバオプションによって制御されます。`-ti` データベースサーバオプションと IdleTimeout (IDLE) 接続パラメータの両方が指定された場合、アイドルタイムアウト値は接続パラメータによって制御されます。

アイドルタイムアウトは TDS クライアント (jConnect クライアントを含む) ではサポートされません。

### 関連情報

[-ti データベースサーバオプション \[484 ページ\]](#)

[ネットワーク通信のトラブルシューティング \[979 ページ\]](#)

[IdleTimeout \(IDLE\) 接続パラメータ \[85 ページ\]](#)

## 1.4.1.86 -tl データベースサーバオプション

活性パケットを送信する期間を設定します。

### 構文

```
dbsrv17 -tl seconds ...
```

### 適用対象

すべてのオペレーティングシステム。TCP/IP のみ。

### 備考

接続が維持されていることを確認するため、クライアント/サーバの TCP/IP 通信プロトコルを介して、定期的に活性パケットが送信されます。接続で活性パケットを検出することなく、指定した LivenessTimeout 時間 (デフォルトは 2 分) にわたってサーバが実行されていると、通信は切断され、サーバはそのクライアントに関連付けられている接続を削除します。非スレッドの UNIX クライアントと TDS 接続では、活性パケットによる確認は行われません。

サーバで -tl オプションを指定すると、活性期間が指定されていないすべてのクライアントに対して LivenessTimeout 値を設定できます。

LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で接続がパケットを送信しない場合に、活性パケットが送信されます。

接続数が 200 を超えると、サーバは指定されている LivenessTimeout 値に基づいて、それより大きい LivenessTimeout 値を自動的に計算するため、多数の接続を効率的に処理できます。活性パケットは、各アイドル接続において、LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で送信されます。大量の活性パケットが同時に送信されることはありません。送信に時間がかかる活性パケットは、LivenessTimeout 値の 3 分の 2 の期間が経過した後で送信できます。活性パケットの送信に時間がかかる場合、データベースサーバメッセージログに警告が表示されます。この警告が発生したら、LivenessTimeout 値の増加を検討してください。

これは一般的にはおすすめしませんが、次のように指定して活性タイムアウトを無効にできます。

```
dbsrv17 -tl 0 -n my_server
```

LivenessTimeout オプションを無効にせずに、次のように値を 1 時間に増やすことを検討してください。

```
dbsrv17 -tl 3600 -n my_server
```

### 関連情報

[-ti データベースサーバオプション \[483 ページ\]](#)

## 1.4.1.87 -tmf データベースサーバオプション

トランザクションマネージャで、分散トランザクションを強制リカバリします。

### 構文

```
dbsrv17 -tmf ...
```

### 適用対象

Windows

### 備考

このオプションは、分散トランザクションコーディネーターが使用できない場合に、分散トランザクションのリカバリ中に使用します。DTC が特定できないときは、未処理の操作をロールバックしてリカバリを続行します。また、DTC が使用できないプラットフォームで、トランザクションログに分散トランザクションがあるデータベースを起動する場合にも使用できます。

### 警告

このオプションを使用すると、分散トランザクションは正常にはリカバリされません。日常的な使用は想定していません。

### 関連情報

[-tmt データベースサーバオプション \[485 ページ\]](#)

## 1.4.1.88 -tmt データベースサーバオプション

分散トランザクションに参加するための再エンリストタイムアウトを設定します。

### 構文

```
dbsrv17 -tmt milliseconds ...
```

## 適用対象

Windows

## 備考

分散トランザクションのリカバリ時に使用します。この値は、データベースサーバが再登録されるまでの待機時間を指定します。デフォルトでは、タイムアウトはありません（データベースサーバは、無期限に待機します）。

## 関連情報

[-tmf データベースサーバオプション \[485 ページ\]](#)

## 1.4.1.89 -tq データベースサーバオプション

指定の時刻にサーバを停止します。

### 構文

```
dbsrv17 -tq { datetime | time } ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、自動オフラインバックアッププロシージャの設定に役立ちます。

時間のフォーマットは `hh:mm` (24 時間表記) で、オプションで前に日付を付けることができます。日付を指定する場合は、日付と時間を二重引用符で囲んで、`YYYY/MM/DDHH:MM` の形式にする必要があります。

## 関連情報

[データベースのバックアップとリカバリ \[894 ページ\]](#)

### 1.4.1.90 -ts データベースサーバオプション

データベースサーバのトレースセッションを設定します。

#### 構文

```
dbsrv17 -ts session-name ( session-option=option-value [ ;... ] )
```

セッションオプション	オプション値
<code>events</code>	システムトレースイベントのカンマで区切られたリスト <code>sp_trace_events</code> システムプロシージャを呼び出し、システム定義のトレースイベントのリストを取得します。
<code>targets</code>	<code>target-type ( target-option=value [ ;... ] )</code> 。ここで、 <code>target-type</code> は <code>file</code> にのみすることができます。

ターゲットファイルには、次のオプションが用意されていることがあります。

ターゲットオプション	オプション値
<code>filename_prefix</code>	パス付きまたはパスなしの ETD ファイル名プレフィクスすべての ETD ファイルには、 <code>.etd</code> という拡張子が付きます。このパラメータは必須です。
<code>max_size</code>	ファイルの最大サイズ (バイト単位)。デフォルトは 0 で、これはファイルサイズに上限がないことを意味し、ディスク領域が許す限り大きくなっていきます。指定したサイズに達すると、新しいファイルが開始されます。
<code>num_files</code>	イベントトレース情報が書き込まれるファイル数で、 <code>max_size</code> が設定されている場合にのみ使用されます。すべてのファイルが指定した最大サイズに達すると、データベースサーバは古いファイルの上書きを開始します。
<code>flush_on_write</code>	記録されるイベントが発生するたびにディスクバッファをフラッシュするかどうかを制御する値。値には、 <code>yes</code> 、 <code>true</code> 、 <code>no</code> 、 <code>false</code> を設定できます。デフォルトは <code>false</code> です。このパラメータをオンにすると、多くのトレースイベントが記録される場合、データベースサーバのパフォーマンスが低下することがあります。
<code>compressed</code>	ディスク領域を節約するための ETD ファイルの圧縮を制御する値。値には <code>on</code> および <code>off</code> を設定できます。デフォルトは <code>off</code> です。

## 適用対象

すべてのオペレーティングシステム。

## 備考

サーバトレースイベントセッションを使用して、システムの動作に関連するトレースイベントや、特定のユーザに関するトレースイベントを取得できます。サーバトレースイベントセッションはメモリに格納され、それらが明示的に削除されていない場合は、データベースサーバが停止すると削除されます。

## 関連情報

[イベントのトレース \[964 ページ\]](#)

[システムイベント \[954 ページ\]](#)

[イベントトレースデータ \(ETD\) ファイル管理ユーティリティ \(dbmanageetd\) \[1104 ページ\]](#)

[-sf データベースサーバオプション \[471 ページ\]](#)

### 1.4.1.91 -u データベースサーバオプション

オペレーティングシステムのディスクキャッシュを使用してファイルを開きます。

#### 構文

```
dbsrv17 -u ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

データベースキャッシュに加え、オペレーティングシステムのディスクキャッシュも使ってファイルが開かれます。

オペレーティングシステムのディスクキャッシュによってパフォーマンスが向上することもあります。データベースキャッシュだけを使用すると、パフォーマンスがさらに向上します。

サーバを専用コンピュータで実行している場合は、`-u` オプションを使用しないでください。通常、データベースキャッシュを使用する方が効率的です。`-u` オプションを使用するのは、サーバを実行するコンピュータに他にもいくつかのアプリケーションがインストールされていて (サイズの大きなデータベースキャッシュが他のアプリケーションを実行するための障害となり)、しかも I/O の多いタスクをサーバ上で断続的に実行する (キャッシュサイズを大きくすることでパフォーマンスが向上する) 場合です。

## 1.4.1.92 `-ua` データベースサーバオプション

非同期 I/O の使用をオフにします。

### 構文

```
dbsrv17 -ua ...
```

### 適用対象

Linux

### 備考

デフォルトで、使用可能な場合にデータベースサーバは Linux で非同期 I/O を使用します。非同期 I/O を使用するには、以下の条件を満たす必要があります。

1. ライブラリ `libaio.so` が実行時にロードできます。
2. カーネルが非同期 I/O をサポートしています。

非同期 I/O の使用をオフにしたい場合は、`-ua` データベースサーバオプションを指定します。

## 1.4.1.93 `-uc` データベースサーバオプション

データベースサーバをシェルモードで起動します。

### 構文

```
dbsrv17 -uc ...
```

## 適用対象

Unix

## 備考

データベースサーバをシェルモードで起動します。これがデフォルトです。-uc、-ui、-um、-uq、-ux のうち 1 つだけを指定してください。-uc を指定すると、データベースサーバはソフトウェアの以前のリリースと同じ方法で起動されます。

-ud データベースサーバオプションを使用すれば、データベースサーバをデーモンとして起動することもできます。

## 関連情報

[-ui データベースサーバオプション \[494 ページ\]](#)

[-um データベースサーバオプション \[495 ページ\]](#)

[-ux データベースサーバオプション \[497 ページ\]](#)

[-ud データベースサーバオプション \[490 ページ\]](#)

## 1.4.1.94 -ud データベースサーバオプション

データベースサーバをデーモンとして実行します。

### 構文

```
dbsrv17 -ud ...
```

## 適用対象

Unix

## 備考

このオプションを使用すると、現在のユーザセッションが終了しても引き続きサーバを実行できます。

-ud オプションを使用してデーモンを直接起動したときは、データベースサーバがデーモンプロセスを作成し、(終了して次のコマンドを実行できるように) すぐに返します。その後、デーモンがそれ自体を初期化するか、コマンドで指定されたデータベースを開こうとします。

-ud オプションの代わりに dbspawn を使用することの利点は、デーモンが起動し、要求を受け入れる状態になったことを確認するまで dbspawn プロセスが終了しないことです。何らかの理由でデーモンの起動が失敗した場合、dbspawn の終了コードは 0 以外の値になります。

データベースサーバをデーモンとして起動すると、現在のユーザの unmask 設定によってパーミッションが制御されます。データベースサーバに適切なパーミッションを付与するため、データベースサーバを起動する前に unmask 値を設定してください。

-um オプションと -ud オプションを同時に使用しないでください。

## 関連情報

[サービスまたはデーモンとしてデータベースサーバを実行する方法 \[339 ページ\]](#)

[セキュリティに関する一般的なヒント \[1598 ページ\]](#)

[サーババックグラウンド起動ユーティリティ \(dbspawn\) \[1161 ページ\]](#)

### 1.4.1.95 -uf データベースサーバオプション

致命的なエラーまたはアサーションの失敗がデータベースサーバで発生したときにデータベースサーバが実行するアクションを指定します。

#### 構文

```
dbsrv17 -uf action ...
```

#### デフォルト

default (UNIX の場合)

defunct (Windows の場合)

#### 指定可能な値

##### abort

データベースサーバが停止し、コアファイルが生成されます。

## default

データベースサーバは、abortと同じ動作をします。ただし、デバイスフルの致命的なエラーが発生した場合は、defunctと同じ動作となります。このアクションによって、システムはフルデバイスとなってもコアファイルを作成しようとしません。

## defunct

データベースサーバは実行を続け、abortを呼び出しません。データベースサーバに対して新しい接続を試みると、元のエラーが発生します。

## 適用対象

すべてのオペレーティングシステム。

## 関連情報

[データベースサーバのロギング \[322 ページ\]](#)

[トラブルシューティング: エラーのレポート \[973 ページ\]](#)

[-oe データベースサーバオプション \[453 ページ\]](#)

[サポートユーティリティ \(dbsupport\) \[1165 ページ\]](#)

[-ufd データベースサーバオプション \[492 ページ\]](#)

## 1.4.1.96 -ufd データベースサーバオプション

致命的なエラーまたはアサーションの失敗がデータベースで発生したときにデータベースサーバが実行するアクションを指定します。

### 構文

```
dbsrv17 -ufd action ...
```

## デフォルト

abort

## 指定可能な値

abort

影響を受けるデータベースが停止します。データベースサーバや他のデータベースのステータスは変化しません。

#### **escalate**

データベースのアサーションの失敗はデータベースサーバのアサーションの失敗となり、-uf データベースサーバオプションの設定に応じて処理されます。

#### **restart**

この値は廃止予定です。代わりに restart\_escalate を使用してください。

#### **restart\_abort**

データベースサーバは、データベースを自動的に再起動します。別のデータベースのアサーションの失敗が再起動の 60 秒以内に起きた場合、再起動は失敗だと見なされます。これに続いて 2 回目の再起動が失敗した場合、影響を受けるデータベースは停止します。データベースサーバや他のデータベースのステータスは変化しません。

#### **restart\_escalate**

データベースサーバは、データベースを自動的に再起動します。別のデータベースのアサーションの失敗が再起動の 60 秒以内に起きた場合、再起動は失敗だと見なされます。これに続いて 2 回目の再起動が失敗した場合、再起動の失敗の原因となったデータベースのアサーションの失敗はサーバのアサーションの失敗となり、-uf データベースサーバオプションの設定に応じて処理されます。

## 適用対象

すべてのオペレーティングシステム。

## 備考

-ufd データベースサーバオプションは、他のどのデータベースファイル指定よりも前に記述する必要があります。

## 関連情報

[データベースサーバのロギング \[322 ページ\]](#)

[トラブルシューティング: エラーのレポート \[973 ページ\]](#)

[-oe データベースサーバオプション \[453 ページ\]](#)

[サポートユーティリティ \(dbsupport\) \[1165 ページ\]](#)

[-uf データベースサーバオプション \[491 ページ\]](#)

## 1.4.1.97 -ui データベースサーバオプション

サーバ起動オプションウィンドウが開き、データベースサーバメッセージウィンドウが表示され、X Window Server が起動するかどうかにかかわらず、データベースサーバが起動します。

### 構文

```
dbsrv17 -ui ...
```

### 適用対象

X-Window Server がサポートされている Linux

### 備考

-ui オプションを使用すると、[サーバ起動オプション](#)ウィンドウを使用して、データベースサーバ起動時のサーバオプションを指定し、データベースサーバが起動したらデータベースサーバメッセージウィンドウを表示できます。

-ui データベースサーバオプションのみを指定した場合は、[サーバ起動オプション](#)ウィンドウが表示され、データベースサーバの起動オプションを入力できます。

-ui データベースサーバオプションを指定すると、データベースサーバは使用可能な表示を探そうとします。DISPLAY 環境変数が設定されていなかったり、X-Window Server が実行されていなかったりしたために、使用可能な表示が見つからなかった場合は、データベースサーバはシェルモードで起動されます。使用可能な表示が見つからない場合にデータベースサーバの起動を中止するには、-ui オプションではなく -ux データベースサーバオプションを指定します。-uc、-ui、-um、-uq、-ux のうち 1 つだけを指定してください。

-ud データベースサーバオプションを使用すれば、データベースサーバをデーモンとして起動することもできます。

### 関連情報

[-uc データベースサーバオプション \[489 ページ\]](#)

[-um データベースサーバオプション \[495 ページ\]](#)

[-ux データベースサーバオプション \[497 ページ\]](#)

[-ud データベースサーバオプション \[490 ページ\]](#)

## 1.4.1.98 -um データベースサーバオプション

DBLauncher.app 内の新しいウィンドウにデータベースサーバメッセージを表示します。

### 構文

```
dbsrv17 -um ...
```

### 適用対象

Mac OS X

### 備考

-um オプションを使用して、DBLauncher.app インスタンスが実行中の場合にこのインスタンスに接続し、DBLauncher.app 内の新しいウィンドウにメッセージを表示することができます。-um オプションは、データベースサーバの起動に必要なその他のオプションと同時に使用する必要があります。サーバメッセージは、シェルではなくこのウィンドウに表示されます。このウィンドウを閉じるとデータベースサーバが停止します。DBLauncher.app インスタンスへの接続を確立できなかった場合、データベースサーバは起動しません。

データベースサーバが DBLauncher.app インスタンスに接続するには、両方が同じ Mac OS X セキュリティコンテキストで実行されている必要があります。たとえば、SSH セッションから起動されたデータベースサーバでは、Launch Services によって起動された DBLauncher.app インスタンスは認識できません。-uc、-ui、-um、-uq、-ux のうち 1 つだけを指定してください。

-ud データベースサーバオプションを使用すれば、データベースサーバをデーモンとして起動することもできます。

-um オプションと -ud オプションを同時に使用しないでください。

### 関連情報

[-uc データベースサーバオプション \[489 ページ\]](#)

[-ui データベースサーバオプション \[494 ページ\]](#)

[-ud データベースサーバオプション \[490 ページ\]](#)

## 1.4.1.99 -uq データベースサーバオプション

データベースサーバをシェルモードで起動しますが、出力を表示しません。

### 構文

```
dbsrv17 -uq ...
```

### 適用対象

Unix

### 備考

データベースサーバをシェルモードで起動しますが、出力を表示しません。Q キーが押されても、データベースサーバは停止しません。uc、-ui、-um、-uq、-ux のいずれかのみを指定します。

## 1.4.1.100 -ut データベースサーバオプション

テンポラリファイルをタッチします。

### 構文

```
dbsrv17 -ut minutes ...
```

### デフォルト

デフォルトは 30 分です。

### 適用対象

Unix

## 備考

このオプションを使用すると、指定の間隔でサーバにテンポラリファイルをタッチさせることができます。

### 1.4.1.101 -ux データベースサーバオプション

Linux で、[サーバ起動オプション](#)ウィンドウを開くか、データベースサーバメッセージウィンドウを表示します (X-Window Server を使用)。

#### 構文

```
dbsrv17 -ux ...
```

## 適用対象

X-Window Server がサポートされている Linux

## 備考

-ux オプションを使用すると、データベースサーバの起動時に 2 つの操作が可能です。それは、[サーバ起動オプション](#)ウィンドウを使用してデータベースサーバの起動時にサーバオプションを指定することと、サーバが起動した後にデータベースサーバメッセージウィンドウを表示することです。

-ux データベースサーバオプションのみを指定した場合は、[サーバ起動オプション](#)ウィンドウが表示され、データベースサーバの起動オプションを入力できます。

-ux が指定されている場合、サーバは使用可能な表示を見つけます。DISPLAY 環境変数が設定されていなかったり、X-Window Server が実行されていなかったりしたために、使用可能な表示が見つからなかった場合は、データベースサーバを起動できません。使用可能な表示が見つからない場合でもデータベースサーバを起動するには、-ux の代わりに -ui オプションを使用します。

-ux の他にもサーバオプションを指定した場合は、データベースサーバが起動するとデータベースサーバメッセージウィンドウが表示されます。-uc、-ui、-um、-uq、-ux のうち 1 つだけを指定してください。

-ud データベースサーバオプションを使用すれば、データベースサーバをデーモンとして起動することができます。

#### 例

次のコマンドを入力すると、データベースサーバの起動オプションを入力するための[サーバ起動オプション](#)ウィンドウが表示されます。

```
dbsrv17 -ux
```

次のコマンドを入力すると、データベースサーバが起動され、データベースサーバメッセージウィンドウが表示されます。

```
dbsrv17 -ux sample.db
```

## 関連情報

[-uc データベースサーバオプション \[489 ページ\]](#)

[-ui データベースサーバオプション \[494 ページ\]](#)

[-qn データベースサーバオプション \[463 ページ\]](#)

[-ud データベースサーバオプション \[490 ページ\]](#)

## 1.4.1.102 -v データベースサーバオプション

ソフトウェアのバージョンを表示します。

### 構文

```
dbsrv17 -v ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

ウィンドウにデータベースサーバのバージョンを表示して、停止します。

## 1.4.1.103 -vss データベースサーバオプション

ボリュームシャドウコピーサービス (VSS) を有効または無効にします。

### 構文

```
dbsrv17 -VSS{ + | - } ...
```

## 適用対象

Microsoft Windows 2003 以降の 32 ビット版と 64 ビット版のオペレーティングシステム

## 備考

デフォルトでは、SQL Anywhere VSS ライター (dbvss17.exe) が実行されている場合、すべてのデータベースでバックアップに VSS サービスを使用できます。VSS は、SQL Anywhere VSS ライターなしでデータベースのバックアップに使用できません。ただし、これらのデータベースをリストアするには、SQL Anywhere の完全なリカバリ手順を使用する必要がある可能性があります。データベースサーバが VSS サービスに参加しないようにするには、データベースサーバの起動時に `-vss-` を指定します。

### 例

次のコマンドは、mydatabase.db データベースを起動し、ライタ (dbvss17.exe) が実行されていても VSS 処理に参加しないようにデータベースサーバに指定します。

```
dbsrv17 -vss- mydatabase.db
```

## 関連情報

[SQL Anywhere ボリュームシャドウコピーサービス \(VSS\) \[900 ページ\]](#)

[データベースファイルのメディア障害からのリカバリ \[938 ページ\]](#)

[Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

## 1.4.1.104 -wc データベースサーバオプション

チェックサムがデフォルトでデータベースによって有効にされていない場合、このデータベースサーバのすべてのデータベースに対する書き込み操作でチェックサムを有効にするかどうかを制御します。

### 構文

```
dbsrv17 -wc[ + | - ] ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

書き込みチェックサム (-wc オプションで有効) とグローバルチェックサム (チェックサムが有効な状態でデータベースを作成) の違いは、-wc ではデータベースページがディスクに書き込まれるときにのみチェックサムが行われる点です。ディスクから読み込まれるページは、ページが書き込まれる前にチェックサム値が計算された場合にのみ検証されます。データベースでチェックサムが有効になっている場合、ページが書き込まれるときにすべてのページに対してチェックサムが計算され、ページが読み込まれるときにすべてのページに対してチェックサムが検証されます。

データベースがリムーバブルストレージデバイス (ネットワーク共有や USB デバイスなど) で実行されていることがデータベースサーバで検出されると、データベースサーバでは、すべてのデータベースページに対してグローバルチェックサムが自動的に有効になります。

デフォルトでは、バージョン 10 または 11 の SQL Anywhere で作成されたデータベースではグローバルチェックサムは有効になっていません。バージョン 12 以降のデータベースサーバで SQL Anywhere 11 を使用して作成されたデータベースを起動すると、デフォルトでは、ページがディスクに書き込まれるときにデータベースサーバでページに対して書き込みチェックサムが作成されます (-wc+)。バージョン 12 以降のデータベースではデフォルトでグローバルチェックサムが有効になっています。デフォルトではすべてのデータベースページにチェックサムがあるため、これらのデータベースに対してデータベースサーバのデフォルト値は -wc- になります。デフォルトのチェックサム設定を使用しない場合は、-wc オプションまたは START DATABASE 文を使用してデータベースサーバのチェックサム動作を変更できます。

次の文を実行することによって、データベースがグローバルチェックサムを有効にして作成されたかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'Checksum' );
```

次の文を実行することによってのみ、書き込み I/O 操作に対して書き込みチェックサムが有効になっているかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'WriteChecksum' );
```

## 関連情報

[チェックサムを使用した破損の検出 \[947 ページ\]](#)

### 1.4.1.105 -x データベースサーバオプション

サーバ側のネットワーク通信プロトコルを指定します。

#### 構文

```
dbsrv17 -x { all | none | tcpip [ ( parm=value;... ) ] ...
```

## 指定可能な値

-x オプションについて選択した設定に関係なく、データベースサーバは常に共有メモリ接続を受け入れます。

デフォルトでは、パーソナルデータベースサーバは共有メモリプロトコルのみを起動します。

デフォルトでは、ネットワークデータベースサーバは共有メモリプロトコルと TCP/IP プロトコルを起動します。

-x オプションには次の値を指定できます。

### TCPIP (TCP)

共有メモリプロトコルと TCP/IP プロトコルを使用するクライアントによる接続試行を受信します。

TCP/IP プロトコルは、ネットワークサーバと、同一コンピュータ通信のパーソナルデータベースサーバでサポートされています。

### NONE

共有メモリプロトコルのみを使用するクライアントによる接続要求を受信します。

### ALL

共有メモリプロトコルと TCP/IP プロトコルを使用するクライアントによる接続試行を受信します。

## 適用対象

すべてのオペレーティングシステム。

## 備考

-x オプションを使用して、クライアント接続の受信に使用する通信プロトコルを指定します。

TCP/IP プロトコルの場合、追加のパラメータは次のフォーマットで提供されます。

```
-x tcpip (PARM1=value1;PARM2=value2;...)
```

UNIX では、複数のパラメータを指定する場合、または括弧などの特定の記号文字を使用する場合に二重引用符が必要です。例:

```
-x "tcpip (PARM1=value1;PARM2=value2;...)"
```

データベースサーバが TCP/IP 接続を受信するとき、デフォルトでポート 2638 からすべてのネットワークカードを受信します。

### 例

ネットワークサーバで次のように共有メモリ接続と TCP/IP 接続を許可します。

```
dbsrv17 -n server_name "%SQLANYXSAMP17%¥demo.db"
```

-x オプションを使用して、TCP/IP の動作をチューニングするプロトコルオプションを指定します。ネットワークサーバに 2 つの具体的なネットワークカードを使用することを許可します。

```
dbsrv17 -x "tcpip(MyIP=192.75.209.12,192.75.209.32)" "%SQLANYSAMP17%¥demo.db"
```

パーソナルサーバで次のように共有メモリ接続と TCP/IP 接続を許可します。

```
dbeng17 -x tcpip "%SQLANYSAMP17%¥demo.db"
```

## 関連情報

[セキュリティに関する一般的なヒント \[1598 ページ\]](#)

[ネットワークプロトコルオプション \[177 ページ\]](#)

[データベースサーバの起動方法 \[310 ページ\]](#)

[TCP/IP プロトコル \[166 ページ\]](#)

[-xd データベースサーバオプション \[504 ページ\]](#)

[-xp データベースオプション \[537 ページ\]](#)

[-xs データベースサーバオプション \[507 ページ\]](#)

[CommLinks \(LINKS\) 接続パラメータ \[57 ページ\]](#)

## 1.4.1.106 -xa データベースサーバオプション

監視サーバに対するデータベース名と認証文字列のカンマ区切りのリストを指定します。

### 構文

```
dbsrv17 -xa [ auth=auth-string [, auth-string... ]; ];DBN={ * | database-name [, database-name... ] }
```

UNIX では、複数のパラメータを指定する場合に二重引用符が必要です。

## 適用対象

すべてのオペレーティングシステム。

ネットワークデータベースサーバ (dbsrv17) のみ。

## 備考

このオプションは、データベースミラーリングシステムで監視サーバを起動する場合のみ指定します。

### auth パラメータ

認証文字列を省略すると、パートナーサーバが提供する認証文字列の検証が行われなくなります。

認証文字列を1つだけ指定する場合は、すべてのデータベースでその認証文字列を使用する必要があります。

複数の認証文字列を認証文字列のリストで指定する場合は、それと同じ数のエントリをデータベース名のリストに含める必要があります。認証文字列は、DBN リスト内の対応するデータベースについて検証されます。

このサーバを監視サーバとして使用するすべてのデータベースでは、認証文字列を SET MIRROR OPTION の `authentication_string` 値に合わせる必要があります。

### DBN パラメータ

このパラメータには、このサーバを監視サーバとして使用できるデータベースをリストします。どのデータベースでもサーバを監視サーバとして使用できるようにするには、**DBN=\*** を指定します。

## 例

次のコマンドは、監視データベースサーバを起動し、データベースがサーバを監視サーバとして使用できるよう指定します。

```
dbsrv17
-n arbiter
-su passwd
-x tcpip(port=6870)
-xf c:¥arbiter¥arbiterstate.txt
-xa "AUTH=abc;DBN=*"
```

次のコマンドは、監視データベースサーバを起動し、2つのデータベースがサーバを監視サーバとして使用できるよう指定します。

```
dbsrv17
-n arbiter
-su passwd
-x tcpip(port=6870)
-xf c:¥arbiter¥arbiterstate.txt
-xa "AUTH=abc,xyz;DBN=database1,database2"
```

この例では、database1 で認証文字列 abc を使用し、database2 で認証文字列 xyz を使用します。

## 関連情報

[チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

[チュートリアル: 監視サーバを共有している複数データベースでデータベースミラーリングシステムを作成する \[1711 ページ\]](#)

[ミラーリングされたデータベースを実行中のミラーリングシステムに追加する \[1686 ページ\]](#)

[DatabaseName \(DBN\) 接続パラメータ \[68 ページ\]](#)

[-xf データベースサーバオプション \[505 ページ\]](#)

[-xp データベースオプション \[537 ページ\]](#)

## 1.4.1.107 -xd データベースサーバオプション

データベースサーバがデフォルトのデータベースサーバにならないようにします。

### 構文

```
dbsrv17 -xd ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

データベースサーバは起動するとき、そのコンピュータのデフォルトのデータベースサーバになろうとします。デフォルトのサーバがない場合、最初に起動したデータベースサーバが、デフォルトのデータベースサーバになります。そのコンピュータで、データベースサーバ名を明示的に指定しないで共有メモリに接続しようとする、デフォルトのサーバに接続されます。

このオプションを指定すると、データベースサーバがデフォルトのデータベースサーバになりません。このオプションを指定した場合、データベースサーバ名を指定しなかったクライアントは、共有メモリ経由でこのデータベースサーバを検出できません。また、-xd オプションを指定すると、データベースサーバでデフォルトの TCP ポートが使用されません。TCP ポートが指定されなかった場合、データベースサーバではポート 2638 以外のポートが使用されます。

### 関連情報

[-n データベースサーバオプション \[448 ページ\]](#)

[StartLine \(START\) 接続パラメータ \[107 ページ\]](#)

[-x データベースサーバオプション \[500 ページ\]](#)

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

[トラブルシューティング: データベースサーバの検出方法 \[251 ページ\]](#)

## 1.4.1.108 -xf データベースサーバオプション

データベースミラーリングシステムに関するステータス情報の管理に使用されるファイルのロケーションを指定します。

このオプションは、データベースミラーリングシステムで監視サーバを起動する場合のみコマンドで使用します。

### 構文

```
dbsrv17 -xf state-file ...
```

### 適用対象

すべてのオペレーティングシステム。

ネットワークデータベースサーバ (dbsrv17) のみ。

### 備考

CREATE MIRROR SERVER を使用して、パートナーサーバのステータス情報ファイルのロケーションを定義します。

-xf オプションでは、データベースミラーリングシステムに関するステータス情報の管理に使用されるファイルのロケーションを指定します。データベースミラーリングには、このオプションは必須です。

### 例

次のコマンドは、ステータス情報ファイル `c:¥arbiter¥arbiter.state` を使用するデータベースサーバ `myarbiter` を起動します。

```
dbsrv17 -n myarbiter -su passwd -x "TCPIP(PORT=6870;DOBROAD=no)" -xf "c:¥arbiter¥arbiter.state" -xa "AUTH=abc;DBN=mirror_demo"
```

### 関連情報

[トラブルシューティング: パートナおよび監視サーバのステータス情報ファイル \[1692 ページ\]](#)

[チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

[チュートリアル: 監視サーバを共有している複数データベースでデータベースミラーリングシステムを作成する \[1711 ページ\]](#)

[-xa データベースサーバオプション \[502 ページ\]](#)

[-xp データベースオプション \[537 ページ\]](#)

## 1.4.1.109 -xm データベースサーバオプション

データベースサーバが新しい IP アドレスをチェックする頻度を指定します。

### 構文

```
dbsrv17 -xm seconds
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

データベースサーバが稼働しているコンピュータが新しいネットワークに接続され、-xm オプションが指定されている場合は、変更が検出され、データベースサーバは新しいコンピュータで接続の受信を開始します。また、コンピュータがネットワークから切断されると、データベースサーバはそのネットワークでの受信を停止します。

SQL Anywhere アプリケーションがポータブルデバイスで実行されていない場合、-xm オプションはデフォルトで 0 に設定されます。ポータブルデバイスでのデフォルト設定は 120 秒です。-xm オプションを無効にするには、0 を指定します。SQL Anywhere アプリケーションがポータブルデバイスで実行されているかどうかを確認するには、IsPortableDevice プロパティを使用します。

このオプションを指定しても、HTTP リスナまたは HTTPS リスナのパフォーマンスには影響を与えません。

MyIP (ME) ネットワークプロトコルオプションを指定した場合、-xm オプションを指定したモニタリングは無効になります。

ネットワークインタフェースが切断されると、ネットワークインタフェースに関連するリスナはすべて停止します。

データベースサーバを停止せずにこのオプションの設定を変更するには、sa\_server\_option システムプロシージャを使用します。

### 例

次のコマンドは、新しい IP アドレスを 60 秒ごとにチェックするデータベースサーバを起動します。

```
dbsrv17 -xm 60
```

### 関連情報

[MyIP \(ME\) プロトコルオプション \[219 ページ\]](#)

## 1.4.1.110 -xs データベースサーバオプション

サーバ側の Web サービス通信プロトコルを指定します。

### 構文

```
dbsrv17 -xs [,...] { protocol[,...] }

protocol : {
  NONE
  | HTTP [ ( option=value;... ) ]
  | HTTPS [ ( option=value;... ) ]
  | ODATA [ ( odata-option=value;... ) ]
}
```

UNIX では、複数のパラメータを指定する場合に二重引用符が必要です。

### 指定可能な値

次のいずれかを指定できます。

#### HTTP

HTTP プロトコルを使用するクライアントによる Web 要求を受信します。受信するデフォルトのポートは 80 です。

#### HTTPS

HTTPS プロトコルを使用するクライアントによる Web 要求を受信します。受信するデフォルトのポートは 443 です。HTTPS を使用するには、サーバの証明書とパスワードを指定します。HTTPS は RSA 暗号化を使用するため、証明書は RSA 証明書であることが必要です。

HTTP サーバは、TLS バージョン 1.0 以降を使用する HTTPS 接続をサポートします。

HTTPS、つまり、FIPS=Y が指定された HTTPS を指定して、FIPS 認定の RSA 暗号化を使用します。FIPS 認定の HTTPS は別の認定ライブラリを使用しますが、HTTPS と互換性があります。

#### NONE

Web 要求を受信しません。これがデフォルトです。

#### ODATA

ODATA プロトコルを使用するクライアントによる Web 要求を受信します。-xs ODATA を指定すると OData サーバが起動し、データベースサーバのサブプロセスとして動作します。

### i 注記

Windows や Linux (非 ARM) 以外のプラットフォーム上で -xs ODATA を指定するとエラーになります。

#### option

ネットワークプロトコルオプション。

#### odata-option

OData プロトコルオプション。

## 適用対象

すべてのオペレーティングシステム。

## 備考

-xs オプションを使用して、要求の受信に使用する Web プロトコルを指定します。

-xs オプションを指定しない場合、データベースサーバは Web 要求を受信しようとしません。

複数のプロトコルを指定する場合は、プロトコルごとに -xs オプションを指定するか、1 つの -xs オプションで複数のプロトコルを指定します。データベースサーバは、指定されたすべてのプロトコルを使用して Web 要求を受信します。

HTTP と OData の両方を指定する場合、少なくとも片方は、デフォルトでない HTTP ポート番号を指定する必要があります。HTTPS と OData の両方を指定する場合、少なくとも片方は、デフォルトでない HTTPS ポート番号を指定する必要があります。

### i 注記

複数の Web サーバを同時に起動する場合は、どのサーバにも同じデフォルトポートがあるので、いずれか 1 つのサーバのポートを変更します。

トランスポートレイヤセキュリティには、HTTPS および FIPS 検証済み HTTPS プロトコルを使用します。

### 例

HTTP Web 要求をポート 80 で受信します。

```
dbsrv17 web.db -xs HTTP (PORT=80)
```

HTTPS を使用して Web 要求を受信します。

```
dbsrv17 web.db -xs  
HTTPS (FIPS=N; PORT=82; IDENTITY=ecserver.id; IDENTITY_PASSWORD=test)
```

ポート 80 とポート 8080 で受信:

```
dbsrv17 -xs HTTP (port=80), HTTP (PORT=8080)
```

または

```
dbsrv17 -xs HTTP (port=80) -xs HTTP (PORT=8080)
```

ポート 8080 で OData 要求を受信するデータベースサーバを起動します。

```
dbsrv17 -xs odata (ServerPort=8080) c:¥mydatabase.db
```

次のコマンドを実行して、ポート 8443 で安全な OData 要求を受信するデータベースサーバを起動します。キーストア `c:¥mykeys.jks` とキーストアパスワード `password` を使用します。また、安全でない OData 要求の受信にはデフォルトのポートを使用することを指定します。

```
dbsrv17 -xs odata(SecureServerPort=8443;SSLKeyStore=c:¥mykeys.jks;SSLKeyStorePassword=password;ServerPort=)
```

## 関連情報

- [ネットワークプロトコルオプション \[177 ページ\]](#)
- [SQL Anywhere Web サービス暗号化 \[1658 ページ\]](#)
- [-fips データベースサーバオプション \[411 ページ\]](#)
- [-sn データベースオプション \[535 ページ\]](#)
- [-x データベースサーバオプション \[500 ページ\]](#)
- [-xa データベースサーバオプション \[502 ページ\]](#)
- [-xf データベースサーバオプション \[505 ページ\]](#)
- [-xp データベースオプション \[537 ページ\]](#)
- [FIPS プロトコルオプション \[195 ページ\]](#)

### 1.4.1.111 -z データベースサーバオプション

診断通信メッセージなどのメッセージをトラブルシューティングのために表示します。

#### 構文

```
dbsrv17 -z ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、問題の原因を突き止める場合にだけ使用します。情報は、データベースサーバメッセージウィンドウに表示されます。

## 関連情報

[-ze データベースサーバオプション \[510 ページ\]](#)

### 1.4.1.112 -ze データベースサーバオプション

データベースサーバメッセージウィンドウにデータベース環境変数を表示します。

#### 構文

```
dbsrv17 -ze ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

-ze オプションを指定すると、起動時に環境変数がデータベースサーバメッセージウィンドウに表示されます。データベースサーバメッセージウィンドウの内容をファイルに保存するには、データベースサーバの起動時に -o オプションを指定します。

#### 例

次のコマンドは、データベースサーバ myserver を起動し、このサーバに設定されている環境変数をデータベースサーバメッセージウィンドウと server-log.txt ファイルに出力します。

```
dbsrv17 -n myserver -ze -o server-log.txt
```

## 関連情報

[環境変数 \[540 ページ\]](#)

[-o データベースサーバオプション \[452 ページ\]](#)

[-z データベースサーバオプション \[509 ページ\]](#)

## 1.4.1.113 -zI データベースサーバオプション

サーバ上のデータベースへの各接続について、最後に作成された SQL 文の取得を有効にします。

### 構文

```
dbsrv17 -zI ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

この機能は、RememberLastStatement サーバ設定を使用してオンにすることもできます。ある接続で最後に準備された SQL 文は、CONNECTION\_PROPERTY 関数の LastStatement 値を使用して取得できます。sa\_conn\_activity ストアドプロシージャを使用すると、サーバ上の現在のすべてのデータベース接続について、最後に作成された SQL 文を取得できます。

LastStatement の値は、文が準備されると同時に設定され、文が削除されると同時にクリアされます。各接続につき 1 つの文の文字列のみが記憶されます。

ある接続について sa\_conn\_activity が空でない値を返した場合、その接続で現在実行されている文である可能性が高くなります。その文が完了している場合は、文がすでに削除され、このプロパティの値がクリアされている可能性があります。アプリケーションが複数の文を準備し、それらの文のステートメントハンドルを保持している場合、LastStatement が返す値は接続で現在実行されている処理を表しません。

ストアドプロシージャコールの場合、プロシージャ内の文ではなく、最も外側のプロシージャコールのみが表示されます。

### 警告

-zI が指定されている場合、または RememberLastStatement サーバ設定がオンになっている場合、すべてのユーザが sa\_conn\_activity システムプロシージャを呼び出すか、LastStatement 接続プロパティの値を取得して、他のユーザが最後に作成した SQL 文を見つけることができます。SERVER OPERATOR、MONITOR、または DROP CONNECTION のいずれかのシステム権限が必要です。

## 1.4.1.114 -zn データベースサーバオプション

保持する要求ログファイルのコピー数を指定します。

### 構文

```
dbsrv17 -zn integer
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

要求ロギングが長時間にわたって有効になっていると、要求ログファイルが大きくなることがあります。-zn オプションを使用すると、保持する要求ログファイルのコピー数を指定できます。ただし、-zs も指定されていなければ有効になりません。-zs オプションにより、元のログファイルが指定のサイズに到達すると、新しいログファイルを作成し、元のログファイルの名前を変更できます。

たとえば、要求ロギング情報を req.out ファイルにリダイレクトし、-zn オプションを使って 5 つのログファイルコピーを要求すると、サーバは以下の順にファイルを作成します。req.out.1、req.out.2、req.out.3、req.out.4、および req.out.5。これらのファイルが存在する場合、アクティブな要求ログが再び一杯になると以下の動作が発生します。

- req.out.1 が削除されます。
- req.out.2 ~ req.out.5 のファイル名が req.out.1 ~ req.out.4 に変更されます。
- アクティブなログファイルのコピーの名前が req.out.5 に変更されます。

要求ロギングを有効にするには、-zr オプションを使用します。このログは、-zo オプションにより別のファイルにリダイレクトできます。また、sa\_server\_option システムプロシージャを使用して、要求ログの数を設定することもできます。その場合、nn には、要求ログファイルのコピーの数を指定します。

```
CALL sa_server_option('RequestLogNumFiles',nn);
```

### 例

次の例では、要求ロギング情報は mydatabase.log という要求ログファイルに出力されます。このファイルは最大サイズが 10 KB で、要求ログのコピーが 3 つ保持されます。

```
dbsrv17 "c:¥my data¥mydatabase.db" -zr all -zn 3  
-zs 10 -zo mydatabase.log
```

## 関連情報

[要求ロギング \[1430 ページ\]](#)

[-zs データベースサーバオプション \[517 ページ\]](#)

[-zo データベースサーバオプション \[513 ページ\]](#)

[-zr データベースサーバオプション \[515 ページ\]](#)

[-zs データベースサーバオプション \[517 ページ\]](#)

### 1.4.1.115 -zo データベースサーバオプション

通常のログファイルとは別のファイルに、要求ログ情報をリダイレクトします。

#### 構文

```
dbsrv17 -ZO filename ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

要求ロギングは、-zr オプションを使用すると有効になります。-zo オプションを指定することによって、出力をこのファイルから通常のログファイルではない別のファイルにリダイレクトできます。

また、このオプションにより、要求ロギングがデータベースサーバメッセージウィンドウに表示されなくなります。

## 関連情報

[要求ロギング \[1430 ページ\]](#)

[-zn データベースサーバオプション \[512 ページ\]](#)

[-zr データベースサーバオプション \[515 ページ\]](#)

[-zs データベースサーバオプション \[517 ページ\]](#)

## 1.4.1.116 -zoc データベースサーバオプション

Web サービスクライアント情報をファイルにリダイレクトします。

### 構文

```
dbsrv17 -zoc filename ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

Web サービスクライアントログファイルには、HTTP 要求とアウトバウンド Web サービスクライアントログコール用に記録されたトランスポートデータが含まれています。Web サービスクライアントログファイルは、sa\_server\_option システムプロシージャを使用して指定することもできます。

```
CALL sa_server_option( 'WebClientLogFile', 'clientinfo.txt' );
```

-zoc サーバオプションを指定すると、ロギングが自動的に有効になります。このファイルへのロギングの有効/無効を切り替えるには、sa\_server\_option システムプロシージャを使用します。

```
CALL sa_server_option( 'WebClientLogging', 'ON' );
```

### 例

次のコマンドは、データベースサーバを起動し、HTTP Web 要求をポート 80 で受信し、アウトバウンド Web サービスクライアント情報を clientinfo.txt ファイルに記録します。

```
dbsrv17 web.db -xs HTTP(PORT=80) -zoc clientinfo.txt
```

## 1.4.1.117 -zp データベースサーバオプション

クエリオプティマイザが最近使用したプランの取得をオンにします。

### 構文

```
dbsrv17 -zp ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、各接続で最近使用されたクエリ実行プランをデータベースサーバに保存する場合に使用します。sa\_server\_option システムプロシージャの RememberLastPlan サーバ設定を使用して、この機能をオンにすることもできます。LastPlanText 接続プロパティを使用することによって、最近使用されたプランのテキストを表示できます。

### 1.4.1.118 -zr データベースサーバオプション

操作の要求ロギングを有効にします。

#### 構文

```
dbsrv17 -zr { SQL | HOSTVARS | PLAN | PROCEDURES | TRIGGERS | OTHER | BLOCKS |  
REPLACE | ALL | NONE } ...
```

## 指定可能な値

### SQL

以下の項目のロギングを有効にします。

- START DATABASE 文
- STOP DATABASE 文
- STOP SERVER 文
- 文の準備と実行
- EXECUTE IMMEDIATE 文
- オプション設定
- COMMIT 文
- ROLLBACK 文
- PREPARE TO COMMIT 操作
- 接続と接続解除
- トランザクションの開始
- DROP STATEMENT 文
- カーソルの説明
- カーソルを開く、閉じる、再開する

- Errors

#### PLAN

実行プランのログギングを有効にします (短期)。プロシージャの実行プランは、プロシージャ (PROCEDURES) のログギングが有効な場合にも記録されます。

#### HOSTVARS

ホスト変数の値のログギングを有効にします。HOSTVARS を指定した場合、SQL にリストされている情報もログギングされません。

#### PROCEDURES

プロシージャとユーザ定義の関数内から実行される文のログギングを有効にします。

#### TRIGGERS

トリガ内から実行されている文のログギングを有効にします。

#### OTHER

SQL に含まれないその他の要求タイプのログギングを有効にします (FETCH や PREFETCH など)。ただし、OTHER を指定して SQL を指定しない場合、SQL+OTHER を指定した場合と同じです。OTHER を含めると、要求ログファイルが急速に拡大し、サーバのパフォーマンス低下につながる可能性があります。

#### BLOCKS

別の接続で接続がブロックされたときと、接続のブロックが解除されたときに表示する詳細のログギングを有効にします。

#### REPLACE

ログギングの開始時に、既存の要求ログは同じ名前を持つ新規の (空の) ログで置換されます。それ以外の場合、既存の要求ログが開き、新規エントリがファイルの末尾に追加されます。

#### ALL

すべてのサポート情報をログギングします。この設定は、SQL+PLAN+HOSTVARS+PROCEDURES+TRIGGERS+OTHER+BLOCKS を指定した場合と同じです。この設定では、要求ログファイルが急速に拡大し、サーバのパフォーマンス低下につながる可能性があります。

#### NO または NONE

要求ログに対するログギングを無効にします。

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、問題の原因を突き止める場合にだけ使用します。情報は、データベースサーバメッセージウィンドウに表示されるか、要求ログに送信されます。複数の値を指定する場合、これらの値は、または + で区切られます。

データベースサーバを起動した後で、sa\_server\_option システムプロシージャを使用し、要求ログ設定を変更してログギングの対象とする情報を増減できます。

### **i** 注記

-zr データベースサーバオプションを使用する場合には、-z0 データベースサーバオプションも指定することをおすすめします。-zr データベースサーバオプションを指定して、-z0 データベースサーバオプションを指定しない場合、データベースサーバのパフォーマンスに影響する可能性があり、メモリ使用量が増加する場合があります。

RequestLogging 設定の現在の値は、次のクエリを使用して検索できます。

```
SELECT PROPERTY( 'RequestLogging' );
```

## 関連情報

[要求ロギング \[1430 ページ\]](#)

[-zn データベースサーバオプション \[512 ページ\]](#)

[-z0 データベースサーバオプション \[513 ページ\]](#)

[-zs データベースサーバオプション \[517 ページ\]](#)

## 1.4.1.119 -zs データベースサーバオプション

要求ログのサイズを制限します。

### 構文

```
dbsrv17 -zs { size[ k | m | g ] } ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

要求ロギングを有効にするには、-zr オプションを使用します。このログは、-z0 オプションにより別のファイルにリダイレクトできます。また、-zs オプションを使用してファイルのサイズを制限できます。

`size` には、要求ログの最大ファイルサイズを、バイト単位で指定します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ `k`、`m`、`g` のいずれかを使用してください。

-zs 0 を指定した場合は、要求ロギングファイルの最大サイズは適用されず、名前は変更されません。これはデフォルト値です。

要求ログファイルが `-zs` オプションまたは `sa_server_option` システムプロシージャで指定したサイズに到達すると、ファイルが拡張子 `.old` の付いた名前に変更されます (既存のファイルが存在する場合は、同じ名前で置換されます)。これにより、要求ログファイルは再起動されます。

#### 例

次の例では、`-zs` オプションを使ってログファイルのサイズを制御しています。次のコマンドラインを使用してデータベースサーバを起動するとします。

```
dbsrv17 -zr all -zs 10k -zo mydatabase.log
```

新しいログファイル `mydatabase.log` が作成されます。このファイルのサイズが 10 KB に達した場合、既存のすべての `mydatabase.old` ファイルが削除され、`mydatabase.log` の名前が `mydatabase.old` に変更され、新しい `mydatabase.log` ファイルが起動されます。このプロセスは、`mydatabase.log` ファイルが指定したサイズ (この場合は 10 KB) に達するたびに繰り返されます。

## 関連情報

[要求ロギング \[1430 ページ\]](#)

[-zn データベースサーバオプション \[512 ページ\]](#)

[-zo データベースサーバオプション \[513 ページ\]](#)

[-zr データベースサーバオプション \[515 ページ\]](#)

## 1.4.1.120 -zt データベースサーバオプション

要求タイミング情報のロギングをオンにします。

#### 構文

```
dbsrv17 -zt ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

-zt オプションを指定すると、RequestTiming プロパティが On に設定されます。要求のタイミングがオンになると、sa\_performance\_diagnostics システムプロシージャを使用して要求のタイミング情報を格納するプロパティのすべての値を取得できます。

次のプロパティは、RequestTiming がオンになったときのみ、要求タイミング情報を返します。それ以外の場合、プロパティは NULL を返します。ReqStatus 以外のプロパティはすべて、接続プロパティ、データベースプロパティ、およびサーバプロパティです。データベースプロパティとサーバプロパティは、データベースまたはサーバが起動されてからの接続値の集まりです。これらのデータベースプロパティとサーバプロパティは、RequestTiming が有効な場合にのみ情報を返します。

- ReqCountActive 接続プロパティ
- ReqCountBlockContention 接続プロパティ
- ReqCountBlockLock 接続プロパティ
- ReqCountUnscheduled 接続プロパティ
- ReqStatus 接続プロパティ
- ReqTimeActive 接続プロパティ
- ReqTimeBlockContention 接続プロパティ
- ReqTimeBlockLock 接続プロパティ
- ReqTimeUnscheduled 接続プロパティ

### 例

RequestTiming 設定の現在の値は、次のクエリを使用して検索できます。

```
SELECT PROPERTY( 'RequestTiming' );
```

## 関連情報

[要求ロギング \[1430 ページ\]](#)

## 1.4.2 データベース起動オプション

データベースオプションは、コマンドラインでデータベースファイルの後に指定し、そのデータベースだけに適用されます。

このセクションの内容:

[-a データベースオプション \[520 ページ\]](#)

指定したトランザクションログを適用します。

[-ad データベースオプション \[521 ページ\]](#)

データベースに適用されるトランザクションログファイルがあるディレクトリを指定します。

[-al データベースオプション \[523 ページ\]](#)

指定したデータベースの指定したユーザに対して標準のユーザ認証を許可します。

- ar [データベースオプション \[523 ページ\]](#)  
現在のトランザクションログと同じディレクトリ内にあるトランザクションログファイルをデータベースに適用します。
- as [データベースオプション \[524 ページ\]](#)  
トランザクションログの適用後もデータベースの実行を継続するように指定します (-ad または -ar とともに使用)。
- ds [データベースオプション \[525 ページ\]](#)  
データベースの DB 領域とトランザクションログの配置先ディレクトリを指定します。
- dh [データベースオプション \[526 ページ\]](#)  
このオプションを指定すると、サーバに対してサーバ列挙ユーティリティ (dblocate) を実行した場合、データベースが表示されません。
- ek [データベースオプション \[527 ページ\]](#)  
強かに暗号化されたデータベースのキーを指定します。
- m [データベースオプション \[528 ページ\]](#)  
チェックポイントの実行後にトランザクションログをトランケートします。
- n [データベースオプション \[529 ページ\]](#)  
データベースの名前を設定します。
- r [データベースオプション \[531 ページ\]](#)  
指定されたデータベースを読み込み専用として起動します。
- ru [データベースオプション \[532 ページ\]](#)  
ポイントインタイムリカバリの実行時にデータベースをリストアするタイムスタンプを指定します。
- ruo [データベースオプション \[533 ページ\]](#)  
ポイントインタイムリカバリの実行時にデータベースをリストアするトランザクションログのオフセットを指定します。
- sbx [データベースオプション \[534 ページ\]](#)  
データベースのディスクサンドボックス機能を制御します。
- sn [データベースオプション \[535 ページ\]](#)  
データベースサーバ上で動作する 1 つのデータベースに代替サーバ名を割り当てます。
- wc [データベースオプション \[536 ページ\]](#)  
チェックサムがデフォルトで有効になっていない場合に、データベースの書き込み操作でチェックサムが有効になっているかどうかを制御します。
- xp [データベースオプション \[537 ページ\]](#)  
高可用性パートナーや、読み込み専用スケールアウトのルートノードまたはコピーノードに対して、データベースミラーリングを有効にします。

## 1.4.2.1 -a データベースオプション

指定したトランザクションログを適用します。

### 構文

```
dbsrv17 [ server-options ] database-file -a log-filename ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションは、データベースファイルのメディア障害からのリカバリに使用されます。このオプションを指定すると、データベースサーバはログを適用した後、停止します。実行は継続されません。複数のトランザクションログを適用する場合は、`-a` オプションを使用するときにトランザクションログの正しい適用順序を把握する必要があります。`-ad` オプションまたは `-ar` オプションを使用すると、複数のトランザクションログが自動的に正しい順序で適用されます。

指定したログファイル名が相対パスを使用する場合、このディレクトリはデータベースファイルディレクトリに対して相対的に読まれます。

`-a` データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

サーバの起動時にキャッシュサイズを指定すると、リカバリ時間を短縮できます。

### 例

次の例は、トランザクションログファイル `mydemo.log` を `mydemo` データベースのバックアップコピーに適用します。

```
dbsrv17 "c:¥backup¥mydemo.db" -a "c:¥backup¥mydemo.log"
```

## 関連情報

[データベースのバックアップとリカバリ \[894 ページ\]](#)

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[データベースファイルのメディア障害からのリカバリ \[938 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

[-ar データベースオプション \[523 ページ\]](#)

[-as データベースオプション \[524 ページ\]](#)

### 1.4.2.2 -ad データベースオプション

データベースに適用されるトランザクションログファイルがあるディレクトリを指定します。

#### 構文

```
dbsrv17 [ server-options ] database-file -ad log-directory-specification ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

`log-directory-specification` は 1 つまたは複数のディレクトリです。複数のディレクトリを指定するときは、セミコロンで区切ります。

`-ad` オプションを使用すると、指定したディレクトリでデータベースのトランザクションログファイルが検索されます。トランザクションログファイルの開始ログオフセットがデータベースファイルに保存されている開始ログオフセットと等しいか、それより大きい場合、トランザクションログファイルはログオフセット順に適用されます。ポイントインタイムリカバリの場合、変更の適用はタイムスタンプまたは指定したオフセットから開始されます。

すべてのトランザクションログファイルが適用されると、データベースは停止します。トランザクションログファイルの適用が完了した後もデータベースの実行を継続する場合は、`-as` データベースサーバオプションも一緒に指定する必要があります。

ログディレクトリが相対パスで指定されている場合、`database-file` ディレクトリからの相対パスであると解釈されます。

`-ad` データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

### 例

次の例では、`c:\¥backup` ディレクトリ内のトランザクションログファイルが `mysample.db` データベースに適用され、トランザクションログファイルの適用が完了すると、データベースが停止します。

```
dbsrv17 "c:\¥mysample.db" -ad "c:\¥backup"
```

次の例では、`c:\¥backup` ディレクトリと `c:\¥backup2` ディレクトリ内のトランザクションログファイルが `mysample.db` データベースに適用され、トランザクションログファイルの適用が完了した後も、データベースの実行は続きます

```
dbsrv17 "c:\¥mysample.db" -ad "c:\¥backup;c:\¥backup2" -as
```

## 関連情報

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[データベースファイルのメディア障害からのリカバリ \[938 ページ\]](#)

[-a データベースオプション \[520 ページ\]](#)

[-ar データベースオプション \[523 ページ\]](#)

[-as データベースオプション \[524 ページ\]](#)

[-ru データベースオプション \[532 ページ\]](#)

[-ruo データベースオプション \[533 ページ\]](#)

### 1.4.2.3 -al データベースオプション

指定したデータベースの指定したユーザに対して標準のユーザ認証を許可します。

#### 構文

```
dbsrv17 [ server-options ] database-file -al userid[ ;userid ... ] ...
```

#### 適用対象

すべてのオペレーティングシステム。

#### 備考

セミコロンで区切られたユーザ ID リストで指定されたユーザに対して、オプションの後に続くデータベースでの標準認証の使用を許可します。最大 5 つのユーザ ID を指定できます。

login\_mode オプションに標準認証が含まれておらず、ユーザが代わりにの方法を使用して認証できない場合に、このオプションが便利です。完全な DBA 権限を持つユーザは、常に標準認証を使用して認証できます。

#### 関連情報

[login\\_mode オプション \[717 ページ\]](#)

### 1.4.2.4 -ar データベースオプション

現在のトランザクションログと同じディレクトリ内にあるトランザクションログファイルをデータベースに適用します。

#### 構文

```
dbsrv17 [ server-options ] database-file -ar ...
```

#### 適用対象

すべてのオペレーティングシステム。

## 備考

-ar オプションを使用すると、データベースサーバのトランザクションログファイルは現在のトランザクションログと同じディレクトリで検索されます。トランザクションログのディレクトリはデータベースから取得されます。トランザクションログファイルの開始ログオフセットがデータベースに保存されている開始ログオフセットと等しいか、それより大きい場合、トランザクションログファイルはログオフセット順に適用されます。すべてのトランザクションログファイルが適用されると、データベースは停止します。トランザクションログファイルの適用が完了した後もデータベースの実行を継続する場合は、-as オプションも一緒に指定する必要があります。

-ar データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

### 例

次の例では、トランザクションログファイル（保存されているディレクトリはデータベースから取得されます）が `mysample.db` データベースに適用されます。トランザクションログファイルの適用が完了した後もデータベースの実行は継続します。

```
dbsrv17 "c:¥mysample.db" -ar -as
```

## 関連情報

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[データベースファイルのメディア障害からのリカバリ \[938 ページ\]](#)

[-a データベースオプション \[520 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

[-as データベースオプション \[524 ページ\]](#)

## 1.4.2.5 -as データベースオプション

トランザクションログの適用後もデータベースの実行を継続するように指定します (-ad または -ar とともに使用)。

### 構文

```
dbsrv17 [ server-options ] database-file { -ad log-dir | -ar } -as ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

-as オプションを指定する場合は、-ad オプションまたは -ar オプションも指定する必要があります。-as オプションを指定すると、トランザクションログの適用後もデータベースの実行が継続します。

-as データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

### 例

次の例では、トランザクションログファイルが `mysample.db` データベースに適用されます。-ar が指定されているため、データベースサーバはトランザクションログのロケーションをデータベースから取得します。トランザクションログファイルの適用が完了した後もデータベースの実行は継続します。

```
dbsrv17 "c:¥mysample.db" -ar -as
```

次の例では、`backup` ディレクトリ内のトランザクションログファイルが `mysample.db` データベースに適用されます。トランザクションログファイルの適用が完了した後もデータベースの実行は継続します。

```
dbsrv17 "c:¥mysample.db" -ad "c:¥backup" -as
```

## 関連情報

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[データベースファイルのメディア障害からのリカバリ \[938 ページ\]](#)

[-a データベースオプション \[520 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

[-ar データベースオプション \[523 ページ\]](#)

## 1.4.2.6 -ds データベースオプション

データベースの DB 領域とトランザクションログの配置先ディレクトリを指定します。

### 構文

```
dbsrv17 database-file -ds dbspace-directory ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

DB 領域のディレクトリを指定すると、データベースサーバは、DB 領域を探すときにこのディレクトリだけを検索対象とします。DB 領域のロケーションは、データベースサーバメッセージウィンドウに表示されます。

フルパス名を指定された DB 領域がバックアップに含まれている場合は、このオプションを使用して、元のデータベースが実行中である間に、元のデータベースと同じコンピュータ上にあるデータベースのバックアップコピーを起動できます。

-ds データベースオプションは、`database-file` の後に指定する必要があるため、そのデータベースだけに適用されます。

このオプションで指定されたディレクトリでトランザクションログが見つからない場合は、このロケーションにトランザクションログが作成されます。

### 警告

-ds オプションはリカバリにのみ使用してください。このオプションを指定し、ここで指定したディレクトリに現在のライブトランザクションログが存在しない場合は、新しいトランザクションログが指定のロケーションで作成されます。

### 例

次の例では、`c:\¥backup¥Nov15` ディレクトリに対して DB 領域を検索するデータベースサーバを起動します。

```
dbsrv17 c:\¥backup¥Nov15¥my.db -ds c:\¥backup¥Nov15¥
```

次の例では、現在のディレクトリに対して DB 領域を検索するデータベースサーバを起動します。

```
dbsrv17 my.db -ds .
```

## 関連情報

[追加の DB 領域の考慮事項 \[270 ページ\]](#)

[default\\_dbpace オプション \[690 ページ\]](#)

## 1.4.2.7 -dh データベースオプション

このオプションを指定すると、サーバに対してサーバ列挙ユーティリティ (dblocate) を実行した場合、データベースが表示されません。

### 構文

```
dbsrv17 [ server-options ] database-file -dh ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

-dh オプションを指定すると、サーバに対してサーバ列挙ユーティリティ (dblocate) を実行したときに、データベースが検出されません。そのため、-d、-dn、-dv のいずれかのオプションを指定して dblocate を実行した場合、-dh オプションを指定したデータベースは表示されません。

-dh データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

## 関連情報

[サーバ列挙ユーティリティ \(dblocate\) \[1145 ページ\]](#)

### 1.4.2.8 -ek データベースオプション

強かに暗号化されたデータベースのキーを指定します。

#### 構文

```
dbsrv17 [ server-options ] database-file -ek key ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

暗号化されたデータベースを起動するには、key 値を -ek オプションに指定してください。key は、大文字、小文字、数字、文字、特殊記号を含む文字列です。

クリアテキストで見ることができないように暗号化キーをウィンドウに入力するには、-ep サーバオプションを使用します。

クライアントアプリケーションとデータベースサーバ間の通信パケットの安全性を保つため、-ec サーバオプションとトランスポートレイヤセキュリティを使用します。

-ek データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

#### 例

次の例では、データベースを起動して、コマンドラインで暗号化キーを指定します。

```
dbsrv17 -x tcpip mydata.db -ek "Akmm9u70y"
```

## 関連情報

[データベースの暗号化と復号化 \[1614 ページ\]](#)

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

[-ep データベースサーバオプション \[406 ページ\]](#)

[DatabaseKey \(DBKEY\) 接続パラメータ \[67 ページ\]](#)

## 1.4.2.9 -m データベースオプション

チェックポイントの実行後にトランザクションログをトランケートします。

#### 構文

```
dbsrv17 [ server-options ] database-file -m ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

シャットダウン時、またはサーバでスケジュールされたチェックポイントの結果としてチェックポイントが実行されたときに、トランザクションログをトランケートします。これでトランザクションログの肥大化が自動的に制限されます。チェックポイントの頻度は、`checkpoint_time` と `recovery_time` オプションによって制御できます (または `-gc` および `-gr` データベースサーバオプションでも設定できます)。

-m オプションは、高速な応答時間を必要とする大容量のトランザクションを処理する場合や、リカバリやレプリケーションがトランザクションログの内容に依存しない場合に役立ちます。このオプションを選択すると、データベースファイルを含むデバイスのメディア障害に対して無防備な状態になります。

データベースファイルの断片化を防ぐためには、トランザクションログをデータベースそのものとは別のデバイスまたはパーティションに保管してください。

このオプションは `-m` サーバオプションと同じですが、現在のデータベースまたは `database-file` 変数で識別されるデータベースにのみ適用されます。

#### 警告

レプリケートされるデータベースまたは同期されるデータベースでは、`-m` オプションを使わないでください。

SQL Remote と Mobile Link で使用されるレプリケーションと同期は、本質的にトランザクションログ情報に依存します。

`-m` データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

#### 例

次の例では、`silver` という名前のデータベースサーバが起動され、データベース `salesdata.db` がロードされます。チェックポイントが終了すると、トランザクションログの内容が削除されます。

```
dbsrv17 -n silver "c:¥inventory details¥salesdata.db" -m
```

## 関連情報

[トランザクションログ \[279 ページ\]](#)

[-m データベースサーバオプション \[447 ページ\]](#)

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

## 1.4.2.10 -n データベースオプション

データベースの名前を設定します。

#### 構文

```
dbsrv17 [ server-options ] database-file -n string ...
```

## デフォルト

データベースには、パスと拡張子を削除したデータベースファイル名が付けられます。

## 適用対象

すべてのオペレーティングシステム。

## 備考

データベースサーバとデータベースはどちらも名前を付けることができます。データベースサーバはいくつかのデータベースをロードできるので、データベース名を使用して、各データベースを区別します。

データベース名には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロンを含む値
- 長さが 250 バイトを超える値

-n データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

ユーティリティデータベースへの接続に使用できるのは `utility_db` という名前のデータベースのみです。

### i 注記

2 つの -n オプションがあります。-n オプションの内容は指定する位置によって異なります。データベースファイル名の前に指定すると、サーバオプションとしてサーバ名を指定します。データベースファイル名の後に指定すると、データベースオプションとしてデータベース名を指定します。

たとえば、次のコマンドでは、サーバ名 `SERV` とデータベース名 `DATA` が指定されます。

```
dbsrv17 -n SERV c:¥mydata.db -n DATA
```

### 例

`%SQLANYSAMP17%¥demo.db` ファイルでデータベースを起動するときに、-n オプションを指定しなかった場合、データベースの名前は `demo` になります。

次の例では、データベースサーバがキャッシュサイズ 3 MB で起動され、データベースがロードされます。データベースには `test` という名前が付けられます。データベースサーバ名が指定されていないため、サーバは最初のデータベースから名前を取得します。そのため、サーバの名前も `test` になります。

```
dbsrv17 -c 3MB "c:¥mydata.db" -n "test"
```

## 関連情報

[データベースサーバ名とデータベース名 \[320 ページ\]](#)

[ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

## 1.4.2.11 -r データベースオプション

指定されたデータベースを読み込み専用として起動します。

### 構文

```
dbsrv17 [ server-options ] database-file -r ...
```

### 適用対象

すべてのオペレーティングシステム。

### 備考

コマンドラインでどのデータベース名よりも前にオプションを指定した場合、テンポラリファイルを除くすべてのデータベースファイル (メインデータベースファイル、DB 領域、トランザクションログ、トランザクションログミラー) が読み込み専用モードで開きます。データベースへの変更はできません。つまり、データベースサーバはデータベースファイルやトランザクションログファイルを変更しません。

あるデータベースファイルの後ろに -r オプションを指定した場合、そのデータベースだけが読み込み専用になります。テンポラリテーブルに変更を加えることはできますが、トランザクションログとロールバックログが無効化されているため、ROLLBACK を実行しても効果はありません。

変更が不可能なデータベースファイルの例として、CD-ROM によって配布されるデータベースがあります。このようなデータベースには読み込み専用モードでアクセスできます。

INSERT 文や DELETE 文などを使ってデータベースを変更しようとすると、SQLSTATE\_READ\_ONLY\_DATABASE エラーが発生します。

リカバリを必要とするデータベースは、読み込み専用モードでは起動できません。たとえば、オンラインバックアップを使って作成したデータベースファイルは、バックアップの開始時に開いているトランザクションがあると、読み込み専用モードで起動できません。これは、バックアップコピーの開始時に、これらのトランザクションがリカバリを必要とするためです。

監査がオンである場合、データベースを読み込み専用モードで起動することはできません。

### 例

2 つのデータベースを読み込み専用モードで開くには、次のように実行します。

```
dbsrv17 -r database1.db database2.db
```

2つのうち最初のデータベースだけを読み込み専用モードで開くには、次のように実行します。

```
dbsrv17 database1.db -r database2.db
```

## 関連情報

[-r データベースサーバオプション \[467 ページ\]](#)

[auditing オプション \[656 ページ\]](#)

## 1.4.2.12 -ru データベースオプション

ポイントインタイムリカバリの実行時にデータベースをリストアするタイムスタンプを指定します。

タイムスタンプの指定時のタイムゾーンの指定は任意ですが、データベースサーバとトランザクションログが異なるタイムゾーンにある場合には必須です。タイムゾーンは、トランザクションログがあるロケーションのタイムゾーンと一致するように設定します。

### 構文

```
dbsrv17 [ server-options ] database-file -ru " time-stamp [ time-zone ]"
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

タイムスタンプの指定にタイムゾーン (-05:00 など) を含めるのはベストプラクティスで、タイムスタンプへのポイントインタイムリカバリを行うときにお勧めします。タイムスタンプの指定でタイムゾーンを省略するときは、リストアを行っているデータベースサーバのローカル時間が使用されます。リストアを実行するデータベースサーバのタイムゾーンがトランザクションログの生成時に使用されたタイムゾーンと一致する場合、問題はありませんが、タイムゾーンが一致しない場合、データベースはデータベースサーバのタイムゾーンの指定した時間にリストアされます (これは正しくありません)。

## 例

次のコマンドは、myDB.db データベースを、トランザクションログの UTC から 5 時間を引いた (-05:00) タイムゾーンのタイムスタンプ 2014-12-15 15:23:15 にリストアします。

```
dbsrv17 myDB.db -as -ar -ru "2014-12-15 15:23:15 -05:00"
```

## 関連情報

[データベースのリカバリ \[915 ページ\]](#)

[ポイントインタイムリカバリ \[917 ページ\]](#)

[チュートリアル: タイムスタンプへのデータベースのリストア \[920 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

[-ruo データベースオプション \[533 ページ\]](#)

### 1.4.2.13 -ruo データベースオプション

ポイントインタイムリカバリの実行時にデータベースをリストアするトランザクションログのオフセットを指定します。

トランザクションログは、記録する操作とイベントに連番を付けます。これらの番号はオフセットと呼ばれます。

## 構文

```
dbsrv17 [ server-options ] database-file -ruo offset-number
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

操作はトランザクションログに逐次記録され、すべての操作がファイルにおける数値のオフセットでユニークに識別されます。それまでデータベースが正常だった時点の操作またはイベントを識別できる場合は、この操作のオフセットまでデータベースをリストアすることができます。



例

次のコマンドは、myDB.db データベースをトラザクションログのオフセット 1245672 にリストアします。

```
dbsrv17 myDB.db -as -ar -ruo 1245672
```

## 関連情報

[データベースのリカバリ \[915 ページ\]](#)

[ポイントインタイムリカバリ \[917 ページ\]](#)

[チュートリアル: トラザクションログのオフセットへのデータベースのリストア \[925 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

[-ru データベースオプション \[532 ページ\]](#)

## 1.4.2.14 -sbx データベースオプション

データベースのディスクサンドボックス機能を制御します。



構文

```
dbsrv17 [ server-options ] database-file -sbx{ + | - } ...
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

このオプションを有効にすると、データベースファイルの読み込み/書き込み操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されます。

このオプションを指定しない場合は、-sbx データベースサーバオプションで指定されたデフォルトのディスクサンドボックス設定が使用されます。

sa\_db\_option システムプロシージャで DiskSandbox オプションを使用すると、データベースの実行中にデータベースのディスクサンドボックス設定を一時的に変更できます。

## 関連情報

[disk\\_sandbox オプション \[696 ページ\]](#)

[-sbx データベースサーバオプション \[470 ページ\]](#)

### 1.4.2.15 -sn データベースオプション

データベースサーバ上で動作する1つのデータベースに代替サーバ名を割り当てます。

#### 構文

```
dbsrv17 [ server-options ] database-file -sn alternate-server-name
```

#### 適用対象

すべてのオペレーティングシステム。

ネットワークデータベースサーバ (dbsrv17) のみ。

#### 備考

-sn データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。

データベースサーバが特定のデータベースサーバに対応する複数のサーバ名を受信するように設定できます。実サーバ名以外のサーバ名を代替サーバ名といいます。代替サーバ名はデータベースサーバ上の特定のデータベースに固有のもので、クライアントは、ServerName (Server) 接続パラメータを使用して代替サーバ名を指定し、そのデータベースに接続できます。

代替サーバ名はネットワーク全体でユニークでなければなりません。データベースをコマンドラインで起動し、指定した代替サーバ名がユニークでない場合、データベースサーバの起動は失敗します。START DATABASE 文を使用して代替サーバ名を指定することもできます。

代替サーバ名を指定するクライアントが接続できるのは、その代替サーバ名が指定されたデータベースだけです。同じデータベースサーバ他のデータベースに対して、接続、作成、停止、削除を行うことはできません。接続パラメータの DBN または DBF を指定する場合は、それぞれデータベース名またはデータベースファイルと一致する必要があります。接続パラメータの DBN または DBF を指定しない場合、データベースはそのサーバのデフォルトデータベースと同じように動作します。

サーバ列挙ユーティリティ (dblocate) では、代替サーバ名も検出されます。

## 例

次のコマンドは、データベースサーバ `myserver` 上のデータベース `satest.db` および `sample.db` を起動します。`-sn` オプションを指定すると、データベースサーバは `sample.db` に接続するときに代替サーバ名として `mysample` を使用します。

```
dbsrv17 -n myserver satest.db sample.db -sn mysample
```

`sample.db` に接続するには、次のいずれかの接続パラメータを使用します。

- `Server=myserver;DBN=sample`
- `Server=mysample`
- `Server=mysample;DBN=sample`

`Server=mysample` を使用して `satest.db` に接続することはできません。

## 関連情報

[データベースミラーリング \[1660 ページ\]](#)

[サーバ列挙ユーティリティ \(dblocate\) \[1145 ページ\]](#)

## 1.4.2.16 -wc データベースオプション

チェックサムがデフォルトで有効になっていない場合に、データベースの書き込み操作でチェックサムが有効になっているかどうかを制御します。

### 構文

```
dbsrv17 [ server-options ] database-file -wc[ + | - ]
```

## 適用対象

すべてのオペレーティングシステム。

## 備考

書き込みチェックサム (`-wc` オプションで有効) とグローバルチェックサム (チェックサムが有効な状態でデータベースを作成) の違いは、`-wc` ではデータベースページがディスクに書き込まれるときにのみチェックサムが行われる点です。ディスクから読み込まれるページは、ページが書き込まれる前にチェックサム値が計算された場合にのみ検証されます。データベースでチェ

ックサムが有効になっている場合、ページが書き込まれるときにすべてのページに対してチェックサムが計算され、ページが読み込まれるときにすべてのページに対してチェックサムが検証されます。

データベースがリムーバブルストレージデバイス (ネットワーク共有や USB デバイスなど) で実行されていることがデータベースサーバで検出されると、データベースサーバでは、すべてのデータベースページに対してグローバルチェックサムが自動的に有効になります。

デフォルトでは、バージョン 10 または 11 の SQL Anywhere で作成されたデータベースではグローバルチェックサムは有効になっていません。バージョン 12 以降のデータベースサーバで SQL Anywhere 11 を使用して作成されたデータベースを起動すると、デフォルトでは、ページがディスクに書き込まれるときにデータベースサーバでページに対して書き込みチェックサムが作成されます (-wc+)。バージョン 12 以降のデータベースではデフォルトでグローバルチェックサムが有効になっています。デフォルトではすべてのデータベースページにチェックサムがあるため、これらのデータベースに対してデータベースサーバのデフォルト値は -wc- になります。デフォルトのチェックサム設定を使用しない場合は、-wc オプションまたは START DATABASE 文を使用してデータベースサーバのチェックサム動作を変更できます。

次の文を実行することによって、データベースがグローバルチェックサムを有効にして作成されたかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'Checksum' );
```

次の文を実行することによってのみ、書き込み I/O 操作に対して書き込みチェックサムが有効になっているかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'WriteChecksum' );
```

## 関連情報

[チェックサムを使用した破損の検出 \[947 ページ\]](#)

[-wc データベースサーバオプション \[499 ページ\]](#)

## 1.4.2.17 -xp データベースオプション

高可用性パートナーや、読み込み専用スケールアウトのルートノードまたはコピーノードに対して、データベースミラーリングを有効にします。

### 構文

```
dbsrv17 [ server-options ] database-file -xp on
```

## 適用対象

すべてのオペレーティングシステム。

ネットワークデータベースサーバ (dbsrv17) のみ。

## 備考

-xp データベースオプションは、`database-file` の後に指定する必要があり、そのデータベースだけに適用されます。トランザクションログを持っていないデータベースをミラーリングモードで起動することはできません。

### on

ミラーリング情報またはスケールアウト情報がデータベースに格納されていても、データベースサーバの起動時に -xp オプションを指定した場合にのみデータベースミラーリングやスケールアウトを使用できます。値 off はサポートされていません。データベースミラーリングとスケールアウトの設定は、次の文を使用してデータベースに定義されます。

- CREATE MIRROR SERVER
- SET MIRROR OPTION

-xp on を指定するときには、-n オプションを使用してミラーリングシステム内のデータベースサーバの名前も指定します。-su オプションを使用して、ユーティリティデータベースのパスワードを指定します。これにより、ユーティリティデータベースを使用してデータベースサーバをシャットダウンしたり、必要に応じてミラーサーバを強制的にプライマリサーバにしたりできるようになります。

### 例

次のコマンドは、ミラーリングシステムに参加できるデータベースサーバで 3 つのデータベースファイルを起動します。

```
dbsrvl7 -n server1 -x tcpip(PORT=6871) -su passwd  
c:¥server1¥one.db -xp on c:¥server1¥two.db -xp on c:¥server1¥three.db -xp on
```

## 関連情報

[読み込み専用スケールアウト \[1733 ページ\]](#)

[データベースミラーリングモード \[1671 ページ\]](#)

[データベースミラーリングシステムで使用するデータベースサーバ \[1677 ページ\]](#)

[データベースミラーリングモード \[1671 ページ\]](#)

[データベースミラーリングシステムの設定 \[1674 ページ\]](#)

[チュートリアル: 読み込み専用のスケールアウトシステムの作成 \[1753 ページ\]](#)

[チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

[チュートリアル: 監視サーバを共有している複数データベースでデータベースミラーリングシステムを作成する \[1711 ページ\]](#)

[-su データベースサーバオプション \[480 ページ\]](#)

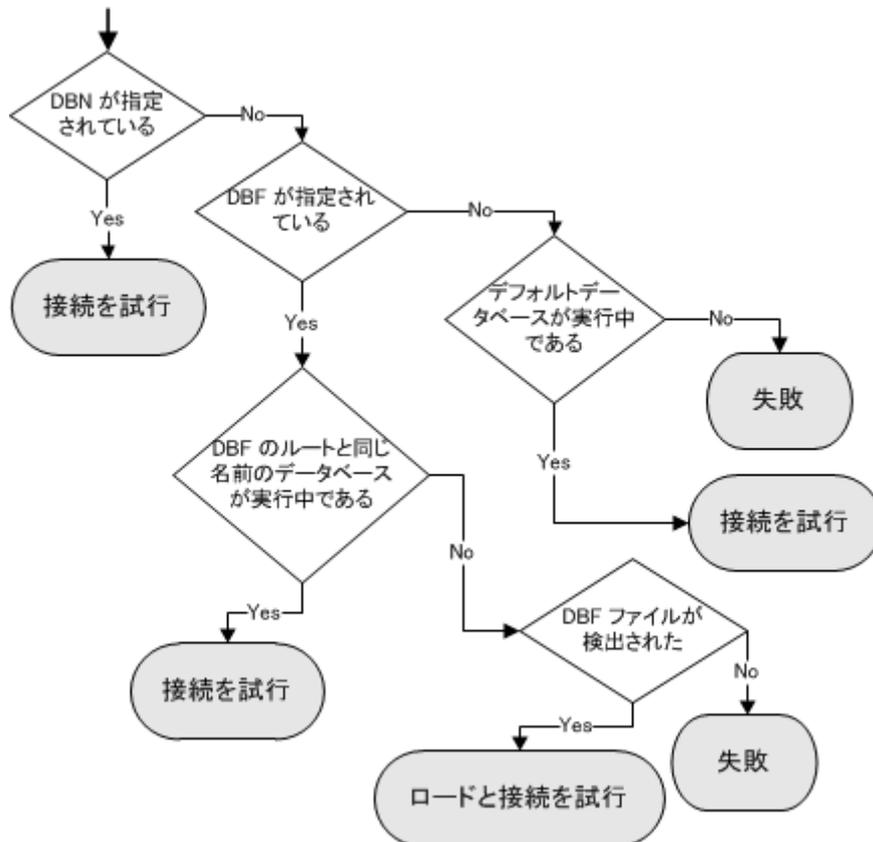
[-xa データベースサーバオプション \[502 ページ\]](#)

[-xf データベースサーバオプション \[505 ページ\]](#)

[-n データベースサーバオプション \[448 ページ\]](#)

### 1.4.3 トラブルシューティング: データベースサーバの検出方法

クライアントは、サーバの検出に成功すると、データベースの検出に移ります。



## 1.5 データベース設定

データベースを設定するためのツールや方法には、いくつかの種類があります。

このセクションの内容:

#### [環境変数 \[540 ページ\]](#)

環境変数は、さまざまなタイプの情報を保存するために使用されます。

#### [ファイルロケーションとインストール設定 \[560 ページ\]](#)

SQL Anywhere をインストールすると、いくつかのディレクトリが作成されます。

#### [タイムゾーンの管理 \[568 ページ\]](#)

ホストコンピュータのタイムゾーンではなく、シミュレートされたタイムゾーンをデータベースで使用できます。

#### [国際言語と文字セット \[571 ページ\]](#)

国際化とは、ソフトウェアの指定言語やソフトウェアを実行しているオペレーティングシステムに関係なく、ソフトウェアが各種言語とそれに適した文字セットに対処できるようにすることです。

#### [ログインポリシー \[613 ページ\]](#)

ログインポリシーは、ユーザのデータベース接続を作成するときに適用されるルールセットで構成されます。

#### [データベースオプション \[622 ページ\]](#)

SET OPTION 文を使用してデータベースオプションを設定します。

#### [接続、データベース、データベースサーバのプロパティ \[838 ページ\]](#)

接続、データベース、およびデータベースサーバのプロパティは、データがデータベースでどのようにアクセスおよび処理されるのかを制御します。

#### [データベースのサイズおよび数に関する物理的な制限 \[891 ページ\]](#)

このトピックは、SQL Anywhere データベースにおけるオブジェクトのサイズと数の物理的制限を示します。一般に、コンピュータのメモリ、CPU、ディスク容量から受ける制限の方が厳しいのが普通です。

## 1.5.1 環境変数

環境変数は、さまざまなタイプの情報を保存するために使用されます。

あらゆる状況ですべての環境変数を設定しなければならないわけではありません。

### Windows での環境変数の設定

SQL Anywhere のインストーラによって、コンピュータのプロパティの次の環境変数が作成または変更されます。PATH および SQLANY17。

データベースサーバに設定された環境変数を参照するには、データベースサーバを起動するときに `-ze` オプションを指定します。データベースサーバの起動後に `xp_getenv` システムプロシージャを使用して環境変数を参照できます。

その他の環境変数は、コンピュータのプロパティを変更して設定できます。また、SET コマンドを使用してコマンドプロンプトやバッチファイルから変更することも可能です。

このセクションの内容:

#### [UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

ソフトウェアのインストール後、システムがアプリケーションの場所を特定して実行できるように、各ユーザはシステムの環境変数を設定します。

#### [sa\\_config.sh、sample\\_config32.sh、sample\\_config64.sh のソース指定 \[Unix および Mac OS X\] \[543 ページ\]](#)

`sa_config.sh/sa_config.csh` および `sample_config32.sh/sample_config64.sh` ファイルのソースを指定します。

#### [DYLD\\_LIBRARY\\_PATH 環境変数 \[Mac OS X\] \[544 ページ\]](#)

Mac OS X 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

#### [LD\\_LIBRARY\\_PATH 環境変数 \[Linux と Solaris\] \[545 ページ\]](#)

Linux と Solaris 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

#### [LIBPATH 環境変数 \[IBM AIX\] \[546 ページ\]](#)

IBM AIX 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

#### [MLTEMPLATE\\_JARS 環境変数 \[546 ページ\]](#)

JDBC ドライバの場所を指定します。この環境変数は Mobile Link テンプレートユーティリティ (mltemplate) によって使用されます。

#### [ODBCHOME 環境変数 \[UNIX\] \[547 ページ\]](#)

.odbc.ini ファイルのロケーションを指定します。

#### [ODBCINI と ODBC\\_INI 環境変数 \[UNIX\] \[547 ページ\]](#)

ODBC データソースを含むシステム情報ファイルのパスと名前を指定します。

#### [PATH 環境変数 \[548 ページ\]](#)

SQL Anywhere 実行プログラムがあるディレクトリのロケーションを指定します。

#### [SACHARSET 環境変数 \[549 ページ\]](#)

SQL Anywhere で使用されるデフォルトの文字セットエンコーディングを指定します。

#### [SADIAGDIR 環境変数 \[550 ページ\]](#)

SQL Anywhere 診断ディレクトリのロケーションを指定します。

#### [SALANG 環境変数 \[551 ページ\]](#)

SQL Anywhere データベースサーバとツールの言語コードを指定します。

#### [SALOGDIR 環境変数 \[552 ページ\]](#)

backup.syb ファイルのロケーションを指定します。

#### [SATMP 環境変数 \[553 ページ\]](#)

データベースサーバとテンポラリディレクトリを必要とする SQL Anywhere コマンドラインユーティリティが使用するテンポラリファイルのロケーションを指定します。

#### [SHLIB\\_PATH 環境変数 \[HP-UX\] \[554 ページ\]](#)

HP-UX 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

#### [SQLANY17 環境変数 \[555 ページ\]](#)

SQL Anywhere 17 を含むディレクトリのロケーションを指定します。

#### [SQLANYSAMP17 環境変数 \[556 ページ\]](#)

SQL Anywhere サンプルディレクトリのロケーションを指定します。

#### [SQLCONNECT 環境変数 \[557 ページ\]](#)

データベースサーバへの接続時に使用する追加の接続パラメータを指定します。

#### [SQLPATH 環境変数 \[558 ページ\]](#)

SQL スクリプトとヘルプファイルのロケーションを指定します。

#### [SQLREMOTE 環境変数 \[558 ページ\]](#)

SQL Remote 中の FILE メッセージリンクのアドレスである、サブフォルダを指定します。

#### [SYBASE 環境変数 \[559 ページ\]](#)

SAP Adaptive Server Enterprise、SAP Open Client、SAP Open Server などの SAP アプリケーションや DSEdit などのユーティリティのインストール用ホームディレクトリを指定します。

[TMP、TMPDIR、TEMP 環境変数 \[559 ページ\]](#)

SQL Anywhere テンポラリファイルのロケーションを指定します。

## 1.5.1.1 UNIX および Mac OS X 環境変数

ソフトウェアのインストール後、システムがアプリケーションの場所を特定して実行できるように、各ユーザはシステムの環境変数を設定します。

システムがアプリケーションの場所を特定して実行するための環境変数の設定に役立つ、次の特殊ファイルがインストーラによって作成されます。sa\_config および sample\_config.sh ファイルです。これらのファイルは、`$SQLANY17/bin32` および `$SQLANY17/bin64` にインストールされます。各ファイルは、必要なユーザ環境変数を設定します。

これらのファイルの変形として .sh と .csh があります。これらのファイル拡張子が示すとおり、1つのタイプは Bourne シェル (sh) とその派生シェル (ksh や bash など) で動作するように設計されています。もう1つのタイプは、C シェル (csh) とその派生シェル (tcsh など) で動作するように設計されています。

一部の文は、各ファイルでコメントアウトされます。システム管理者は、システムの設定に応じてこれらのファイルを編集し、コメントを削除できます。

SQL Anywhere アプリケーションは、いくつかの方法で実行できます。

1. これらのファイルに指定されている環境変数をシステム環境に追加した場合は、X-Window Server などの GUI から起動したり、ターミナルウィンドウでアプリケーション名を入力したりしてアプリケーションを実行できます。
2. ファイルのいずれかのソースを指定してある場合は、ターミナルウィンドウでアプリケーション名を入力することでアプリケーションを実行できます。
3. `$SQLANY17/bin32s` と `$SQLANY17/bin64s` には、SQL Anywhere アプリケーションと同じ名前のスクリプトが含まれています。これらのスクリプトは、適切な環境変数を設定してから、アプリケーションを起動します。このため、対応するスクリプトを実行することで、アプリケーションを実行できます。スクリプトを実行する前にこれらのファイルのソースを指定する必要はありません。

### Mac OS X の Finder の環境変数の設定

sa\_config ファイルによって次の環境変数が設定されます。DYLD\_LIBRARY\_PATH、PATH、および SQLANY17。

sample\_config ファイルによって ODBCINI 環境変数が設定されます。再起動する必要はありません。

ターミナルセッションは、Finder の環境変数を継承しません。

### 関連情報

[sa\\_config.sh、sample\\_config32.sh、sample\\_config64.sh のソース指定 \[Unix および Mac OS X\] \[543 ページ\]](#)

## 1.5.1.2 sa\_config.sh、sample\_config32.sh、sample\_config64.sh のソース指定 [Unix および Mac OS X]

sa\_config.sh/sa\_config.csh および sample\_config32.sh/sample\_config64.sh ファイルのソースを指定します。

### コンテキスト

サンプルを使用するには、sample\_config32.sh または sample\_config64.sh の設定を実行する必要があります。

### 手順

1. 提供されているスクリプトのいずれかを使用して SQL Anywhere の環境変数のソースを指定します。

#### Bourne シェルとその派生シェル

Bourne シェルとその派生シェルでは、このコマンド名は `.` (単一のピリオド) です。たとえば、SQL Anywhere が `/opt/sqlanywhere17` にインストールされている場合、次の文を使用して `sa_config.sh` のソースを指定します (32 ビット環境では `bin32` を使用してください)。

```
. /opt/sqlanywhere17/bin64/sa_config.sh
```

Mac の場合は、次のコマンドを実行して `sa_config.sh` のソースを指定します。

```
. /Applications/SQLAnywhere17/System/bin64/sa_config.sh
```

#### C シェルとその派生シェル

C シェルとその派生シェルの場合、コマンドは `source` です。たとえば、SQL Anywhere が `/opt/sqlanywhere17` にインストールされている場合、次のコマンドを使用して `sa_config.csh` のソースを指定します (32 ビット環境では `bin32` を使用してください)。

```
source /opt/sqlanywhere17/bin64/sa_config.csh
```

2. サンプルを設定します。SQL Anywhere17 のサンプルを使用する前に、サンプルで使用する SQL Anywhere のビット数に対応した設定スクリプトを実行してください。

たとえば、32 ビットのソフトウェアでサンプルを使用できるように設定するには、次のように実行します。

```
. /opt/sqlanywhere17/samples/sample_config32.sh
```

Mac の場合、32 ビットのソフトウェアでサンプルを使用できるように設定するには、次のように実行します。

```
. /Applications/SQLAnywhere17/samples/sample_config32.sh
```

## 結果

ファイルのソースが指定されます。

### 1.5.1.3 DYLD\_LIBRARY\_PATH 環境変数 [Mac OS X]

Mac OS X 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

#### 構文

```
DYLD_LIBRARY_PATH=path-list
```

#### デフォルト

```
/Applications/SQLAnywhere17/System/lib64:/Applications/SQLAnywhere17/System/lib32
```

#### 備考

インストーラによって作成される sa\_config.sh ファイルと sa\_config.csh ファイルは、これを含むすべての環境変数を作成または変更するスクリプトです。

#### 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[LD\\_LIBRARY\\_PATH 環境変数 \[Linux と Solaris\] \[545 ページ\]](#)

[LIBPATH 環境変数 \[IBM AIX\] \[546 ページ\]](#)

[SHLIB\\_PATH 環境変数 \[HP-UX\] \[554 ページ\]](#)

## 1.5.1.4 LD\_LIBRARY\_PATH 環境変数 [Linux と Solaris]

Linux と Solaris 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

### 構文

```
LD_LIBRARY_PATH=path-list
```

### デフォルト

#### 32 ビット Linux

```
$$SQLANY17/lib32  
$$SQLANY17/lib64  
$$SQLANY17/bin32/jre180/lib/i386/server
```

#### 64 ビット Linux および Solaris

```
$$SQLANY17/lib64  
$$SQLANY17/lib32  
$$SQLANY17/bin64/jre180/lib/amd64/server  
$$SQLANY17/bin64/jre180/lib/amd64
```

### 備考

インストーラによって作成される sa\_config.sh ファイルと sa\_config.csh ファイルは、これを含むすべての環境変数を作成または変更するスクリプトです。

### 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[DYLD\\_LIBRARY\\_PATH 環境変数 \[Mac OS X\] \[544 ページ\]](#)

[LIBPATH 環境変数 \[IBM AIX\] \[546 ページ\]](#)

[SHLIB\\_PATH 環境変数 \[HP-UX\] \[554 ページ\]](#)

## 1.5.1.5 LIBPATH 環境変数 [IBM AIX]

IBM AIX 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

### 構文

```
LIBPATH=path-list
```

### デフォルト

- /usr/lpp/sqlanywhere17/lib32 (32 ビットのプラットフォーム)
- /usr/lpp/sqlanywhere17/lib64 (64 ビットのプラットフォーム)

### 備考

インストーラによって作成される sa\_config.sh ファイルと sa\_config.csh ファイルは、これを含むすべての環境変数を作成または変更するスクリプトです。

### 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[DYLD\\_LIBRARY\\_PATH 環境変数 \[Mac OS X\] \[544 ページ\]](#)

[LD\\_LIBRARY\\_PATH 環境変数 \[Linux と Solaris\] \[545 ページ\]](#)

[SHLIB\\_PATH 環境変数 \[HP-UX\] \[554 ページ\]](#)

## 1.5.1.6 MLTEMPLATE\_JARS 環境変数

JDBC ドライバの場所を指定します。この環境変数は Mobile Link テンプレートユーティリティ (mltemplate) によって使用されます。

### 構文

```
MLTEMPLATE_JARS=path-to-JDBC-driver
```

## 備考

この環境変数は、SQL Anywhere 以外の統合データベースのデータベーススキーマをインポートするために、mltemplate ユーティリティによって使用されます。

SQL Anywhere 以外のデータベースの JDBC ドライバが収録された JAR ファイルへのパスを指定します。

SQL Anywhere JDBC ドライバのパスは指定する必要がありません。

## 1.5.1.7 ODBC\_HOME 環境変数 [UNIX]

.odbc.ini ファイルのロケーションを指定します。

### 構文

```
ODBC_HOME=odbc-ini-directory
```

## 備考

.odbc.ini ファイルは、ODBC データソースを含むシステム情報ファイルです。このファイルの名前が .odbc.ini ではない場合は、ODBCINI または ODBC\_INI 環境変数を使用してロケーションを指定する必要があります。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[UNIX での ODBC データソース \[121 ページ\]](#)

[ODBCINI と ODBC\\_INI 環境変数 \[UNIX\] \[547 ページ\]](#)

## 1.5.1.8 ODBCINI と ODBC\_INI 環境変数 [UNIX]

ODBC データソースを含むシステム情報ファイルのパスと名前を指定します。

### 構文

```
ODBCINI=odbc-ini-file
```

```
ODBC_INI=odbc-ini-file
```

## 備考

システム情報ファイルは、これらのいずれかの環境変数を使用して指定している場合は、`.odbc.ini` という名前でもかまいません。2つの環境変数が用意されているのは、他製品との互換性を確保するためです。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[UNIX での ODBC データソース \[121 ページ\]](#)

[ODBCHOME 環境変数 \[UNIX\] \[547 ページ\]](#)

## 1.5.1.9 PATH 環境変数

SQL Anywhere 実行プログラムがあるディレクトリのロケーションを指定します。

### 構文

```
PATH=path-list
```

## デフォルト

次のパスは、対応するコンポーネントがインストールされた場合にのみ追加されます。

オペレーティングシステム	デフォルトのロケーション
Windows (32 ビット)	<code>C:¥Program Files¥SQL Anywhere 17¥bin32</code>
Windows (64 ビット)	<code>C:¥Program Files¥SQL Anywhere 17¥bin64;C:¥Program Files¥SQL Anywhere 17¥bin32</code>
Mac OS X (64 ビット)	<code>/Applications/SQLAnywhere17/System/bin64</code>
IBM AIX (64 ビット)	<code>/usr/lpp/sqlanywhere17/bin64</code>
Solaris SPARC (64 ビット版)	<code>/opt/sqlanywhere17/bin64/jre180/bin/sparcv9:/opt/sqlanywhere17/bin64:/opt/sqlanywhere17/bin32</code>
Linux (32 ビット)	<code>/opt/sqlanywhere17/bin32/jre180/bin:/opt/sqlanywhere17/bin32</code>
Linux (64 ビット)	<code>/opt/sqlanywhere17/bin64/jre180/bin:/opt/sqlanywhere17/bin64:/opt/sqlanywhere17/bin32</code>

オペレーティングシステム	デフォルトのロケーション
その他の UNIX オペレーティングシステム (64 ビット)	<code>/opt/sqlanywhere17/bin64</code>

## 備考

Windows では、PATH 環境変数の設定がインストーラによって変更され、SQL Anywhere 実行プログラムの保存先ディレクトリが追加されます。

UNIX では、これを含むすべての環境変数を作成または変更するスクリプトとして、`sa_config.sh` ファイルと `sa_config.csh` ファイルがインストーラによって作成されます。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[sa\\_config.sh、sample\\_config32.sh、sample\\_config64.sh のソース指定 \[Unix および Mac OS X\] \[543 ページ\]](#)

## 1.5.110 SACHARSET 環境変数

SQL Anywhere で使用されるデフォルトの文字セットエンコーディングを指定します。

### 構文

```
SACHARSET=charset-label
```

## 備考

`charset-label` は文字セットエンコーディングです。

`dbinit -le` オプションを使用すると、データベースで利用可能なすべての文字セットエンコーディングの一覧を取得できます。

SACHARSET が指定されない場合、オペレーティングシステムの文字セットエンコーディングがデフォルトとして使用されます。

すべてのコンポーネントで SACHARSET 環境変数が使用されるわけではありません。Embedded SQL 接続と ODBC 接続では、この環境変数が使用されます。初期化ユーティリティ (`dbinit`) などのデータベースツールでも使用されます。Interactive SQL (`dbisql`) を含む JDBC 接続では使用されません。.NET 接続でも使用されません。

## i 注記

SACHARSET を設定することは、クライアント/データベース接続に対して CharSet (CS) 接続パラメータを指定することと同じではありません。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[推奨文字セットと照合 \[607 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

## 1.5.1.11 SADIAGDIR 環境変数

SQL Anywhere 診断ディレクトリのロケーションを指定します。

### 構文

```
SADIAGDIR=diagnostic-information-directory
```

## デフォルト

この環境変数はデフォルトで設定されていません。

## 備考

この環境変数は、SQL Anywhere 診断ディレクトリのデフォルトのロケーションに優先されます。SQL Anywhere プログラムは、ソフトウェアに致命的なエラーがある場合に、このディレクトリを使用してレポートを格納します。

SQL Anywhere では、クラッシュレポートと機能診断情報が診断ディレクトリに保存されます。SQL Anywhere によってクラッシュレポートが書き込まれる診断ディレクトリのロケーションは、SADIAGDIR 環境変数の設定によって決まります。

環境変数で指定されたディレクトリが存在しない場合、データベースサーバは環境変数が設定されていないものとして処理を実行します。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[トラブルシューティング: エラーのレポート \[973 ページ\]](#)

[サポートユーティリティ \(dbsupport\) \[1165 ページ\]](#)

### 1.5.1.12 SALANG 環境変数

SQL Anywhere データベースサーバとツールの言語コードを指定します。

#### 構文

```
SALANG=language-code
```

## 備考

`language-code` は言語を表す 2 文字の組み合わせです。たとえば、**SALANG=DE** を設定すると、デフォルトの言語がドイツ語に設定されます。

次の方法で最初に値を返すものが、メッセージテキストのデフォルトの言語を決定します。

1. SALANG 環境変数をチェックします。
2. (Windows の場合) インストール時、または `dblang.exe` によって設定されたレジストリをチェックします。
3. オペレーティングシステムに対して言語情報を問い合わせます。
4. 言語情報が設定されていない場合は、英語がデフォルトになります。

## 関連情報

[ロケール言語 \[588 ページ\]](#)

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[言語選択ユーティリティ \(dblang\) \[1131 ページ\]](#)

[インストール時のレジストリ設定 \[568 ページ\]](#)

## 1.5.1.13 SALOGDIR 環境変数

backup.syb ファイルのロケーションを指定します。

### 構文

```
SALOGDIR=directory-name
```

### 備考

SALOGDIR 環境変数が設定されている場合、バックアップ履歴ファイル backup.syb を書き込むことができるディレクトリのパスが含まれることが想定されます。このファイルは BACKUP 文または RESTORE 文を実行するたびに更新されます。

Windows では、次のうち最初の書き込み可能なロケーションに backup.syb ファイルが書き込まれます。

1. SALOGDIR 環境変数
2. %ALLUSERSPROFILE%¥SQL Anywhere17
3. 現在のドライブのルートディレクトリ。
4. 現在のディレクトリ。

Windows CE では、次のうち最初の書き込み可能なロケーションに backup.syb ファイルが書き込まれます。

1. SALOGDIR 環境変数
2. データベースサーバが起動されたディレクトリ。
3. ルートディレクトリ。
4. 現在のディレクトリ。

UNIX では、次のうち最初の書き込み可能なロケーションに backup.syb ファイルが書き込まれます。

1. SALOGDIR 環境変数
2. \$HOME/.sqlanywhere17
3. 現在のディレクトリ。

### 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

## 1.5.1.14 SATMP 環境変数

データベースサーバとテンポラリディレクトリを必要とする SQL Anywhere コマンドラインユーティリティが使用するテンポラリファイルのロケーションを指定します。

### 構文

```
SATMP=directory-name
```

### 備考

SQL Anywhere では、データベースサーバ関連のテンポラリファイル (すべてのプラットフォームで作成) と通信関連のテンポラリファイル (UNIX のクライアントとサーバでのみ作成) の 2 種類のテンポラリファイルが作成されます。

SATMP 環境変数には、データベースサーバとテンポラリディレクトリを必要とする SQL Anywhere コマンドラインユーティリティが使用するテンポラリファイルのロケーションが指定されます。このファイルは、データベースサーバをサービスとして実行する場合に役立ちます。テンポラリファイルをほかのプログラムがアクセスできないディレクトリに保存できるからです。

データベースサーバの起動時にテンポラリファイルのロケーションが `-dt` オプションで指定されていない場合は、データベースサーバによって SATMP 環境変数の値が確認され、テンポラリファイルの配置場所が決まります。SATMP 環境変数が存在しない場合は、TMP、TMPDIR、TEMP のうち最初に存在する環境変数が使用されます。Windows では、これらの環境変数がいずれも存在しない場合、現在のディレクトリが使用されます。UNIX では、これらの環境変数のいずれも存在しない場合、`/tmp` が使用されます。

UNIX では、共有メモリを介して接続する場合、クライアントとデータベースサーバの両方で SATMP を同じ値に設定する必要があります。

### i 注記

SQL Anywhere バージョン 9 以前では、環境変数 ASTMP は SATMP と同じです。共有メモリを使用してバージョン 9 およびバージョン 10 ソフトウェアに接続している場合は、SATMP 環境変数と ASTMP 環境変数を設定して、テンポラリディレクトリの (同じ) ロケーションを指定する必要があります。

データベースサーバや UNIX 上のクライアントで作成されたテンポラリファイルのパーミッションを制限する場合は、この環境変数に、次のリストにないディレクトリを設定します。

- `/tmp`
- `/tmp/.SQLAnywhere`
- TMP 環境変数の値 (設定している場合)
- TMPDIR 環境変数の値 (設定している場合)
- TEMP 環境変数の値 (設定している場合)
- 上記のいずれかのディレクトリを示すシンボリックリンク

SATMP が上記のリストにないディレクトリに設定されている場合、データベースサーバは、現在のユーザが所有し、パーミッションが 770、707、700 のいずれかに設定されているディレクトリで指定のディレクトリパスを検索します。パーミッションがそれ以外の値に設定されている場合は、パーミッションを 777 に設定してファイルを作成します。データベースサーバは、検出さ

れた各ディレクトリについて、該当するパーミッション（それぞれ other、group、other+group に対応）をテンポラリファイルの作成に使用するパーミッションマスクから削除します。

### 警告

上記のリストにないディレクトリを SATMP に設定すると、別の UNIX アカウントを使用するユーザが共有メモリを介してデータベースサーバに接続できなくなる場合があります。

UNIX では、共有メモリ接続を保護する場合、SATMP 環境変数をユニークなディレクトリに設定できます。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[-dt データベースサーバオプション \[400 ページ\]](#)

[TMP、TMPDIR、TEMP 環境変数 \[559 ページ\]](#)

[トラブルシューティング: データベースサーバの起動 \[974 ページ\]](#)

## 1.5.1.15 SHLIB\_PATH 環境変数 [HP-UX]

HP-UX 上で SQL Anywhere アプリケーションが必要とするライブラリの実行時の検索場所になるディレクトリを指定します。

### 構文

```
SHLIB_PATH=path-list
```

## デフォルト

- /opt/sqlanywhere17/lib32 (32 ビットのプラットフォーム)
- /opt/sqlanywhere17/lib64 (64 ビットのプラットフォーム)

## 備考

インストーラによって作成される sa\_config.sh ファイルと sa\_config.csh ファイルは、これを含むすべての環境変数を作成または変更するスクリプトです。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[DYLD\\_LIBRARY\\_PATH 環境変数 \[Mac OS X\] \[544 ページ\]](#)

[LD\\_LIBRARY\\_PATH 環境変数 \[Linux と Solaris\] \[545 ページ\]](#)

[LIBPATH 環境変数 \[IBM AIX\] \[546 ページ\]](#)

### 1.5.1.16 SQLANY17 環境変数

SQL Anywhere 17 を含むディレクトリのロケーションを指定します。

#### 構文

```
SQLANY17=directory-name
```

#### デフォルト

オペレーティングシステム	ロケーション
Windows	<i>C:¥Program Files¥SQL Anywhere 17</i>
IBM AIX	<i>/usr/lpp/sqlanywhere17</i>
Mac OS X	<i>/Applications/SQLAnywhere17/System</i>
その他の UNIX オペレーティングシステム	<i>/opt/sqlanywhere17</i>

#### 備考

この環境変数は必ず設定してください。これには、いくつかの理由があります。その 1 つとして、サンプルが SQL Anywhere アプリケーションを探すときに、この環境変数が必要となることがあります。

Windows では、インストーラによって SQLANY17 環境変数が作成されます (Windows レジストリに格納されます)。

UNIX では、これを含むすべての環境変数を作成または変更するスクリプトとして、sa\_config.sh ファイルと sample\_config32.csh ファイルがインストーラによって作成されます。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

sa\_config.sh、sample\_config32.sh、sample\_config64.sh のソース指定 [Unix および Mac OS X] [543 ページ]

## 1.5.1.17 SQLANYSAMP17 環境変数

SQL Anywhere サンプルディレクトリのロケーションを指定します。

### 構文

```
SQLANYSAMP17=directory-name
```

### デフォルト

オペレーティングシステム	デフォルトのロケーション
Windows	C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples
UNIX (シングルユーザインストール)	\$SQLANY17/samples

### 備考

Windows では、インストーラによって %SQLANYSAMP17% 環境変数が作成されます (Windows レジストリに格納されます)。Windows エクスプローラーウィンドウで SQL Anywhere サンプルディレクトリを開くには、**スタート** > **すべてのプログラム** > **SQL Anywhere17** > **サンプルアプリケーションとプロジェクト** をクリックします。

Mac OS X を含む UNIX では、sample\_config32.sh スクリプトまたは sample\_config64.sh スクリプトを元にして、SQL Anywhere サンプルのユーザごとのコピーを作成し、\$SQLANYSAMP17 環境変数のロケーションを、新しく作成したユーザごとのコピーに設定します。

### 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[SQLANY17 環境変数 \[555 ページ\]](#)

## 1.5.1.18 SQLCONNECT 環境変数

データベースサーバへの接続時に使用する追加の接続パラメータを指定します。

### 構文

```
SQLCONNECT=parameter=value; ...
```

### 備考

この文字列は、`parameter = value` 形式のパラメータ設定をセミコロンで区切ったリストで指定します。

接続文字列ですでに指定されている場合、SQLCONNECT 環境変数によって指定された接続パラメータは使用されません。

### 警告

パスワードはプレーンテキストなので、それを SQLCONNECT 環境変数に含めることはセキュリティリスクとなります。

SQLCONNECT 環境変数を使用する代わりに、接続文字列情報をファイルに挿入した後、SQL Anywhere ツールのコマンドラインで `@filename` を使用できます。

### 注記

SQL Anywhere ツールの 1 つを使用するときには、これが唯一のオプションです。

SQLCONNECT 環境変数で指定された接続パラメータによって、DSN 内の各パラメータが上書きされます。

### 例

次の例では、接続文字列情報をファイルに挿入した後、SQL Anywhere ツールのコマンドラインで、そのファイル名を使用します。

```
echo -c uid=dba;pwd=mypass > connect_info
dbping @connect_info
```

次の一連のコマンドでは、ユーザ (Henry ではなく) George のパスワード "mypass" を使用して、Sales データベースへの接続を試みます。

```
dbdsn -w mydsn -c uid=henry
set sqlconnect=uid=george
dbping -c dsn=mydsn;pwd=mypass;dbn=sales -d
```

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

### 1.5.1.19 SQLPATH 環境変数

SQL スクリプトとヘルプファイルのロケーションを指定します。

#### 構文

```
SQLPATH=path-list
```

#### 備考

Interactive SQL は、システムパスを検索する前に、SQLPATH に指定されているディレクトリでスクリプトファイルとヘルプファイルを検索します。

#### 関連情報

### 1.5.1.20 SQLREMOTE 環境変数

SQL Remote 中の FILE メッセージリンクのアドレスである、サブフォルダを指定します。

#### 構文

```
SQLREMOTE=path
```

#### 備考

SQL Remote 中の FILE メッセージリンクのアドレスは、SQLREMOTE 環境変数のサブフォルダです。この環境変数には共有ディレクトリを指定してください。

Windows オペレーティングシステムでは、SQLREMOTE 環境変数を設定する代わりに、SQL Remote¥Directory レジストリエントリを適切なルートディレクトリに設定することもできます。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

### 1.5.1.21 SYBASE 環境変数

SAP Adaptive Server Enterprise、SAP Open Client、SAP Open Server などの SAP アプリケーションや DSEdit などのユーティリティのインストール用ホームディレクトリを指定します。

#### 構文

```
SYBASE==directory-name
```

## 備考

この環境変数が必要になるのは、その他の SAP アプリケーションを使用する場合のみです。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

### 1.5.1.22 TMP、TMPDIR、TEMP 環境変数

SQL Anywhere テンポラリファイルのロケーションを指定します。

#### 構文

```
TMP=path
```

```
TMPDIR=path
```

```
TEMP=path
```

## 備考

SQL Anywhere ソフトウェアは、データベースサーバが起動すると同時に作成され、データベースサーバが停止すると消去されるテンポラリファイルのような、さまざまな操作のためのテンポラリファイルを作成する場合があります。その名前が示すように、テンポラリファイルとはデータベースサーバの起動中に一時的に情報を保持するために使用されます。テンポラリファイルには、セッション間にわたって維持する必要がある情報は格納されません。

テンポラリファイルは、環境変数 TMP、TMPDIR、TEMP のいずれかで指定されたディレクトリに作成されます。これらの環境変数が重複指定されている場合は、TMP、TMPDIR、TEMP の中で最初に出現したものが採用されます。SATMP 環境変数を設定して、データベースサーバとテンポラリディレクトリを必要とする SQL Anywhere コマンドラインユーティリティが使用するテンポラリファイルのロケーションを指定することもできます。

データベースサーバが使用するテンポラリファイルのロケーションは、データベースサーバの起動時に -dt サーバオプションを使用して指定できます。テンポラリファイルのロケーションを指定せずにデータベースサーバを起動した場合は、SQL Anywhere によって環境変数が次の順にチェックされます。SATMP TMP、TMPDIR、TEMP 環境変数が定義されていない場合、SQL Anywhere によって、Windows の場合は現在のディレクトリ、UNIX の場合は /tmp ディレクトリにテンポラリファイルが作成されます。

UNIX では、テンポラリディレクトリは共有メモリ接続を確立するために使用されます。

### i 注記

#### UNIX における旧バージョンのソフトウェアでの共有メモリ接続の使用

SQL Anywhere バージョン 9 以前では、環境変数 ASTMP は SATMP と同じです。共有メモリを使用してバージョン 9 およびバージョン 12 ソフトウェアに接続している場合は、SATMP 環境変数と ASTMP 環境変数を設定して、テンポラリファイルのロケーションを指定する必要があります。

## 関連情報

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[-dt データベースサーバオプション \[400 ページ\]](#)

[SATMP 環境変数 \[553 ページ\]](#)

[トラブルシューティング: データベースサーバの起動 \[974 ページ\]](#)

## 1.5.2 ファイルロケーションとインストール設定

SQL Anywhere をインストールすると、いくつかのディレクトリが作成されます。

このディレクトリ内のファイルの中には、不可欠なものもあり、そうでないものもあります。

1つの製品として入手した場合も、他の製品の一部として入手した場合も、SQL Anywhere ソフトウェアは単一のインストールディレクトリにインストールされます。SQLANY17 環境変数は、インストールディレクトリのロケーションを指定します。

## SQL Anywhere インストールディレクトリ

SQL Anywhere のインストールディレクトリには、次のようなものがあります。

はじめにお読みください

"はじめにお読みください" のファイルには、最新の情報が記載されています。

インストールディレクトリの下には、次のようなディレクトリがあります。

実行プログラムディレクトリ

オペレーティングシステムプラットフォームごとに別々のディレクトリがあり、それぞれに設定ファイルとコンテキスト別のヘルプファイルが保存されています。

Windows の場合、これらのファイルは、bin32、または bin64 ディレクトリにインストールされます。UNIX を使用している場合は、これらのファイルは bin32 または bin64 ディレクトリと lib32 または lib64 ディレクトリにインストールされます。

使用しているオペレーティングシステムのバージョンに必要なディレクトリだけがあります。

java ディレクトリ

このディレクトリには JAR ファイルが保存されます。

scripts ディレクトリ

scripts ディレクトリには、データベース管理ユーティリティによって使用され、サンプルとしても使用される SQL スクリプトがあります。

SDK¥Include ディレクトリ

SDK¥Include ディレクトリには、SQL Anywhere 対応の C/C++ アプリケーションを開発するためのヘッダファイルが保存されます。UNIX では、このディレクトリは include と呼ばれます。

## UNIX ファイルのロケーション

言語リソースは res ディレクトリにインストールされ、共有オブジェクトは lib32 または lib64 ディレクトリにインストールされます。

このセクションの内容:

[サンプルディレクトリ \[562 ページ\]](#)

SQL Anywhere をインストールするとき、サンプルのインストール先ディレクトリを選択できます。インストーラにより、%SQLANY%SAMP17% 環境変数が作成され、この場所が特定されます。

[SQL Anywhere のファイルを配置する方法 \[562 ページ\]](#)

クライアントライブラリとデータベースサーバは、主に次の 2 つの理由からファイルを配置する必要があります。

[レジストリと INI ファイル \[565 ページ\]](#)

Windows オペレーティングシステムでは、SQL Anywhere は複数のレジストリ設定を使用します。UNIX の場合、これらの設定は初期化ファイルに保存されています。

## 関連情報

[sa\\_config.sh、sample\\_config32.sh、sample\\_config64.sh のソース指定 \[Unix および Mac OS X\] \[543 ページ\]](#)

[SQLANYAMP17 環境変数 \[556 ページ\]](#)

[SQLANY17 環境変数 \[555 ページ\]](#)

### 1.5.2.1 サンプルディレクトリ

SQL Anywhere をインストールするとき、サンプルのインストール先ディレクトリを選択できます。インストーラにより、%SQLANYAMP17% 環境変数が作成され、この場所が特定されます。

## 関連情報

[sa\\_config.sh、sample\\_config32.sh、sample\\_config64.sh のソース指定 \[Unix および Mac OS X\] \[543 ページ\]](#)

[SQLANYAMP17 環境変数 \[556 ページ\]](#)

[SQLANY17 環境変数 \[555 ページ\]](#)

### 1.5.2.2 SQL Anywhere のファイルを配置する方法

クライアントライブラリとデータベースサーバは、主に次の 2 つの理由からファイルを配置する必要があります。

- SQL Anywhere を実行するためには、DLL と初期化ファイルが必要です。不正な DLL が検索されると、バージョンミスマッチエラーが発生する可能性があります。
- INSTALL JAVA や LOAD TABLE など、SQL 文で指定して、ランタイムに検索する必要があるものがあります。

ファイル名を使用する SQL 文の例は次のとおりです。

#### INSTALL JAVA 文

Java クラスを保存するファイル名です。

#### LOAD TABLE 文と UNLOAD TABLE 文

データがロードまたはアンロードされるファイル名です。

#### CREATE DATABASE 文

この文と、ファイルを作成できる同様の文には、ファイル名が必要です。

SQL Anywhere は簡易アルゴリズムを使用してファイルを検索する場合があります。それ以外の場合、より広範囲の検索が行われます。

このセクションの内容:

[簡単なファイル検索 \[563 ページ\]](#)

SQLを使用してファイル検索を実行する場合、ファイル名はデータベースサーバの現在作業中のディレクトリに対する相対名として解釈されます。

#### Windows での広範囲なファイル検索 [563 ページ]

Windows では、データベースサーバと管理ユーティリティを含む SQL Anywhere プログラムは、DLL や共有ライブラリなど、必要なファイルをより広範囲に検索できます。

#### UNIX での広範囲なファイル検索 [564 ページ]

UNIX では、データベースサーバと管理ユーティリティを含む SQL Anywhere プログラムは、DLL や共有ライブラリなど、必要なファイルをより広範囲に検索できます。

### 1.5.2.2.1 簡単なファイル検索

SQLを使用してファイル検索を実行する場合、ファイル名はデータベースサーバの現在作業中のディレクトリに対する相対名として解釈されます。

また、データベースサーバが起動され、データベースファイル名 (DatabaseFile (DBF) パラメータ) が提供されると、パスは現在作業中のディレクトリに対する相対名として解釈されます。

### 1.5.2.2.2 Windows での広範囲なファイル検索

Windows では、データベースサーバと管理ユーティリティを含む SQL Anywhere プログラムは、DLL や共有ライブラリなど、必要なファイルをより広範囲に検索できます。

この場合、SQL Anywhere プログラムは次の順序でファイルを検索します。

1. 現在のモジュールのディレクトリ (現在実行中のプログラムの実行ファイルまたはライブラリファイルがあるディレクトリ)。
2. 現在の実行ファイルのディレクトリ (現在実行中のプログラムの実行ファイルがあるディレクトリ)。
3. SQL Anywhere インストールパスは単一のディレクトリであり、定義されている場合は %SQLANY17% 環境変数で指定されます。
4. フルパスが指定されます。パスが何も指定されない場合は、現在の作業ディレクトリが指定されます。
5. %APPDATA%\SQL Anywhere 17 ディレクトリ。
6. %ALLUSERSPROFILE%\SQL Anywhere17 ディレクトリ。
7. ロケーションレジストリエントリ。
8. システムに応じたディレクトリ。このディレクトリには、一般的なオペレーティングシステムファイルが格納されているディレクトリが含まれます。たとえば Windows では、Windows ディレクトリや Windows\system32 ディレクトリになります。
9. PATH ディレクトリ。システムパスとユーザパスのディレクトリを検索します。

#### i 注記

Windows では、SQL Anywhere によって、前述のリストのディレクトリからの相対パスで次のパスが検索されます。

1. .
2. ..
3. .\bin32 および ..\bin32 (32ビットプログラムのみ)

4. `./bin64` および `../bin64` (64ビットプログラムのみ)
5. `./java` (Java 関連のファイル)
6. `../java` (Java 関連のファイル)
7. `./scripts` (SQL スクリプトファイル)
8. `../scripts` (SQL スクリプトファイル)

### 1.5.2.2.3 UNIX での広範囲なファイル検索

UNIX では、データベースサーバと管理ユーティリティを含む SQL Anywhere プログラムは、DLL や共有ライブラリなど、必要なファイルをより広範囲に検索できます。

この場合、SQL Anywhere プログラムは次の順序でファイルを検索します。

1. 実行プログラムのパス (判別できる場合)。
2. SQL Anywhere インストールパスは単一のディレクトリであり、定義されている場合は `$SQLANY17` 環境変数で指定されます。
3. パスなし (現在の作業ディレクトリ)。
4. `$HOME/.sqlanywhere17` ディレクトリ。
5. PATH 環境変数
6. LIBPATH 環境変数
  - LD\_LIBRARY\_PATH (Linux と Solaris)
  - LD\_LIBRARY\_PATH と SHLIB\_PATH (HP-UX)
  - LIBPATH (IBM AIX)
  - DYLD\_LIBRARY\_PATH (Mac OS X)

#### i 注記

UNIX では、SQL Anywhere によって、前述のリストのディレクトリからの相対パスで次のパスが検索されます。

1. `.`
2. `..`
3. `./bin32` および `../bin32` (32ビットプログラムのみ)
4. `./bin64` および `../bin64` (64ビットプログラムのみ)
5. `./lib32` および `../lib32` (32ビットプログラムのライブラリファイルのみ)
6. `./lib64` および `../lib64` (64ビットプログラムのライブラリファイルのみ)
7. `./java` (Java 関連のファイル)
8. `../java` (Java 関連のファイル)
9. `./scripts` (SQL スクリプトファイル)
10. `../scripts` (SQL スクリプトファイル)
11. `./res` (`.res` ファイル)
12. `../res` (`.res` ファイル)
13. `./tix` (`.tix` ファイル)
14. `../tix` (`.tix` ファイル)

## 1.5.2.3 レジストリと INI ファイル

Windows オペレーティングシステムでは、SQL Anywhere は複数のレジストリ設定を使用します。UNIX の場合、これらの設定は初期化ファイルに保存されています。

これらの設定はソフトウェアのインストール中に行われるので、通常はユーザがレジストリや初期化ファイルにアクセスする必要はありません。ここでは、オペレーティング環境を変更するユーザのために説明します。

SQL Anywhere で使用される .ini ファイルは、ファイル非表示ユーティリティ (dbfhide) を使用して暗号化できます。

### ⚠ 警告

SQL Anywhere データソースだけを使用している場合を除き、UNIX ではファイル非表示ユーティリティ (dbfhide) を使用して、システム情報ファイル (デフォルトのファイル名は .odbc.ini) を暗号化しないでください。

他のデータソース (Mobile Link 同期など) を使用する予定の場合、システム情報ファイルの内容を難読化すると、他のドライバが正しく機能しなくなることがあります。

このセクションの内容:

#### [現在のユーザとローカルマシン設定 \[565 ページ\]](#)

オペレーティングシステムによっては、2つのレベルのシステム設定 (ユーザ設定とローカルマシン設定) があります。

#### [.ini ファイルの内容を非表示にする \[566 ページ\]](#)

暗号化を使用して、.ini ファイル内のパスワードやその他の情報をファイル非表示ユーティリティ (dbfhide) によって保護します。

#### [レジストリ構造 \[567 ページ\]](#)

Windows では、レジストリエディタを使用してレジストリに直接アクセスできます。

#### [インストール時のレジストリ設定 \[568 ページ\]](#)

Windows では、インストールプログラムによって HKEY\_LOCAL\_MACHINE¥Software¥SAP レジストリキーに設定が作成されます。

### 1.5.2.3.1 現在のユーザとローカルマシン設定

オペレーティングシステムによっては、2つのレベルのシステム設定 (ユーザ設定とローカルマシン設定) があります。

一部の設定は個々のユーザに固有であり、そのユーザがログオンしたときだけ使用できます。これらの設定を現在のユーザ設定と呼びます。また、コンピュータ全体に関連し、すべてのユーザが使用できるものを、ローカルマシン設定と呼びます。ローカルマシン設定を変更するには、コンピュータの管理者パーミッションを取得する必要があります。

SQL Anywhere は、現在のユーザ設定とローカルマシン設定の両方を許可します。たとえば、Windows では、それぞれ HKEY\_CURRENT\_USER と HKEY\_LOCAL\_MACHINE のレジストリキーに保存されます。

## 現在のユーザを優先

現在のユーザとローカルマシンレジストリの両方に設定がある場合は、現在のユーザ設定がローカルマシン設定より優先されます。

## ローカルマシン設定が必要なとき

SQL Anywhere プログラムをサービスとして実行する場合は、設定がローカルマシンレベルで行われていることを確認してください。

サービスは、コンピュータ全体を停止しないかぎり、コンピュータからログオフしても特別なアカウントで実行を継続できます。サービスは、個々のアカウントに依存しないようにできます。そのため、ローカルマシン設定にアクセスする必要があります。

SQL Anywhere プログラムの他に、一部の Web サーバがサービスとして実行されます。そのような Web サーバで Apache や Microsoft IIS を使用するには、ローカルマシン設定をしてください。

一般的には、ローカルマシン設定をお奨めします。

### 1.5.2.3.2 .ini ファイルの内容を非表示にする

暗号化を使用して、.ini ファイル内のパスワードやその他の情報をファイル非表示ユーティリティ (dbfhide) によって保護します。

## コンテキスト

SQL Anywhere では .ini ファイルに特定の名前が付けられていると想定します。名前が重要なファイル (saldap.ini など) を暗号化する場合、元の暗号化されていないファイルのコピーを別の名前を付けて保存してください。元のファイルのコピーを保存していない場合、ファイルがいったん暗号化されると、その内容を変更できません。

### 警告

SQL Anywhere データソースだけを使用する場合を除き、UNIX ではファイル非表示ユーティリティ (dbfhide) を使用して、システム情報ファイル (デフォルトのファイル名は .odbc.ini) を暗号化しないでください。

他のデータソース (Mobile Link 同期など) を使用する予定の場合、システム情報ファイルの内容を難読化すると、他のドライバが正しく機能しなくなることがあります。

## 手順

1. ファイルを別の名前で作成してください。

```
rename saldap.ini saldap.ini.org
```

2. ファイル非表示ユーティリティを使用してファイルを暗号化し、暗号化されたファイルに必要なファイル名を付けます。

```
dbfhide saldap.ini.org saldap.ini
```

3. ファイルシステム保護またはオペレーティングシステム保護を使用して saldap.ini.org ファイルを保護するか、ファイルを安全な場所に保存します。

saldap.ini ファイルに変更を加えるには、saldap.ini.org ファイルを編集し、手順 2 を繰り返します。

## 結果

.ini ファイルの内容が非表示になります。

## 関連情報

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

### 1.5.2.3.3 レジストリ構造

Windows では、レジストリエディタを使用してレジストリに直接アクセスできます。

SQL Anywhere レジストリエントリは、以下のロケーションにある HKEY\_CURRENT\_USER または HKEY\_LOCAL\_MACHINE キーに保管されています。

```
Software\SAP\SQL Anywhere\17.0
```

#### 警告

レジストリは、ユーザ自身の責任で変更してください。システムをバックアップしてからレジストリを変更することをおすすめします。

## 1.5.2.3.4 インストール時のレジストリ設定

Windows では、インストールプログラムによって HKEY\_LOCAL\_MACHINE¥Software¥SAP レジストリキーに設定が作成されます。

次のリストは、レジストリ設定の一部を示します。

### SQL Anywhere¥17.0.4¥Location

このエントリには、SQL Anywhere ソフトウェアのインストールディレクトリが設定されます。次に例を示します。

```
Location "C:¥Program Files¥SQL Anywhere 17"
```

### SQL Anywhere¥17.0.4¥Samples Location

このエントリには、サンプルプログラムのインストールディレクトリが設定されます。次に例を示します。

```
Samples Location "C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples"
```

### SQL Anywhere¥17.0.4¥Online Resources

このエントリにはオンラインリソースのロケーションが設定されます。次に例を示します。

```
Online Resources "C:¥Program Files¥SQL Anywhere 17¥support¥OnlineResources.html"
```

### SQL Anywhere¥17.0.4¥Language

このエントリには、メッセージとエラーに使用される現在の言語を示す 2 文字のコードが設定されます。次に例を示します。

```
Language "EN"
```

言語は、インストール中に指定された言語選択に基づいて設定されます。

### 警告

特定の Intel ストレージドライバを使用するシステムの停電に対する堅牢性を高めるには、特定のレジストリエントリを設定します。このパラメータを設定しないと、停電発生時にデータが失われたり、データベースが破損したりする可能性があります。

## 関連情報

[ロケール言語 \[588 ページ\]](#)

## 1.5.3 タイムゾーンの管理

ホストコンピュータのタイムゾーンではなく、シミュレートされたタイムゾーンをデータベースで使用できます。

デフォルトでは、データベースサーバは、データベースサーバが実行されるコンピュータのタイムゾーンを使用します。ただし、データベースが異なるタイムゾーンにあるかのように実行されるように設定できます。シミュレートされたタイムゾーンでデータ

ベースを実行する場合、時間の計算や機能ではシミュレートされたタイムゾーンが使用されます。スケジュールされたイベント、トランザクションログ、監査レコードのみが、シミュレートされたタイムゾーンではなく、データベースサーバのタイムゾーンを引き続き使用します。

この機能により、同じデータベースサーバからさまざまなデータベースを実行し、各データベースを異なるタイムゾーンで実行しているように表示できます。たとえば、ある企業がロンドン、ローマ、上海の顧客に対して、ドイツにある同じデータベースサーバ上の 3 つのデータベースをホストし、各データベースのタイムゾーンで、顧客のいる都市のタイムゾーンを表示することができます。

また、この機能により、さまざまなロケーションにあるデータベースを実行し、それらすべてのデータベースを同じタイムゾーンで実行されているように表示できます。たとえば、アラスカ、ニューヨーク、ロサンゼルスにデータベースを配置し、それらすべてのデータベースをロサンゼルスで実行しているように表示できます。データベースを実行するタイムゾーンを決定するには、タイムゾーン、CurrentTimezoneOffset データベースプロパティ、または SYSTIMEZONE システムビューを問い合わせます。TimeZoneAdjustment データベースサーバプロパティは、データベースサーバのタイムゾーンを計算するのに使用できます。

シミュレートされたタイムゾーンの使用は、CURRENT TIME 特別値を使用する計算や、NOW または GETDATE 関数に影響しますが、CURRENT UTC TIME の値や CURRENT SERVER TIME 特別値には影響しません。時間、日付、タイムスタンプ (テーブルから返される値以外) を結果とするクエリは、データベースに time\_zone オプションが設定されている場合、データベースのタイムゾーンを使用して返されます。

#### i 注記

別のデータベースに対して時間、日付、タイムスタンプを返すクエリを実行する場合、クエリを実行する対象のデータベースのタイムゾーンではなく、現在接続しているデータベースのタイムゾーンを使用して値が返されます。

このセクションの内容:

#### [シミュレートされたタイムゾーンの作成 \(SQL Central\) \[569 ページ\]](#)

SQL Central を使用して、タイムゾーンを作成してデータベースを設定し、そのシミュレートされたタイムゾーンでデータベースを実行します。

#### [シミュレートされたタイムゾーンの作成 \(SQL\) \[570 ページ\]](#)

Interactive SQL を使用して、タイムゾーンを作成してデータベースを設定し、そのシミュレートされたタイムゾーンでデータベースを実行します。

## 1.5.3.1 シミュレートされたタイムゾーンの作成 (SQL Central)

SQL Central を使用して、タイムゾーンを作成してデータベースを設定し、そのシミュレートされたタイムゾーンでデータベースを実行します。

### 前提条件

MANAGE TIME ZONE システム権限が必要です。

## コンテキスト

デフォルトでは、データベースサーバは、データベースサーバが実行されるコンピュータのタイムゾーンを使用します。このタスクでは、データベースが異なるタイムゾーンにあるかのように実行されるように設定します。シミュレートされたタイムゾーンでデータベースを実行する場合、時間の計算や機能ではシミュレートされたタイムゾーンが使用されます。トランザクションログと監査レコードのみが、シミュレートされたタイムゾーンではなく、データベースサーバのタイムゾーンを引き続き使用します。

## 手順

1. SQL Central で、左ウィンドウ枠で**タイムゾーン**をクリックします。
2. 右ウィンドウ枠で右クリックし、**新規** > **タイムゾーン** をクリックします。

## 結果

新しいタイムゾーンが作成されます。デフォルトではデータベースの現在のタイムゾーンに設定されます。

### 1.5.3.2 シミュレートされたタイムゾーンの作成 (SQL)

Interactive SQL を使用して、タイムゾーンを作成してデータベースを設定し、そのシミュレートされたタイムゾーンでデータベースを実行します。

## 前提条件

MANAGE TIME ZONE システム権限が必要です。

## コンテキスト

デフォルトでは、データベースサーバは、データベースサーバが実行されるコンピュータのタイムゾーンを使用します。このタスクでは、データベースが異なるタイムゾーンにあるかのように実行されるように設定します。シミュレートされたタイムゾーンでデータベースを実行する場合、時間の計算や機能ではシミュレートされたタイムゾーンが使用されます。トランザクションログと監査レコードのみが、シミュレートされたタイムゾーンではなく、データベースサーバのタイムゾーンを引き続き使用します。

## 手順

1. 次の手順では、オーストラリア東部標準時に基づいてシミュレートされたタイムゾーンを作成し、データベースでこのタイムゾーンを使用するように設定します。Interactive SQL で、データベースに接続します。
2. 次の文を実行し、NewSouthWales というタイムゾーンを作成します。このタイムゾーンは、UTC より 10 時間進んでいます。夏時間は、10 月の第 1 日曜日の午前 2 時から 4 月の第 1 日曜日の午前 2 時までです。

```
CREATE TIME ZONE NewSouthWales OFFSET '10:00'  
STARTING 'Oct/Sun>=1' AT '2:00'  
ENDING 'Apr/Sun>=1' AT '2:00';
```

3. 次の文を実行し、データベースでシミュレートされたタイムゾーン NewSouthWales を使用するように設定します。

```
SET OPTION PUBLIC.time_zone='NewSouthWales';
```

## 結果

新しいタイムゾーンが作成されます。デフォルトではデータベースの現在のタイムゾーンに設定されます。

## 1.5.4 国際言語と文字セット

国際化とは、ソフトウェアの指定言語やソフトウェアを実行しているオペレーティングシステムに関係なく、ソフトウェアが各種言語とそれに適した文字セットに対処できるようにすることです。

次に、要求されたり使用されたりすることが多い機能について説明します。

### Unicode のサポート

Unicode は次のようにサポートされています。

- クライアントでは、ODBC、OLE DB、ADO.NET、JDBC の SQL Anywhere クライアントライブラリの UTF-16 をサポートしています。
- UTF-8 の Unicode 文字データの格納に NCHAR データ型を使用できます。
- CHAR データ型では UTF-8 エンコードを使用できます。

### コードページと文字セット

データベースサーバと関連ツールは、Windows (ANSI/ISO)、UTF-8、UNIX のコードページと文字セットをサポートしています。

### 照合

照合アルゴリズムとして、SQL Anywhere 照合アルゴリズム (SACA) と、International Components for Unicode (ICU) を使用した Unicode 照合アルゴリズム (UCA) の 2 つをサポートしています。

SACA を使用すると、ソートが高速、簡潔、実用的になりますが、言語的な正確さは若干低下します。UCA を使用すると、言語的な処理は正確になりますが、記憶領域の要件と実行時間が多少増加します。

高度なソートおよび比較機能として、SORTKEY および COMPARE 関数が用意されています。これらの関数は、辞書や電話帳並みに言語的に高度なソート機能を実現します。必要に応じて、大文字小文字とアクセント記号を区別しないソートや比較ができます。

文字カラムで SORTKEY ベースのソートを自動的に使用する設計機能も用意されています。sort\_collation データベースオプションを使用すると、文字カラムに ORDER BY が指定されたときに使用されるソート順序を指定できます。文字カラムのソートキーの格納に計算カラムも使用できるので、ORDER BY が指定されるたびに計算カラムを計算する必要はありません。

#### 文字セット変換

データは、サーバシステムとクライアントシステムの文字セットエンコード間で変換されます。複数の異なる文字セットを使用している環境でもデータの整合性が維持されます。

#### 識別子

ほとんどのシングルバイト文字やマルチバイト文字を含む識別子を引用符で囲まなくても使用できます。例外は、スペースや句読表記号です。

#### 通貨

通貨記号は、ユーロ記号も含めて、ソートの対象になります。通貨の書式サポートはありません。

#### 日付と時刻の形式

グレゴリオ暦をサポートしており、日付と時刻の設定用にさまざまな形式を用意しています。カスタム形式は、date\_format、time\_format、timestamp\_format の各データベースオプションを使用して実現できます。date\_format と timestamp\_format オプションのデフォルトは、ISO 互換の日付形式 YYYY-MM-DD です。CONVERT 関数が用意されており、日付と時刻の出力形式を一般的な各種形式に変換できます。

このセクションの内容:

#### [ローカライズ版ソフトウェア \[573 ページ\]](#)

ローカライズとは、製品を目的のロケールの言語や文化に適用させることです。ロケールとは、通常は言語と国/地域の組み合わせです。

#### [文字セット \[576 ページ\]](#)

文字セットは記号、文字、数字、スペースなどから成ります。

#### [文字セット変換 \[581 ページ\]](#)

文字セットまたはコードページの異なる場所で同じ文字を表している文字セット間で、文字セット変換を実行できます。

#### [ロケール \[587 ページ\]](#)

データベースサーバとクライアントライブラリは、ロケール定義を使用して、言語と文字セットの環境を認識します。

#### [照合 \[591 ページ\]](#)

照合とは、特定の文字セットまたはエンコードに対する文字のソートおよび比較の方法です。

#### [推奨文字セットと照合 \[607 ページ\]](#)

数多くの文字セット、コードページ、エンコード、照合の名前が認識されますが、使用言語に応じて、Windows プラットフォームと UNIX プラットフォームでの使用が推奨されるものがいくつかあります。

#### [トルコ語文字セットと照合 \[610 ページ\]](#)

トルコ語には、1 に相当する文字に 2 通りの表記形式があります。

## 関連情報

[ICU とは何か、いつ必要になるか \[575 ページ\]](#)

[照合の考慮事項 \[600 ページ\]](#)

[sort\\_collation オプション \[783 ページ\]](#)

[date\\_format オプション \[684 ページ\]](#)

[time\\_format オプション \[808 ページ\]](#)

[timestamp\\_format オプション \[811 ページ\]](#)

### 1.5.4.1 ローカライズ版ソフトウェア

ローカライズとは、製品を目的のロケールの言語や文化に適用させることです。ロケールとは、通常は言語と国/地域の組み合わせです。

ローカライズは、包装、インストーラ、マニュアル、ソフトウェアのユーザインタフェース、エラー/警告/情報メッセージなど、多くのコンポーネントに反映されます。

SQL Anywhere ソフトウェアは、次の 5 言語にローカライズされています。

- 英語
- フランス語
- ドイツ語
- 日本語
- 中国語 (簡体文字)

使用言語はインストール時に選択します。

マニュアルは、英語版、ドイツ語版、日本語版、簡体字中国語版が用意されています。

Windows の場合、[スタートメニュー](#)を使用して、インストールされた言語と英語を切り替えることができます。言語選択ユーティリティ (dblank) を使用すると、追加で導入した言語を含む使用可能なすべての言語に設定し直すことができます。

次の表では、各言語のサポート状況をオペレーティングシステムプラットフォーム別に示します。

すべての Windows および Linux プラットフォームでは、英語版、フランス語版、ドイツ語版、日本語版、簡体字中国語版のソフトウェアが用意されています。

UNIX および Mac OS X では、英語版のソフトウェアのみが利用可能です。

このセクションの内容:

[ソフトウェアとマニュアルの完全ローカライズ \[574 ページ\]](#)

ソフトウェアとマニュアルが複数の言語で利用可能で、開発、配備、管理に適しています。

[Windows での配備ソフトウェアのローカライズ \[574 ページ\]](#)

主要 5 言語 (英語、フランス語、ドイツ語、日本語、中国語 (簡体文字)) の他に、次の言語の配備ソフトウェアリソースが用意されています。

[ICU とは何か、いつ必要になるか \[575 ページ\]](#)

ICU (International Components for Unicode) は、IBM が開発および保守しているオープンソースライブラリです。ICU は、Unicode サポートを提供することによって、ソフトウェアの国際化を容易にします。

## 関連情報

[言語選択ユーティリティ \(dblang\) \[1131 ページ\]](#)

### 1.5.4.1.1 ソフトウェアとマニュアルの完全ローカライズ

ソフトウェアとマニュアルが複数の言語で利用可能で、開発、配備、管理に適しています。

- 英語
- フランス語
- ドイツ語
- 日本語
- 中国語 (簡体文字)

英語版、ドイツ語版、日本語版、簡体字中国語版では、次に示すすべてのコンポーネントがローカライズされています。

- インストーラ
- マニュアルとコンテキスト別のヘルプ
- ソフトウェア

フランス語版では、インストーラ、ソフトウェア、コンテキスト別のヘルプがローカライズされています。

### 1.5.4.1.2 Windows での配備ソフトウェアのローカライズ

主要 5 言語 (英語、フランス語、ドイツ語、日本語、中国語 (簡体文字)) の他に、次の言語の配備ソフトウェアリソースが用意されています。

- イタリア語
- 韓国語
- リトアニア語
- ポーランド語
- ブラジルポルトガル語
- ロシア語
- スペイン語
- 中国語 (繁体文字)
- ウクライナ語

配備ローカライズは、エンドユーザに配備されることが多い一部のソフトウェアコンポーネントが対象となります。包装、マニュアル、管理ソフトウェア、開発ソフトウェア、およびインストールソフトウェアはローカライズされません。ローカライズされるソフトウェアコンポーネントは次のとおりです。

- データベースサーバとクライアントライブラリ
- Mobile Link サーバおよびクライアント
- SQL Remote クライアント
- コマンドラインツール (dbinit、dbunload など)

### 1.5.4.1.3 ICU とは何か、いつ必要になるか

ICU (International Components for Unicode) は、IBM が開発および保守しているオープンソースライブラリです。ICU は、Unicode サポートを提供することによって、ソフトウェアの国際化を容易にします。

ICU は、特定の文字セット変換と照合オペレーションの実装で使用されています。

#### データベースサーバで ICU が必要になるとき

データベースサーバで常に ICU を利用できることが理想的です。次の表に、ICU が必要になるときとその理由を示します。

ICU が必要になるとき	注記
NCHAR 文字セットまたは CHAR 文字セットの照合として Unicode Collation Algorithm (UCA) が使用されています。	UCA は ICU を必要とします。
データベースの文字セットは UTF-8 でなく、マルチバイト文字セットです。	データベースの文字セットから UTF-8 にパスワード変換するために必要です (データベースパスワードは内部的に UTF-8 で格納されています)。
クライアントとデータベースの文字セットが異なり、いずれかがマルチバイト (UTF-8 を含む) です。これには、Unicode ODBC、OLE DB、ADO.NET、SQL Anywhere JDBC アプリケーションが含まれます。ICU のインストールされていないクライアントが使用するデータベース文字セットには関係ありません。	マルチバイト文字セットの適切な変換を行うためには ICU が必要です。
データベースの文字セットが UTF-8 でなく、CHAR と NCHAR 間の変換が必要です。	データベースサーバは、UTF-8 を別の文字セットに変換するために ICU を必要とします。
Embedded SQL クライアントが UTF-8 以外の NCHAR 文字セットを使用します。	データベースサーバは、UTF-8 を別の文字セットに変換するために ICU を必要とします。Embedded SQL クライアントのデフォルトの NCHAR 文字セットは、最初のクライアントの CHAR 文字セットと同じです。これは、db_change_nchar_charset 関数を使用して変更できます。
CSCONVERT または SORTKEY 関数が使用されています。CSCONVERT 関数は、上記の 3 番目の項目要件に準ずる文字セットを変換するために呼び出されます。	マルチバイト文字セットの文字セット変換を行うためには ICU が必要です。SORTKEY のラベル用に SORTKEY を生成するには UCA が必要であるため、結果として ICU が必要になります。

#### データベースサーバ上で ICU を使用しないで正しい文字セット変換を実行できるとき

データベースの文字セットとクライアントの文字セットの両方がシングルバイトであり、sqlany.cvf を利用できるとき (すべてのプラットフォーム)、またはオペレーティングシステムが変換をサポートするとき (Windows のみ) は、ICU を使用しないで

正しい文字セット変換を行うことができます。これは、`sqlany.cvf` を利用できる場合、またはホストのオペレーティングシステムに適切な変換機能がインストールされている場合にかぎり、ICU を使用しなくてもシングルバイト間の変換が処理可能なためです。

## クライアントで ICU が必要になるとき

Unicode のクライアントアプリケーションでは、使用するデータベースの文字セットにかかわらず、すべてのクライアントに ICU がインストールされている場合に、クライアントとデータベースサーバ間のパフォーマンスが向上する可能性があります。これは、必要な変換処理の一部がデータベースサーバからクライアントに渡されて、負荷が軽減するからです。

また、Windows プラットフォームで ODBC を使用している場合は、ANSI アプリケーションであっても、クライアントに ICU をインストールしておく必要があります。これは、ドライバマネージャが ANSI ODBC 呼び出しを Unicode ODBC 呼び出しに変換するからです。

### 1.5.4.2 文字セット

文字セットは記号、文字、数字、スペースなどから成ります。

各ソフトウェアは、文字セットを使用します。"ISO-8859-1" は文字セットの例です。Latin1 と呼ばれます。

文字を内部的に適切に表すため、各ソフトウェアはエンコード (文字コードとも呼ばれる) を使用します。エンコードとは、各文字を 1 バイトまたは複数バイトの情報にマッピングする方法で、16 進数で表します。UTF-8 はエンコードの例です。

"文字セット" と "エンコード" は密接に関連しており、どちらも "エンコード" の意味で使用されることがあります。

コードページは、エンコードの一形態です。コードページとは文字と数値表現とのマッピングのことで、通常、数値表現は 0 ~ 255 の整数です。コードページの例には、Windows コードページ 1252 があります。

このマニュアルでは、"エンコード"、"文字コード"、"文字セットエンコード"、"コードページ" を同じ意味で使用します。

データベースサーバは、文字をソート (たとえば、名前をアルファベット順にリスト) するときに照合を使用します。照合は文字コード (文字と表現間のマッピング) と文字のソート順の組み合わせです。各文字列にソート順が複数ある場合があります。たとえば、大文字/小文字を区別するソート順と大文字/小文字を区別しないソート順があります。また、言語間で同じ文字に対するソート順が異なる場合もあります。

文字はフォントを使って画面上に表示されます。これは文字セットの文字とその外観との間のマッピングです。フォントはオペレーティングシステムによって処理されます。

オペレーティングシステムは、キーボードマッピングを使って、キーボードのキーまたはキーの組み合わせを文字セットの文字にマッピングします。

このセクションの内容:

#### [クライアント/サーバコンピューティングにおける言語の問題 \[577 ページ\]](#)

クライアントアプリケーションで作業するデータベースユーザは、複数のソースから文字列を参照したり、文字列にアクセスしたりする場合があります。

#### [シングルバイト文字セット \[578 ページ\]](#)

多くの言語では文字の数はシングルバイトの文字セットで扱える程度です。

#### [マルチバイト文字セット \[578 ページ\]](#)

言語によっては、256 文字よりもはるかに多い文字があります。こうした言語はシングルバイトで表すことができないため、マルチバイトエンコードを使用します。

#### [Windows 環境の ANSI コードページと OEM コードページ \[579 ページ\]](#)

Windows の場合、ANSI コードページと OEM コードページという 2 つのコードページが使用されています。

#### [データベースの文字セット \[579 ページ\]](#)

データベースでは、文字データの格納に 1 つまたは 2 つの文字セット (エンコード) を使用できます。

#### [サポートされている文字セット \[580 ページ\]](#)

数多くの文字セットとラベルがサポートされており、その対象は増え続けています。

#### [文字セットに関する質問とその回答 \[580 ページ\]](#)

文字セットに関する質問への答えを見つけるために使用できる複数分野のドキュメントがあります。

## 1.5.4.2.1 クライアント/サーバコンピューティングにおける言語の問題

クライアントアプリケーションで作業するデータベースユーザは、複数のソースから文字列を参照したり、文字列にアクセスしたりする場合があります。

#### データベース内のデータ

データベースには文字列やその他のテキストデータが格納されています。データベースサーバは要求に応答するときに、これらの文字列を処理します。たとえば、データベースサーバが、テーブルの N より後の文字で始まるすべての名前を表示するよう求められることがあります。この要求では、文字列比較を実行する必要がありますが、特定の文字セットのソート順序が想定されています。

#### データベースサーバソフトウェアメッセージ

アプリケーションによってデータベースエラーが引き起こされることがあります。たとえば、存在しないカラムを参照するクエリをアプリケーションが送信した場合です。この場合、データベースサーバは警告かエラーメッセージを返します。このメッセージは言語リソースライブラリに保持されます。これはデータベースサーバが使用する DLL または共有ライブラリです。

#### クライアントアプリケーション

クライアントアプリケーションのインターフェイスはテキストを表示します。また、内部でテキストを処理できます。

#### クライアントソフトウェアメッセージ

クライアントライブラリは、データベースサーバと同じ言語を使用してクライアントアプリケーションにメッセージを提供します。

#### オペレーティングシステム

クライアントとサーバのオペレーティングシステムは、メッセージを提供したりテキストを処理したりします。

環境を適切に動作させるためには、テキストの入力箇所のすべてで統合的に機能しなければなりません。大まかに言うと、すべてユーザの言語および文字セット、またはそのいずれかで動作させてください。

## 1.5.4.2.2 シングルバイト文字セット

多くの言語では文字の数はシングルバイトの文字セットで扱える程度です。

このような文字セットでは、各文字はシングルバイト (2 桁の 16 進数) で表現されます。

シングルバイトのセットでは最大 256 文字を表すことができます。アクセント記号の付いた文字を含め、国際的に使用するすべての文字を保持できるシングルバイト文字セットはありません。この問題は、1 つ以上の国の言語に適する文字セットを記述するコードページのセットの開発により解決されました。たとえば、コードページ 1253 にはギリシャ語の文字セットが含まれており、コードページ 1252 には西ヨーロッパ言語の文字セットが含まれています。コードページは多数あり、その名前も多数あります。上記の例は、Windows のコードページです。

### 上方ページと下方ページ

わずかな例外はありますが、文字 0 ~ 127 はすべてのコードページで共通です。この範囲の文字のマッピングを ASCII 文字セットと呼びます。これには、英語のアルファベットの大文字と小文字、共通の句読表記記号、数字が含まれます。この範囲は 7 ビット範囲 (127 までの文字を表すのに 7 ビットしか必要ないため) または下方ページと呼ばれます。128 ~ 255 までの文字は拡張文字、または上方コードページ文字と呼ばれ、コードページ間で異なります。

英語のアルファベット文字だけを使用する場合は、各コードページの ASCII 部分 (0 ~ 127) のみで表せるため、コードページの互換性の問題が起こることはほとんどありません。しかし、その他の文字を使用すると、非英語環境ではよく起こることで、データベースとアプリケーションが異なるコードページを使用している場合に、問題が起こる可能性があります。

たとえば、UTF-8 文字セットを使用するデータベースが cp1252 データを含むファイルからテーブルをロードするときに、LOAD TABLE 文でエンコードが cp1252 に指定されていないとします。エンコードが指定されていないので、データのエンコードは UTF-8 であると想定され、文字変換は実行されないため、cp1252 エンコードがデータベースに直接格納されます。cp1252 では 16 進数の 80 で表されているユーロ記号などの文字は UTF-8 に変換されません。UTF-8 のユーロ記号は E2 82 AC の 3 バイトシーケンスで表されますが、この例の場合はデータベースに 80 として格納されます。後でアプリケーションからデータを要求されたとき、データベースサーバはデータを UTF-8 からクライアントの文字セットに変換しようとします。この変換により、文字の破損が発生します。

## 1.5.4.2.3 マルチバイト文字セット

言語によっては、256 文字よりもはるかに多い文字があります。こうした言語はシングルバイトで表すことができないため、マルチバイトエンコードを使用します。

さらに、多くの言語の文字を単一の文字セットで表現するために、マルチバイトの文字セットよりも多くの文字を使う文字セットも存在します。この例として、UTF-8 が挙げられます。

マルチバイト文字セットは可変幅で、いくつかの文字はシングルバイト、他はダブルバイトなどになります。

### 例

たとえば、コードページ 932 (日本語) の文字の長さは 1 バイトまたは 2 バイトです。最初のバイト (リードバイトとも呼ばれる) の値が 16 進数値 ¥x81 ~ ¥x9F または ¥xE0 ~ ¥xFC (10 進数値 129 ~ 159 または 224 ~ 252) の範囲にある場合、その文字は 2 バイト文字であり、直後のバイト (フォローバイトとも呼ばれる) と併せて文字が成立します。フォローバイトとは、最初のバイト以外のすべてのバイトのことです。

最初のバイトがリードバイトの範囲外にある場合、その文字はシングルバイト文字であり、次のバイトは次の文字の最初のバイトになります。

## 関連情報

[SQL Anywhere 照合アルゴリズム \(SACA\) \[594 ページ\]](#)

### 1.5.4.2.4 Windows 環境の ANSI コードページと OEM コードページ

Windows の場合、ANSI コードページと OEM コードページという 2 つのコードページが使用されています。

Windows グラフィカルユーザインタフェースを使用するアプリケーションでは、Windows コードページが使用されます。Windows コードページには、ISO 文字セットおよび ANSI 文字セットとの互換性があります。このようなコードページは、しばしば **ANSI** コードページと呼ばれます。

Windows で動作する文字モードアプリケーション (コマンドプロンプトウィンドウを使用するアプリケーション) は DOS で使用されていたコードページを使用します。これは歴史的な理由から **OEM** コードページ (Original Equipment Manufacturer) と呼ばれます。

OEM コードページと ANSI コードページの両方に基づく照合がサポートされています。OEM 照合は互換性確保のためにサポートされていますが、新しいデータベースでは使用しないでください。

## 関連情報

[代替照合 \[597 ページ\]](#)

### 1.5.4.2.5 データベースの文字セット

データベースでは、文字データの格納に 1 つまたは 2 つの文字セット (エンコード) を使用できます。

CHAR データ型 (CHAR、VARCHAR、LONG VARCHAR など) では、シングルバイト文字セットまたはマルチバイト文字セットが使用されます。UTF-8 も使用できます。NCHAR データ型 (NCHAR、NVARCHAR、LONG NVARCHAR など) では、UTF-8 が使用されます。

LOAD TABLE などの SQL 文や CSCONVERT、TO\_CHAR、TO\_NCHAR などの関数は、パラメータとして **db\_charset** と **nchar\_charset** を使用して、それぞれデータベースの CHAR 文字セットと NCHAR 文字セットを参照します。

## 1.5.4.2.6 サポートされている文字セット

数多くの文字セットとラベルがサポートされており、その対象は増え続けています。

文字セットエンコードラベルは、さまざまな名前やラベルが知られています。サポートされている文字セットのリストを表示するには、次のコマンドを実行します。

```
dbinit -le
```

出力の各行には、指定された文字セットエンコードで最も一般的なラベルがカンマで区切られた形式で表示されます。各行の最初のラベルは、その文字セットエンコードで優先される名称です。その他のラベルは、別の機関、組織、または標準によって使用されるもので、IANA (Internet Assigned Numbers Authority)、MIME (Multipurpose Internet Mail Extensions)、ICU (International Components for Unicode)、Java、および ASE (Adaptive Server Enterprise) があります。

目的の文字セットが見つからない場合は、次のコマンドを実行して、あまり一般的でない文字セットを含むリストを参照できます。

```
dbinit -le+
```

文字セットエンコードのラベルを指定すると、データベースサーバによってラベルの検索が行われます。場合によっては、他の機関が異なる文字セットに対して同じラベルを使用することがあります。データベースサーバは、可能なかぎり、あいまいさを解決しようとします。たとえば、あいまいなラベルで設定されている文字を参照する JDBC アプリケーションは、Java 標準のラベルに解決されます。あいまいさを回避するため、常に SQL Anywhere ラベルを使用することをお奨めします。

dbinit -le オプションによって返される文字セットエンコードのラベルのほかに、次の文字セットエイリアスも使用できます。

### os\_charset

データベースサーバをホストしているオペレーティングシステムが使用する文字セットのエイリアス。

### char\_charset

データベースが使用する CHAR 文字セットのエイリアス。

### nchar\_charset

データベースが使用する NCHAR 文字セットのエイリアス。

特定の文字セットまたはラベルがサポートされているかどうかを簡単に確認するには、CSCONVERT 関数を使用します。

## 1.5.4.2.7 文字セットに関する質問とその回答

文字セットに関する質問への答えを見つけるために使用できる複数分野のドキュメントがあります。

質問	検索および参照すべき対象
データベースで使用する照合を決定する方法は?	照合
ソフトウェア、特に SQL Anywhere における文字の表示方法は?	文字セット
SQL Anywhere が提供する照合の種類は?	照合の考慮事項
SQL Anywhere がサポートする文字セットエンコードは?	サポートされている文字セット

質問	検索および参照すべき対象
クライアントコンピュータとデータベースとで使用文字セットが違います。クライアントとサーバ間で文字を正しくやり取りする方法は？	文字セット変換
接続文字列で使用できる文字セットは？	接続文字列と文字セット
既存のデータベースの照合順を変更する方法は？	データベースの照合の変更

## 関連情報

[照合 \[591 ページ\]](#)

[照合の考慮事項 \[600 ページ\]](#)

[サポートされている文字セット \[580 ページ\]](#)

[文字セット変換 \[581 ページ\]](#)

[接続文字列と文字セット \[582 ページ\]](#)

[データベースの照合の変更 \[605 ページ\]](#)

### 1.5.4.3 文字セット変換

文字セットまたはコードページの異なる場所で同じ文字を表している文字セット間で、文字セット変換を実行できます。

これを可能にするには、文字セット間にある程度の互換性が必要になります。たとえば、EUC-JIS と cp932 との間では文字セット変換を実行できますが、EUC-JIS と cp1252 との間では実行できません。

文字セット変換は、IBM が開発および保守している International Components for Unicode (ICU) オープンソースライブラリを使用して実装されています。

このセクションの内容:

[接続文字列と文字セット \[582 ページ\]](#)

使用している文字セットがクライアント間で異なる場合、接続文字列の文字セット変換はたいへん困難になります。

[SQL 文と文字セット \[582 ページ\]](#)

文字セット変換によって、すべての SQL 文は解析と実行の前にデータベース文字セットに変換されます。

[データを表示した場合の予期しない記号のトラブルシューティング \[583 ページ\]](#)

Interactive SQL などのクライアントアプリケーションを使用してデータを表示すると、四角形、矢印、疑問符などの予期しない記号が文字としてデータに表示される場合があります。

[外国語での大文字と小文字の区別 \[583 ページ\]](#)

識別子では常に大文字と小文字を維持し、大文字と小文字を区別しません。

[文字セット変換とクライアント API \[584 ページ\]](#)

マルチ文字セット環境で作業する場合、文字セット変換の問題が発生すると、その問題がどこで発生したかの特定が困難である可能性があります。

### 1.5.4.3.1 接続文字列と文字セット

使用している文字セットがクライアント間で異なる場合、接続文字列の文字セット変換はたいへん困難になります。

接続文字列は、データベースサーバを検出または起動するために、クライアントライブラリによって解析されますが、解析はデータベースサーバの文字セットや言語が未知のまま実行されます。

インタフェースライブラリは、接続文字列を次のように解析します。

1. 接続文字列は、`keyword=value` の組み合わせに分解されます。これは、CommLinks (LINKS) 接続パラメータの前後に中カッコ {} を使用しないかぎり、文字セットにかかわらず実行されます。中カッコの代わりにカッコ () を使用することをお奨めします。中カッコは、一部のマルチバイト文字セットで有効なフォロワーバイト (最初のバイト以外のバイト) です。
2. データベースサーバが検出されます。サーバ名に対して文字セット変換は実行されません。クライアントとデータベースサーバ間で文字セットが異なる場合、サーバ名に拡張文字が使用されているとサーバが見つからないことがあります。クライアントとサーバを異なるオペレーティングシステムやロケールで実行している場合は、サーバ名に 7 ビットの ASCII 文字を使用してください。
3. DatabaseName (DBN) 接続パラメータまたは DatabaseFile (DBF) 接続パラメータは、クライアントの文字セットからデータベースサーバの文字セットに変換されます。
4. データベースが検出されると、残りの接続パラメータがデータベースの文字セットに変換されます。

### 1.5.4.3.2 SQL 文と文字セット

文字セット変換によって、すべての SQL 文は解析と実行の前にデータベース文字セットに変換されます。

この変換による副作用として、SQL 文に含まれるデータベース文字セットに変換できない文字は置換文字に変換されます。任意の Unicode 文字が含まれた SQL 文は、次のいずれかの方法で実行できます。

- UNISTR 関数を使用して Unicode 文字値を指定する
- ホスト変数を使用して Unicode 文字値を指定する
- データベースの文字セットとして UTF-8 を使用する

CHAR 照合として UTF8BIN を選択する場合、データベース文字セットは UTF-8 になります。CHAR 照合として UCA を選択する場合、エンコードとして UTF-8 を選択できます。

Unicode 照合アルゴリズム (UCA) では、高度な比較、ソート、大文字と小文字の変換ができますが、パフォーマンスが低下する場合があります。UTF8BIN は必要とする領域が少なく、高速ですが、ソート順と比較はバイナリで実行されます。SQL 文で Unicode 文字を使用する場合は、CHAR 照合に UTF8BIN を指定しますが、ソートと比較では UCA の一部の機能しか必要ありません。必要な場合のみ、SORTKEY と COMPARE 関数を使用して UCA を使用します。

## 関連情報

[Unicode 照合アルゴリズム \(UCA\) \[595 ページ\]](#)

[SQL Anywhere 照合アルゴリズム \(SACA\) \[594 ページ\]](#)

[接続パラメータと接続文字列 \[34 ページ\]](#)

### 1.5.4.3.3 データを表示した場合の予期しない記号のトラブルシューティング

Interactive SQL などのクライアントアプリケーションを使用してデータを表示すると、四角形、矢印、疑問符などの予期しない記号が文字としてデータに表示される場合があります。

予期しない記号が表示される最も一般的な理由は、データの表示に使用されたフォントが該当の文字をサポートしていないことです。このような問題は、Unicode のフォントに変更することで解決できます。クライアントアプリケーションのフォントを変更できない場合は、代わりにオペレーティングシステムのデフォルトのフォントを変更できます。

また、データベースに格納されている基本となるデータに問題がある場合にも、予期しない記号がクライアントアプリケーションで表示される可能性があります。たとえば、データをデータベースに追加する場合に文字セット変換が必要で、元の文字セットの一部の文字と対応する文字がデータベースの文字セットになかった場合、その文字の代わりに置換文字が挿入されます。

たとえば、Windows では、標準の英語フォント Tahoma は日本語の文字をサポートしていません。データベースの文字セットが cp932 で、データベースに日本語のデータが含まれている場合、データベースにクエリを実行すると、結果セットの日本語の文字は小さい四角記号で表示されます。Interactive SQL では、**ツール** > **オプション** > **SQL Anywhere** > **Ultra Light** > **結果** > **フォント** をクリックして、Arial Unicode MS や Lucida Sans Unicode などのユニコードフォントを指定すると、結果ウィンドウ枠のフォントを変更できます。ユニコードフォントは多言語の文字を表示できるため、このような場合の使用をお奨めします。

クライアントアプリケーションがフォント設定の変更を許可しない場合は、オペレーティングシステムのデフォルトのフォントを使用している可能性があります。このような場合、デフォルトのシステムフォントをユニコードフォントに変更する方法については、オペレーティングシステムのマニュアルを参考にしてください。

## 関連情報

[Interactive SQL \[980 ページ\]](#)

### 1.5.4.3.4 外国語での大文字と小文字の区別

識別子では常に大文字と小文字を維持し、大文字と小文字を区別しません。

名前は作成時の大文字と小文字のまま格納されますが、識別子へのアクセスは大文字と小文字の区別なしで行われます。

たとえば、システムビューの名前は大文字 (SYSDOMAIN、SYSTAB など) で格納されますが、アクセスは大文字と小文字の区別なく行うことができます。したがって、次の 2 つの文は同等です。

```
SELECT * FROM systab;  
SELECT * FROM SYSTAB;
```

照合では、大文字と小文字が同等と定義されています。ただし、一部の照合では、識別子の大小文字の区別を前提とする場合、特別な注意が必要です。たとえば、トルコ語の照合では、予期できない複雑なエラーが発生するような大文字と小文字の変換動作があります。最も一般的なエラーは、*ı* または *İ* という文字を含むシステムオブジェクトが見つからないというものです。

## 関連情報

[トルコ語文字セットと照合 \[610 ページ\]](#)

### 1.5.4.3.5 文字セット変換とクライアント API

マルチ文字セット環境で作業する場合、文字セット変換の問題が発生すると、その問題がどこで発生したかの特定が困難である可能性があります。

クライアント API で文字セット変換の問題が発生した場合は、文字セット変換を制御するデータベースと接続のオプションやプロパティを確認してください。

変換の問題が発生する可能性がある箇所は、2つのカテゴリに分けられます。1つ目のカテゴリは、誤ったフォーマットのデータをクライアント API に送信したことに関係します。このカテゴリの問題は、Unicode API では発生しませんが、他のすべてのクライアント API で発生する可能性があり、データは無効になります。

問題の2つ目のカテゴリは、最終的な文字セットまたはいずれかの中間文字セットに対応するものがない文字に関係します。この場合、置換文字が使用されます。これは損失を伴う変換と呼ばれ、どのクライアント API でも発生する可能性があります。データベース文字セットに UTF-8 を使用するようにデータベースを設定すると、損失を伴う変換を回避できます。

### 文字セット変換に影響するオプションとプロパティの設定

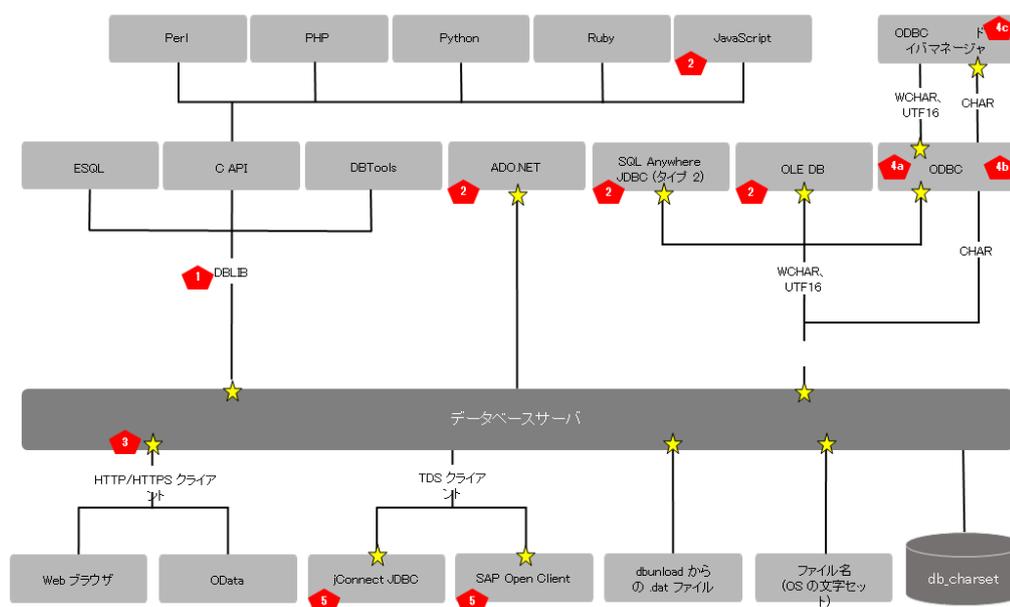
接続に影響するデータベースオプションとデータベースや接続のプロパティは、通常、完全な接続文字列で使用できます。ただし、PROPERTY、DB\_PROPERTY、CONNECTION\_PROPERTY などのシステム関数を使用して設定を問い合わせることもできます。例:

クエリ	説明
SELECT PROPERTY( 'CharSet' );	データベースサーバのオペレーティングシステムの文字セットを返します。
SELECT DB_PROPERTY( 'CharSet' );	データベースの CHAR 文字セットを返します。
SELECT DB_PROPERTY( 'NcharCharSet' );	データベースの NCHAR 文字セットを返します。
SELECT DB_PROPERTY( 'MultibyteCharSet' );	CHAR データでマルチバイト文字セットを使用するかどうか (On=使用、Off=使用しない) を返します。
SELECT CONNECTION_PROPERTY( 'CharSet' );	クライアントの CHAR 文字セットを返します。
SELECT CONNECTION_PROPERTY( 'NcharCharSet' );	クライアントの NCHAR 文字セットを返します。
SELECT CONNECTION_PROPERTY( 'on_charset_conversion_failure' );	on_charset_conversion_warning オプションの値を返します。

## クライアント API と文字セット変換のポイント

次の図は、クライアント API がデータベースサーバと対話するときに文字セット変換が行われる可能性がある場所を示しています。

図の黄色い星印は、文字セット変換が行われる可能性がある場所を示しています。図の数字は、図の後にある詳細な説明の番号に対応しています。



### 1 - DBLIB

デフォルトで DBLIB CHAR 文字セットと NCHAR 文字セットがクライアントのオペレーティングシステムの文字セットになります。CHAR 文字セットは、BINARY と NCHAR を除くすべての文字列データに使用されます。NCHAR 文字セットは、NCHAR 型のホスト変数とホストパラメータ（これらは DBTools または C API インタフェースでは使用できません）に使用されます。データベースサーバでは、すべての文字セット変換が実行されます。

CHAR 文字セットと NCHAR 文字セットは、それぞれ CharSet 接続パラメータと db\_change\_nchar\_charset 関数を使用して設定できます。

### 2 - Unicode API

Unicode API を使用すると、ドライバは文字データに Unicode UTF-16 エンコードを使用し、Unicode データとデータベース文字セット間の変換を行います。ドライバは Unicode ホストパラメータを NCHAR 文字セット (UTF-8) に変換してからデータベースサーバに送信します。NCHAR として記述された結果セットのカラムは、NCHAR 文字セット (UTF-8) でフェッチされ、受信後に Unicode に変換されます。CharSet 接続パラメータを設定すると損失を伴う変換が発生する可能性があるため、このパラメータの設定はおすすめしません。

### 3 - HTTP/HTTPS クライアント

**HTTP サービス (サーバとしての SQL Anywhere):** HTTP サービスの場合、2つのタイプの要求があります。URL エンコード要求と multipart/form-data 要求です。

## URL エンコード要求

要求は application/x-www-form-urlencoded の形式であり、変数はキーと値のペアで渡されます。データベースサーバは % でエンコードされたデータ (UTF-8 またはデータベース文字セット) をデコードし、キーと値のペアをデータベース文字セットに変換します。処理された値は HTTP\_VARIABLE 関数を使用して抽出できます。この関数で @BINARY または @TRANSPORT 属性を使用すると、% でデコードされ、文字セットが変換されていない値または未加工の HTTP (transport) 値をそれぞれ返すことができます。

## Multipart/form-data 要求

要求はバイナリと見なされます (@BINARY または @TRANSPORT 属性を使用すると、同じ値が返されます)。

HTTP サービス要求の文字セット変換は、sa\_set\_http\_option システムプロシージャを使用して設定できる HTTP オプション CharSetConversion と AcceptCharset によって制御されます。

要求や応答の文字セット変換に影響する可能性があるその他の設定には、優先文字セットの指定に使用される Accept-Charset ヘッダとコンテンツの文字セットの指定に使用される Content-Type ヘッダがあります。

**HTTP ストアドプロシージャ (クライアントとしての SQL Anywhere):** HTTP ストアドプロシージャの場合、CHAR データはデータベース文字セット内で送信されます。パラメータが NCHAR 型の場合、すべてのデータは UTF-8 として送信されます (すべての CHAR パラメータは UTF-8 に変換されます)。要求によって、優先文字セットとしてデータベース文字セットを指定する Accept-Charset HTTP ヘッダが送信されます。優先文字セットのリストには、常に UTF-8 が含まれます。応答の Content-Type ヘッダにデータベース文字セットではない文字セットが指定されている場合、応答はクライアントによってデータベース文字セットに変換されます。

## 4a - Unicode エントリポイントを使用する ODBC

ODBC アプリケーションは Unicode エントリポイントを使用する場合、Unicode クライアント API と見なされます。WCHAR データは Unicode API の場合と同じ方法で処理されます。ODBC アプリケーションが Unicode エントリポイントと CHAR (ANSI) データを使用する場合、データはデータベース文字セットであると思われ見なされます。1つのアプリケーションで CHAR データと WCHAR データを混在させることはおすすめしません。

## 4b - ANSI エントリポイントを使用する ODBC

ODBC アプリケーションは ANSI エントリポイントを使用する場合、ANSI クライアント API と見なされます。デフォルトで CHAR 文字セットがクライアントオペレーティングシステムの文字セットになります。CHAR 文字セットは CharSet 接続パラメータを使用して変更できます。

ANSI データの変換は、データベースサーバによって実行されます。WCHAR ホスト変数にフェッチすると、データベースサーバで CHAR データの損失を伴う変換が発生する可能性があります。1つのアプリケーションで CHAR データと WCHAR データを混在させることはおすすめしません。

## 4c - ODBC ドライバマネージャ

一部の ODBC ドライバマネージャは、すべての CHAR データを WCHAR データに変換してから Unicode エントリポイントを呼び出します。

## 5 - TDS クライアント

SAP Open Client や jConnect などの TDS クライアントは、接続時にデータベースサーバとネゴシエートし、クライアントで文字セット変換を実行することを選択します。ネゴシエーション中、データベースサーバは文字セット変換を実行しないよう指示されます。

## 関連情報

[CharSet \(CS\) 接続パラメータ \[53 ページ\]](#)

[on\\_charset\\_conversion\\_failure オプション \[746 ページ\]](#)

### 1.5.4.4 ロケール

データベースサーバとクライアントライブラリは、ロケール定義を使用して、言語と文字セットの環境を認識します。

アプリケーションのロケールまたはクライアントのロケールは、データベースサーバへの要求時にクライアントまたはクライアントライブラリによって使用され、返す結果に使用される文字セットと、エラーメッセージ、警告、その他のメッセージの言語を決定します。データベースサーバは自身のロケールをアプリケーションのロケールと比較し、必要な文字セット変換を判断します。サーバ上のデータベースによって、ロケール定義が異なる場合があります。また、クライアントによってロケールが異なる場合もあります。

ロケールは次のコンポーネントで構成されています。

#### 言語

ISO 639-1 標準の値を使用した 2 文字の文字列です (たとえばドイツ語は DE)。データベースサーバとクライアントのどちらにも、自身のロケールに対する言語の値があります。

データベースサーバは、ロードする言語ライブラリをロケール言語で判断します。データベースの作成時に照合が指定されていない場合、データベースサーバは使用する照合の判断に文字セットと併せてロケール言語も使用します。

クライアントライブラリでは、ロードする言語ライブラリをロケール言語で判断し、データベースからの要求もロケール言語で判断します。

#### 文字セット

文字セットとは、使用しているコードページまたはエンコードのことです。クライアントとサーバのどちらにも文字セットの値がありますが、両者が異なる場合もあります。異なる場合は、文字セット変換を実行して相互運用性を確保します。

このセクションの内容:

#### [ロケール言語 \[588 ページ\]](#)

ロケール言語は、クライアントアプリケーションのユーザによって使用される言語、またはデータベースサーバのユーザによって使用されることが予測される言語です。

#### [言語ラベルの値および言語コード \[588 ページ\]](#)

次の表は、有効な言語ラベルの値と対応する ISO 639 言語コードを示します。

#### [ロケール文字セット \[589 ページ\]](#)

アプリケーションとサーバのロケール定義のいずれにも文字セットがあります。

#### [ロケール情報 \[590 ページ\]](#)

ロケール情報は、PROPERTY、DB\_PROPERTY、CONNECTION\_PROPERTY などの関数を使用して確認できません。

#### [ロケールの設定 \[591 ページ\]](#)

環境を認識するためにデータベースサーバおよびクライアントライブラリで使用するロケールを設定します。

## 1.5.4.4.1 ロケール言語

ロケール言語は、クライアントアプリケーションのユーザによって使用される言語、またはデータベースサーバのユーザによって使用されることが予測される言語です。

クライアントライブラリとデータベースサーバはどちらも同じ方法でロケールの言語コンポーネントを特定します。

1. SALANG 環境変数が指定されている場合は、その値を使用します。
2. Windows では、SALANG 環境変数がない場合、SQL Anywhere 言語レジストリのエントリを確認します。
3. オペレーティングシステムの言語設定を確認します。
4. 上記の設定で言語を特定できなかった場合は、デフォルトで英語になります。

### 関連情報

[ロケール情報 \[590 ページ\]](#)

[SALANG 環境変数 \[551 ページ\]](#)

[インストール時のレジストリ設定 \[568 ページ\]](#)

## 1.5.4.4.2 言語ラベルの値および言語コード

次の表は、有効な言語ラベルの値と対応する ISO 639 言語コードを示します。

言語	ISO 639-1 言語コード	言語ラベル	代替ラベル
アラビア語	AR	arabic	N/A
チェコ語	CS	czech	N/A
デンマーク語	DA	danish	N/A
オランダ語	NL	dutch	N/A
英語	EN	us_english	english
フィンランド語	FI	finnish	N/A
フランス語	FR	french	N/A
ドイツ語	DE	german	N/A
ギリシャ語	EL	greek	N/A
ヘブライ語	HE	hebrew	N/A
ハンガリー語	HU	hungarian	N/A
イタリア語	IT	italian	N/A
日本語	JA	japanese	N/A
韓国語	KO	korean	N/A
リトアニア語	LT	lithuanian	N/A

言語	ISO 639-1 言語コード	言語ラベル	代替ラベル
ノルウェー語	NO	norwegian	norweg
ポーランド語	PL	polish	N/A
ポルトガル語	PT	portuguese	portugue
ロシア語	RU	russian	N/A
中国語 (簡体文字)	ZH	chinese	simpchin
スペイン語	ES	spanish	N/A
スウェーデン語	SV	swedish	N/A
タイ語	TH	thai	N/A
中国語 (繁体文字)	TW	tchinese	tradchin
トルコ語	TR	turkish	N/A
ウクライナ語	UK	ukrainian	N/A

### 1.5.4.4.3 ロケール文字セット

アプリケーションとサーバのロケール定義のいずれにも文字セットがあります。

アプリケーションは、データベースサーバから文字列を要求するときに文字セットを使用します。データベースサーバは、データベースの文字セットをアプリケーションの文字セットと比較し、文字セット変換が必要かどうかを判断します。データベースサーバがデータベースとクライアント間の文字セット変換を実行できない場合は、接続が失敗します。

1. SACHARSET 環境変数が設定されている場合、その値で文字セットが特定されます。  
データベースサーバが SACHARSET を使用するの、照合の指定がない状態で新しいデータベースを作成する場合だけです。
2. 接続文字列で文字セットが指定されている場合は、その文字セットが使用されます。
3. SAP Open Client アプリケーションは、SAP リリースディレクトリの locales サブディレクトリの locales.dat ファイルを確認します。
4. オペレーティングシステムの文字セット情報によって、次の方法でロケールを特定します。
  - Windows オペレーティングシステムの場合、最新の Windows ANSI コードページが使用されます。
  - UNIX プラットフォームでは、LC\_ALL、LC\_MESSAGES、LC\_CTYPE、LANG の各ロケール環境変数が指定された順序で確認されます。この中で最初に設定されていた環境変数の値が、文字セットの決定に使用されます。文字セットをオペレーティングシステムから特定できなかった場合は、デフォルトで iso\_1 (Windows コードページ 28591、ISO 8859-1 Latin 1、ISO 8859-1 Latin-1、iso\_8859-1:1987 と呼ばれる) が使用されます。
5. それ以外のプラットフォームでは、デフォルトでコードページ cp1252 が使用されます。

## 関連情報

[ロケール情報 \[590 ページ\]](#)

[CharSet \(CS\) 接続パラメータ \[53 ページ\]](#)

## 1.5.4.4.4 ロケール情報

ロケール情報は、PROPERTY、DB\_PROPERTY、CONNECTION\_PROPERTY などの関数を使用して確認できます。

次の表は、これらの関数を使用して、クライアント接続、データベース、データベースサーバのロケール情報を返す方法を示します。

システム関数とパラメータ	戻り値
<code>SELECT PROPERTY( 'CharSet' );</code>	データベースサーバの文字セット。通常は、データベースサーバを実行しているコンピュータの文字セット
<code>SELECT PROPERTY( 'DefaultCollation' );</code>	データベースサーバがデータベース作成時に使用するデフォルトの CHAR 照合
<code>SELECT PROPERTY( 'DefaultNcharCollation' );</code>	データベースサーバがデータベース作成時に使用するデフォルトの NCHAR 照合
<code>SELECT PROPERTY( 'Language' );</code>	データベースサーバのロケール言語
<code>SELECT DB_PROPERTY( 'CharSet' );</code>	CHAR データをデータベースに格納するときに使用する文字セット
<code>SELECT DB_PROPERTY( 'NcharCharSet' );</code>	NCHAR データをデータベースに格納するときに使用する文字セット
<code>SELECT DB_PROPERTY( 'MultiByteCharSet' );</code>	CHAR データでマルチバイト文字セットを使用するかどうか (On=使用、Off=使用しない)
<code>SELECT DB_PROPERTY( 'Language' );</code>	データベースの CHAR 照合でサポートしている言語を表す 2 文字のコードをカンマで区切ったリスト
<code>SELECT DB_PROPERTY( 'Collation' );</code>	データベースサーバが使用する CHAR 照合名
<code>SELECT DB_PROPERTY( 'NcharCollation' );</code>	データベースサーバが使用する NCHAR 照合名
<code>SELECT CONNECTION_PROPERTY( 'CharSet' );</code>	クライアントの CHAR データ文字セット
<code>SELECT CONNECTION_PROPERTY( 'NcharCharSet' );</code>	接続に使用する NCHAR データの文字セット
<code>SELECT CONNECTION_PROPERTY( 'Language' );</code>	クライアントが接続に使用する言語

## 1.5.4.4.5 ロケールの設定

環境を認識するためにデータベースサーバおよびクライアントライブラリで使用するロケールを設定します。

### コンテキスト

ロケールは言語と文字セットで構成されています。デフォルトのロケールで問題ない場合は、何の作業も必要ありません。

### 手順

ロケールの変更が必要な場合は、SACHARSET 環境変数か SALANG 環境変数、またはその両方を次のように設定します。

```
SACHARSET=charset  
SALANG=language-code
```

`charset` は有効な文字セットラベルで、`language-code` は有効な言語コードのリストにある 2 文字の言語コードです (EN、DE、ZH など)。

### 結果

ロケールが設定されます。

### 関連情報

[ロケール情報 \[590 ページ\]](#)

[サポートされている文字セット \[580 ページ\]](#)

[照合の考慮事項 \[600 ページ\]](#)

[言語ラベルの値および言語コード \[588 ページ\]](#)

[SACHARSET 環境変数 \[549 ページ\]](#)

[SALANG 環境変数 \[551 ページ\]](#)

## 1.5.4.5 照合

照合とは、特定の文字セットまたはエンコードに対する文字のソートおよび比較の方法です。

サポートされている照合アルゴリズムは、SQL Anywhere 照合アルゴリズム (SACA) と Unicode 照合アルゴリズム (UCA) の 2 つです。SACA を使用すると、ソートが高速、簡潔、実用的になりますが、言語的な正確さは若干低下します。UCA を使

用すると、言語的な処理は正確になりますが、記憶領域の要件と実行時間が多少増加します。UCA はまた、照合の実行方法を制御する照合適合理化オプションもサポートしています。

データベースの照合は、データベースの作成時に設定します。ただし、照合は後から変更できます。

## データベースでの照合

SQL Anywhere 照合アルゴリズム (SACA) は、シングルバイトおよびマルチバイト文字セットの実用的な比較、ソート、大文字小文字変換を提供します。

このアルゴリズムは、必要とする領域が少なく、高速です。インデックスなどの文字列は、マッピング後も元の文字列と長さが同じです。比較、ソート、大文字小文字変換のマッピングでは、文字列の各バイト値ごとの簡単なテーブルルックアップを使用しています。

## シングルバイト文字セット

典型的なシングルバイト文字セットの照合では、各文字のアクセント記号付きとアクセント記号なしのあらゆる形が同じ値にマッピングされており、照合ではアクセント記号は区別されません。同じ文字のアクセント記号付き形とアクセント記号なし形は同じとみなされ、ソートされると隣り合わせになります。

この照合では、アクセント記号を維持したまま、大文字と小文字が変換されます。

## マルチバイト文字セット

マルチバイト文字セットの場合、リードバイトが 256 個の異なる値にマッピングされています。フォロワーバイトはバイナリ値として比較されます。

マルチバイト文字セットのほとんどの照合では、文字セットエンコードにより、文字はリードバイトによって識別される 256 バイト単位のページにグループ化されるので、このマッピング方法で実用的なソート結果が得られます。これらのページと各ページに含まれる文字は、該当する文字セットとして合理的な順序で配置されています。通常、この照合では文字セットにおけるページの順序 (リードバイト) が保持されます。ページによっては、他の特徴に基づいてソートされます。たとえば、日本語コードページ 932 用の 932JPN 照合では、全角文字 (漢字) と半角文字 (カタカナ) をグループ分けしています。

大文字小文字変換は、7 ビットの英文字についてのみ提供されています。

## UTF-8 文字セット

UTF-8 はマルチバイト文字セットです。各文字は 1 ~ 4 バイトで構成されます。UTF8BIN 照合は、UTF-8 文字のソート用に提供されています。

UTF8BIN では、リードバイトが 256 の異なる値にマッピングされており、フォロワーバイトはバイナリ値として比較されます。UTF-8 の文字表現方法と 256 というマッピングの数的制限に起因して、同じ文字のアクセント記号付き形とアクセント記号なし形など、関連文字をグループ化できません。ソートの基準は基本的にバイナリです。

大文字小文字変換は、7ビットの英文字についてのみサポートされています。

このセクションの内容:

[データベースでの照合 \[593 ページ\]](#)

データベースにはいくつかの照合があります。

[SQL Anywhere 照合アルゴリズム \(SACA\) \[594 ページ\]](#)

SQL Anywhere 照合アルゴリズム (SACA) は、シングルバイトおよびマルチバイト文字セットの実用的な比較、ソート、大文字小文字変換を提供します。

[Unicode 照合アルゴリズム \(UCA\) \[595 ページ\]](#)

Unicode 照合アルゴリズム (UCA) は、Unicode 文字セット全体のソートに使用するアルゴリズムです。

[代替照合 \[597 ページ\]](#)

SORTKEY および COMPARE 関数で使用できる互換性の照合は複数あります。

[照合の考慮事項 \[600 ページ\]](#)

データベースの作成時に、データベースで使用する照合を指定します。

[データベースサーバによる新しいデータベースのデフォルトの照合の選択方法 \[604 ページ\]](#)

新しいデータベースが作成され、照合が明示的に指定されていない場合、データベースサーバでは言語と文字セットを使用して照合を決定します。

[データベースの照合の変更 \[605 ページ\]](#)

選択した照合へのアンロードおよび再ロードを実行することで、ある照合から別の照合へデータベースの照合を変更します。

## 関連情報

[データベースの作成 \[265 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

[照合の適合化オプション \[601 ページ\]](#)

### 1.5.4.5.1 データベースでの照合

データベースにはいくつかの照合があります。

## CHAR 照合

CHAR データ型 (CHAR、VARCHAR、LONG VARCHAR など) の照合では、SQL Anywhere 照合アルゴリズム (SACA) が使用される場合と Unicode 照合アルゴリズム (UCA) が使用される場合があります。どちらの場合も、使用される照合は CHAR 照合と呼ばれます。

## NCHAR 照合

NCHAR データ型 (NCHAR、NVARCHAR、LONG NVARCHAR など) の照合では、Unicode 照合アルゴリズム (UCA) が使用される場合と UTF8BIN 照合が使用される場合があります。UTF8BIN 照合では SQL Anywhere 照合アルゴリズム (SACA) が使用されます。

### 大文字小文字とアクセント記号の区別

作成時に大文字小文字を区別するよう指定されなかったデータベースでは、大文字と小文字は区別されません。データベースで大文字と小文字が区別されるようにするには、該当するオプションを指定します。データベースの作成後、データベースを再構築しないで大文字と小文字が区別されるようにすることはできません。

データベースが大文字と小文字を区別するかどうかによって、SACA 照合でも UCA 照合でも大文字と小文字を区別するかどうかが決まり、それにより CHAR 照合と NCHAR 照合で大文字と小文字を区別するかどうかが決まります。

作成時にアクセント記号を区別するよう指定されなかったデータベースでは、アクセント記号は区別されません。データベースでアクセント記号が区別されるようにするには、該当するオプションを指定します。データベースの作成後、データベースを再構築しないでアクセント記号が区別されるようにすることはできません。

データベースがアクセント記号を区別するかどうかは、UCA 照合にのみ影響を与えます。UCA 照合が CHAR 照合や NCHAR 照合で使用されているかどうかには関係ありません。CHAR 照合と NCHAR 照合のどちらにも SACA 照合を使用することを選択した場合、アクセント記号に関するオプションに効力はありません。アクセント記号の区別は SACA 照合の属性であり、該当するオプションをデータベース作成時に使用して指定することはできません。

## 1.5.4.5.2 SQL Anywhere 照合アルゴリズム (SACA)

SQL Anywhere 照合アルゴリズム (SACA) は、シングルバイトおよびマルチバイト文字セットの実用的な比較、ソート、大文字小文字変換を提供します。

このアルゴリズムは、必要とする領域が少なく、高速です。インデックスなどの文字列は、マッピング後も元の文字列と長さが同じです。比較、ソート、大文字小文字変換のマッピングでは、文字列の各バイト値ごとの簡単なテーブルルックアップを使用しています。

### シングルバイト文字セット

典型的なシングルバイト文字セットの照合では、各文字のアクセント記号付きとアクセント記号なしのあらゆる形が同じ値にマッピングされており、照合ではアクセント記号は区別されません。同じ文字のアクセント記号付き形とアクセント記号なし形は同じとみなされ、ソートされると隣り合わせになります。

この照合では、アクセント記号を維持したまま、大文字と小文字が変換されます。

## マルチバイト文字セット

マルチバイト文字セットの場合、リードバイトが 256 個の異なる値にマッピングされています。フォロワーバイトはバイナリ値として比較されます。

マルチバイト文字セットのほとんどの照合では、文字セットエンコードにより、文字はリードバイトによって識別される 256 バイト単位のページにグループ化されるので、このマッピング方法で実用的なソート結果が得られます。これらのページと各ページに含まれる文字は、該当する文字セットとして合理的な順序で配置されています。通常、この照合では文字セットにおけるページの順序 (リードバイト) が保持されます。ページによっては、他の特徴に基づいてソートされます。たとえば、日本語コードページ 932 用の 932JPN 照合では、全角文字 (漢字) と半角文字 (カタカナ) をグループ分けしています。

大文字小文字変換は、7 ビットの英文字についてのみ提供されています。

## UTF-8 文字セット

UTF-8 はマルチバイト文字セットです。各文字は 1 ~ 4 バイトで構成されます。UTF8BIN 照合は、UTF-8 文字のソート用に提供されています。

UTF8BIN では、リードバイトが 256 の異なる値にマッピングされており、フォロワーバイトはバイナリ値として比較されます。UTF-8 の文字表現方法と 256 というマッピングの数的制限に起因して、同じ文字のアクセント記号付き形とアクセント記号なし形など、関連文字をグループ化できません。ソートの基準は基本的にバイナリです。

大文字小文字変換は、7 ビットの英文字についてのみサポートされています。

### 1.5.4.5.3 Unicode 照合アルゴリズム (UCA)

Unicode 照合アルゴリズム (UCA) は、Unicode 文字セット全体のソートに使用するアルゴリズムです。

これにより、言語的に正しい比較、ソート、大文字小文字変換が実現されます。UCA は Unicode 規格の一部として開発されました。UCA は、IBM が開発および保守している International Components for Unicode (ICU) オープンソースライブラリを使用して実装されています。

#### i 注記

デフォルトの UCA ソート順により、ほとんどの言語のほとんどの文字が適切な順序でソートされます。ただし、同じ文字を使用する言語間でソートや比較にさまざまな違いがあるため、UCA ですべての言語について適切なソート順が得られるわけではありません。そのため、ICU は UCA に対して照合適合化オプションを提供しています。

UCA を使用すると、少ない領域と時間で高度な比較、ソート、大文字小文字変換を実現できます。

マッピング後の文字列は元の文字列より長くなります。このアルゴリズムは、複雑な文字を的確に処理できます。

SQL Anywhere 照合アルゴリズム (SACA) とは異なり、Unicode 照合アルゴリズム (UCA) はシングルバイトの UTF-8 文字セットにのみ使用され、各文字を 1 つまたは複数の属性で区別します。文字の場合、属性は基底文字、アクセント記号、大文字小文字です。

文字以外の場合、通常は基底文字だけが属性になります。

UCA は、次の方法で文字を比較します。

- 基底文字を比較します。文字列の基底文字が他の文字列と異なる場合は、その時点で比較が完了します。アクセント記号や大文字小文字の違いは考慮されません。
- データベースでアクセント記号の違いが区別される場合は、アクセント記号が比較されます。アクセント記号が異なる場合、その時点で比較が完了します。大文字小文字の違いは考慮されません。
- データベースで大文字と小文字が区別される場合は、各文字の大文字と小文字が比較されます。

元の文字列の値が同じとみなされるのは、基底文字、アクセント記号、大文字小文字がまったく同じ場合だけです。

### 例

UCA を使用して次の表の第 1 カラムの文字列を比較するとします。後続のカラムには各文字列の 3 つの属性が記載されています。基底文字は同じで、違うのはアクセント記号と大文字小文字だけです。

文字列	基底文字	アクセント記号	大文字小文字
noel	noel	なし、なし、なし、なし	小、小、小、小
noël	noel	なし、なし、アクセント記号、なし、	小、小、小、小
Noel	noel	なし、なし、なし、なし	大、小、小、小
Noël	noel	なし、なし、アクセント記号、なし、	大、小、小、小

次の表は、UCA を使用した場合にアクセント記号と大文字小文字の区別により可能な 4 つの条件によるソート順を示します。

アクセント記号の区別	大文字と小文字の区別	ORDER BY による結果	説明
N	N	Noel, Noël, Noël, noel (順不同)	<ul style="list-style-type: none"> <li>● アクセント記号は区別しない</li> <li>● 大文字と小文字も区別しない</li> <li>● すべての値が同じとみなされる</li> <li>● 順序は要素数 4 の集合内でランダム</li> </ul>
Y	N	Noel, noel (順不同)、その後に Noël, Noël (順不同)	<ul style="list-style-type: none"> <li>● アクセント記号なしが先、アクセント記号ありが後。したがって ë の前に e がくる</li> <li>● 大文字と小文字は区別しない。N と n は 2 つの間でランダム</li> </ul>

アクセント記号の区別	大文字と小文字の区別	ORDER BY による結果	説明
N	Y	Noel, Noël (順不同)、その後に noël, noel (順不同)	<ul style="list-style-type: none"> <li>大文字が先、小文字が後。したがって n の前に N がくる</li> <li>アクセント記号は区別しない。e と è は 2 つの間でランダム</li> </ul>
Y	Y	Noel noel Noël noël	<ul style="list-style-type: none"> <li>アクセント記号なしが先、アクセント記号ありが後。したがって è の前に e がくる</li> <li>大文字が先、小文字が後。したがって n の前に N がくる</li> </ul>

## 関連情報

[照合の適合理化オプション \[601 ページ\]](#)

### 1.5.4.5.4 代替照合

SORTKEY および COMPARE 関数で使用できる互換性の照合は複数あります。

照合ラベル	説明
874THAIBIN	Code Page 874、Windows Thai、ISO8859-11、バイナリ順序
932JPN	Code Page 932、日本語シフト JIS、Microsoft 拡張文字付き
936ZHO	Code Page 936、中国語 (簡体文字)、PRC GBK 2312-80 8 ビットコード
949KOR	Code Page 949、韓国語 KS C 5601-1987 コード、完成型
950ZHO_HK	Code Page 950、中国語 (繁体文字)、Big 5 コード (HKSCS を含む)
950ZHO_TW	Code Page 950、中国語 (繁体文字)、Big 5 コード
1250LATIN2	Code Page 1250、Windows Latin 2、中央/東ヨーロッパ言語
1250POL	Code Page 1250、Windows Latin 2、ポーランド語
1251CYR	Code Page 1251、キリル語
1252BIN	Code Page 1252、Windows Latin 1、バイナリ順序
1252LATIN1	Code Page 1252、Windows Latin 1、西ヨーロッパ言語

照合ラベル	説明
1252LT1ACC	Code Page 1252、Windows Specialty Latin 1、西ヨーロッパ言語、アクセント記号付きの文字は等しくない
1252NOR	Code Page 1252、Windows Latin 1、ノルウェー語
1252SPA	Code Page 1252、Windows Latin 1、スペイン語
1252SWEFIN	Code Page 1252、Windows Latin 1、スウェーデン語/フィンランド語
1253ELL	Code Page 1253、Windows ギリシア語、ISO8859-7 拡張付き
1254TRK	Code Page 1254、Windows Latin 5、トルコ語、ISO 8859-9 拡張付き
1254TRKALT	Code Page 1254、Windows トルコ語、拡張付き ISO8859-9、I-dot と I-no-dot は同じ
1255HEB	Code Page 1255、Windows ヘブライ語、ISO8859-8 拡張付き
1256ARA	Code Page 1256、Windows アラビア語、ISO8859-6 拡張付き
1257LIT	Code Page 1257、リトアニア語
CESU8BIN	CESU-8、Unicode 用の 8 ビットマルチバイトエンコード、バイナリ順序
EUC_CHINA	中国語 (簡体文字) の GB 2312-80 コード
EUC_JAPAN	日本語の EUC JIS X 0208-1990 と JIS X 0212-1990 コード
EUC_KOREA	韓国語の KS C 5601-1992 コード、Johab (組合型)
EUC_TAIWAN	台湾語の Big 5 コード
ISO1LATIN1	ISO8859-1、ISO Latin 1、西ヨーロッパ言語、Latin 1 順序
ISO9LATIN1	ISO8859-15、ISO Latin 9、西ヨーロッパ言語、Latin 1 順序
ISO_1	ISO8859-1、Latin 1、西ヨーロッパ言語
ISO_BINENG	バイナリ順序、英語 ISO/ASCII 7 ビット文字ケースマッピング
UCA	標準のデフォルト UCA 照合
UTF8BIN	UTF-8、Unicode 用 8 ビットマルチバイトエンコード、バイナリ順序

## 代替照合

代替照合は、古いバージョンの SQL Anywhere との互換性や、その他の特殊な用途で使用できます。サポートされている代替照合の一覧を表示するには、次のコマンドを実行します。

```
dbinit -l+
```

## Adaptive Server Enterprise の照合

次の表は、SORTKEY 関数などの機能で使用するためにサポートされている Adaptive Server Enterprise 照合を示します。

説明	照合名	照合 ID
デフォルト Unicode マルチ言語	default	0
CP 850 代替: アクセント記号なし	altnoacc	39
CP 850 代替: 小文字優先	altdict	45
CP 850 西ヨーロッパ: 大文字/小文字の優先指定なし	altnocsp	46
CP 850 スカンジナビア語辞書順	scandict	47
CP 850 スカンジナビア: 大文字/小文字の優先指定なし	scannocp	48
GB ピンイン	gbpinyin	該当なし
バイナリソート	binary	50
Latin-1 英語、フランス語、ドイツ語辞書順	dict	51
Latin-1 英語、フランス語、ドイツ語、大文字/小文字の区別なし	nocase	52
Latin-1 英語、フランス語、ドイツ語、大文字/小文字の優先指定なし	nocasep	53
Latin-1 英語、フランス語、ドイツ語、アクセント記号なし	noaccent	54
Latin-1 スペイン語辞書順	espdict	55
Latin-1 スペイン語、大文字/小文字の区別なし	espnocs	56
Latin-1 スペイン語、アクセント記号なし	espoac	57
ISO 8859-5 ロシア語辞書順	rusdict	58
ISO 8859-5 ロシア語、大文字/小文字の区別なし	rusnocs	59
ISO 8859-5 キリル語辞書順	cyrdict	63
ISO 8859-5 キリル語、大文字/小文字の区別なし	cyrnocs	64
ISO 8859-7 ギリシャ語辞書順	elldict	65
ISO 8859-2 ハンガリー語辞書順	hundict	69
ISO 8859-2 ハンガリー語、アクセント記号なし	hunnoac	70
ISO 8859-2 ハンガリー語、大文字/小文字の区別なし	hunnocs	71
ISO 8859-5 トルコ語辞書順	turdict	72

説明	照合名	照合 ID
ISO 8859-5 トルコ語、アクセント記号なし	turnoac	73
ISO 8859-5 トルコ語、大文字/小文字の区別なし	turnocs	74
CP 874 (TIS 620) タイ語辞書順	thaidict	1
ISO 14651 順序付け標準	14651	22
シフト JIS バイナリ順	sjisbin	179
Unicode UTF-8 バイナリソート	utf8bin	24
EUC JIS バイナリ順	ejisbin	192
GB2312 バイナリ順	gb2312bn	137
CP932 MS バイナリ順	cp932bin	129
Big5 バイナリ順	big5bin	194
EUC KSC バイナリ順	euckscbn	161

## 関連情報

[ICU とは何か、いつ必要になるか \[575 ページ\]](#)

### 1.5.4.5.5 照合の考慮事項

データベースの作成時に、データベースで使用する照合を指定します。

デフォルトの照合は、データベースサーバのオペレーティングシステムで使用されるコードページと言語から推定されます。通常、デフォルト照合は適切な選択ですが、用意されている多数の照合の中からニーズに合った照合を明示的に指定することもできます。特定の言語に対して複数の照合をサポートできます。

データベースのデータに適した文字セットとソート順を使用する照合を選択してください。文字列のソートや比較を詳細に制御することを目的に、照合の適合化オプションを指定することもできます。

使用するデータベース用の照合を選択するには、次の点を考慮します。

- 文字セット変換を使用すると、パフォーマンスが犠牲になり、またシステム設定が複雑になります。このため、文字セット変換の必要がない照合を選択してください。データベースサーバとクライアントが同じ文字セットを使用している場合は、文字セット変換は使用されません。  
文字セット変換を回避するには、クライアントコンピュータのオペレーティングシステムで使用されている文字セットに対応する照合順をデータベースで使用します。クライアントコンピュータのオペレーティングシステムが Windows の場合は、ANSI 文字セットを選択してください。
- クライアントコンピュータでさまざまな文字セットを扱う場合、またはデータベースに Unicode データを格納する必要がある場合、UCA 照合か UTF8BIN 照合またはその両方を使用することを検討します。ただし、UTF-8 以外のマルチバイト文字セットに対しては UCA 照合を使用する必要があります。
- データベースのデータに適した文字セットとソート順を使用する照合を選択します。この条件を満たす照合が複数ある場合もあります。

このセクションの内容:

[照合の適合化オプション \[601 ページ\]](#)

データベースの作成時に UCA 照合を選択すると、オプションで照合の適合化オプションを指定できます。

## 関連情報

[データベースの作成 \[265 ページ\]](#)

[データベースの作成 \(SQL Central の場合\) \[266 ページ\]](#)

[データベースの作成 \(dbinit ユーティリティの場合\) \[267 ページ\]](#)

[推奨文字セットと照合 \[607 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

### 1.5.4.5.5.1 照合の適合化オプション

データベースの作成時に UCA 照合を選択すると、オプションで照合の適合化オプションを指定できます。

UCA 照合を選択しない場合は、適合化構文を使用して大文字と小文字の区別を制御できます。また、COMPARE 関数や SORTKEY 関数を使用してデータの比較やソートを行う場合も適合化オプションを指定できます。

#### スウェーデン語の照合の適合化

V と W の文字をレベル 1 で区別する Swedish Academy の 2005 年度標準に準拠するよう UCA 照合を適合化するには、UCA (locale=swe;sorttype=phonebook) を指定します。sorttype=phonebook を指定しないと、スウェーデン語のロケールでは V と W は同じ文字として認識されます。

#### 日本語の照合の適合化

UCA では、一部のひらがなとカタカナの文字はレベル 3 でのみ区別され、大文字と小文字が区別されない照合ではそれらの違いはなくなります。すべてのひらがな文字間におけるレベル 1 の違いとすべてのカタカナ文字におけるレベル 1 の違いを定義するように UCA 照合を適合化するには、UCA (locale=ja;sorttype=direct;...) を指定します。これらの適合化オプションによって完全に正しいソートセマンティックは提供されませんが、正しい等号セマンティックが提供されます。

## 照合の適合化の概要

照合の適合化オプションは、キーワードと値の組み合わせの形式で指定します。次の表は、サポートされているキーワードを示したものです。使用できる代替形式と値も示します。

### i 注記

照合の適合化オプションを使用して作成したデータベースは、バージョン 10.0.1 以前のデータベースサーバでは起動できません。

キーワード	照合	代替形式	指定可能な値
Locale	UCA	(なし)	任意の有効なロケールコードです。たとえば、en があります。
CaseSensitivity	サポートされているすべての照合	CaseSensitive、Case	<p><b>respect</b></p> <p>文字の大文字小文字の違いが考慮されます。UCA 照合の場合、UpperFirst を指定したことに等しくなります。その他の照合の場合は、照合によって異なります。</p> <p><b>ignore</b></p> <p>文字の大文字小文字の違いは無視されます。</p> <p><b>UpperFirst</b></p> <p>常に、大文字を先にソートします (Aa)。</p> <p><b>LowerFirst</b></p> <p>常に、小文字を先にソートします (aA)。</p>
AccentSensitivity	UCA	AccentSensitive、Accent	<p><b>respect</b></p> <p>文字のアクセント記号の違いが考慮されます。</p> <p><b>ignore</b></p> <p>文字のアクセント記号の違いは無視されます。</p> <p><b>French</b></p> <p>フランス語の規則に従ってアクセント記号の違いが考慮されます。</p>

キーワード	照合	代替形式	指定可能な値
PunctuationSensitivity	UCA	PunctuationSensitive、Punct	<p><b>ignore</b></p> <p>句読表記の違いは無視されます。</p> <p><b>primary</b></p> <p>第 1 レベルのソートを使用します (文字のみを考慮します)。たとえば、a &gt; b になります。</p> <p><b>quaternary</b></p> <p>第 4 レベルのソートを使用します。文字、大文字小文字、アクセント記号、句読表記の順に考慮されます。たとえば、multiByte、multibyte、multi-byte、multi-Byte をソートすると、次のようになります。</p> <ul style="list-style-type: none"> <li>• multiByte</li> <li>• multibyte</li> <li>• multi-Byte</li> <li>• multi-byte</li> </ul> <p>大文字と小文字やアクセント記号を区別しないデータベースでは、quaternary (第 4 レベル) は指定できません。</p>
SortType	UCA	(なし)	<p>使用するソートタイプです。使用できる値は、次のとおりです。</p> <ul style="list-style-type: none"> <li>• phonebook</li> <li>• traditional</li> <li>• standard</li> <li>• pinyin</li> <li>• stroke</li> <li>• direct</li> <li>• posix</li> <li>• big5han</li> <li>• gb2312han</li> </ul> <p>これらのソートタイプの詳細については、<a href="#">Unicode Technical Standard #35</a> を参照してください。</p>

## 関連情報

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

### 1.5.4.5.6 データベースサーバによる新しいデータベースのデフォルトの照合の選択方法

新しいデータベースが作成され、照合が明示的に指定されていない場合、データベースサーバでは言語と文字セットを使用して照合を決定します。

- 言語は、SALANG 環境変数 (存在する場合)、レジストリ、オペレーティングシステムで判断します。
- 文字セットは、SACHARSET 環境変数 (存在する場合) またはオペレーティングシステムで判断します。

このセクションの内容:

[デフォルトの照合の判断 \[604 ページ\]](#)

デフォルトの照合を判断します。

## 関連情報

[SALANG 環境変数 \[551 ページ\]](#)

[SACHARSET 環境変数 \[549 ページ\]](#)

#### 1.5.4.5.6.1 デフォルトの照合の判断

デフォルトの照合を判断します。

## コンテキスト

照合とは、特定の文字セットまたはエンコードに対する文字のソートおよび比較の方法です。データベースの作成時に照合を指定しないと、オペレーティングシステムに基づくデフォルトの照合が使用されます。

## 手順

1. データベースに接続します。

2. 次の文を実行します。

```
SELECT PROPERTY( 'DefaultCollation' );
```

## 結果

データベースのデフォルトの照合が返されます。

## 関連情報

[照合の考慮事項 \[600 ページ\]](#)

[文字セット \[576 ページ\]](#)

## 1.5.4.5.7 データベースの照合の変更

選択した照合へのアンロードおよび再ロードを実行することで、ある照合から別の照合へデータベースの照合を変更します。

## 前提条件

SELECT ANY TABLE および SERVER OPERATOR システム権限が必要です。

デフォルトでは、UNLOAD 文を実行するには SELECT ANY TABLE システム権限が必要です。-gl データベースサーバオプションを使用すると、必要な権限を変更できます。

## コンテキスト

照合はデータベースの作成時に選択されるものであり、データベースを再構築せずに変更することはできません。

## 手順

1. データベースを起動します。

```
START databasefile
```

2. 既存のデータベースの文字セットを確認します。

```
SELECT DB_PROPERTY( 'CharSet' );
```

SQL Anywhere の初期バージョンでは、このプロパティがない場合があります。文字セットは照合名で判断することもできます。たとえば、照合 1252LATIN1 はコードページ 1252 を使用しています。

既存のデータベースの文字セットは、オペレーティングシステムおよびクライアントの文字セットと同じか、互換性があります。そうでない場合は、データベースの再構築が必要になりますが、再構築のプロセスには細心の注意が必要です。

特に、使用しているデータベースの照合が 850LATIN1 であり、使用している SQL Anywhere が初期バージョンであるため文字セット変換がサポートされていないか (バージョン 5 以前) デフォルトで無効になっており (バージョン 6 と 7)、クライアントアプリケーションが標準的な Windows アプリケーションの場合、データベースにコードページ 1252 の文字データが含まれている可能性があります (通常はコードページ 850 に含まれる解釈データです)。

3. 既存のデータベースに格納されているデータの文字セットを確認します。

```
UNLOAD TABLE mytable TO 'mytable-data-in-utf8.dat' ENCODING 'UTF-8';
```

結果をテキストエディタで表示します。

アクセント記号付きデータが正しい場合、データベースに含まれている文字データは Windows ANSI コードページに対応しており、英語と西ヨーロッパ言語の場合はコードページ 1252 です。データが DOS ベースのエディタで正常に表示される場合、文字データは Windows OEM コードページに対応しており、通常は 437 または 850 です。

4. データベースをアンロードします。

データの文字セットがデータベースの文字セットと互換性がない場合、文字セット変換なしでデータをアンロードすることが重要です。使用されている SQL Anywhere によっては、dbunload の内部アンロード機能を使用したり、UNLOAD TABLE 文を使用してデータを手動でアンロードしたりできます。

5. 新しいデータベースを作成し、使用する照合と文字セットを指定し、DBA ユーザ ID とパスワードを DBA および passwd に設定します。

```
dbinit -dba DBA,passwd -z 1252LATIN1 c:%newdatabase.db
```

6. 古いデータベースサーバを停止して、新しいデータベースサーバを起動します。

```
dbsrv17 -n new-server c:%newdatabase.db
```

7. データを新しいデータベースにロードします。

アンロードしたデータとスキーマ (reload.sql) が再ロードに使用するコンピュータの文字セットに対応している場合、dbunload の外部再ロードオプションを使用できます。データは、データベースの正しい文字セットに自動的に変換されません。次に例を示します。

```
dbunload -ix c:%databasefile
```

データのエンコードがデータベースの文字セットと一致してなく、データのロードに LOAD TABLE 文 (内部再ロード) を使用している場合は、ENCODING 句を使用する必要があります。データベースサーバは、LOAD TABLE 文を使用してロードされたデータに対し、デフォルトでは文字セット変換を実行しません。

データのエンコードが作業に使用しているコンピュータのコードページと一致せず、ロードに INPUT 文 (外部再ロード) を使用している場合は、ENCODING 句を使用する必要があります。そうしないと、データベースサーバはデータがコンピュータのネイティブ文字セットであると想定します。

## 結果

新しい照合でデータベースが再作成されます。

### 例

たとえば、照合 ISO1LATIN1 のデータベースがあるが、ユーロ通貨記号をサポートする必要があることに気付いたとします。ISO1LATIN1 はユーロ通貨記号をサポートしませんが、1252LATIN1 のように他のヨーロッパ言語文字セットはサポートするため、データベースをアンロードし、1252LATIN1 の新しいデータベースを作成し、そのデータベースを再ロードします。

## 関連情報

[照合の考慮事項 \[600 ページ\]](#)

[アンロードユーティリティ \(dbunload\) \[1176 ページ\]](#)

[CharSet \(CS\) 接続パラメータ \[53 ページ\]](#)

[-gl データベースサーバオプション \[418 ページ\]](#)

[Changing the Collation and Codepage When Rebuilding an SQL Anywhere Database \(SQL Anywhere データベースの再構築時の照合とコードページの変更\)](#)

## 1.5.4.6 推奨文字セットと照合

数多くの文字セット、コードページ、エンコード、照合の名前が認識されますが、使用言語に応じて、Windows プラットフォームと UNIX プラットフォームでの使用が推奨されるものがいくつかあります。

dbinit -le オプションを使用すると、データベースで使用可能なすべての文字セットラベルのリストを取得できます。dbinit -l オプションを使用すると、データベースで使用可能な照合のリストを取得できます。

文字セットエンコードまたはラベルを指定する必要がある場合は、「文字セットラベル」列の値または dbinit -le によって表示されるいずれかのラベルを使用します。照合を指定する必要がある場合は、「照合」列か「代替照合」列の値または dbinit -l によって表示されるいずれかのラベルを使用します。

### 注記

次の表に示されていない言語については、UTF-8 エンコードを UCA 照合または UTF8BIN 照合と組み合わせて使用してください。

## Windows プラットフォーム

言語	Windows コードページ	文字セットラベル	照合	代替照合
アラビア語	1256	Windows-1256	1256ARA	
中央および西ヨーロッパ言語	1250	Windows-1250	1250LATIN2	
デンマーク語	1252	Windows-1252	1252LATIN1	
オランダ語	1252	Windows-1252	1252LATIN1	
英語	1252	Windows-1252	1252LATIN1	
フィンランド語	1252	Windows-1252	1252SWEFIN	
フランス語	1252	Windows-1252	1252LATIN1	
ドイツ語	1252	Windows-1252	1252LATIN1	
ギリシャ語	1253	Windows-1253	1253ELL	
ヘブライ語	1253	Windows-1253	1255HEB	
イタリア語	1252	Windows-1252	1252LATIN1	
日本語	932	Windows-31J	932JPN	
韓国語	949	IBM949	949KOR	
リトアニア語	1257	Windows-1257	1257LIT	
ノルウェー語	1252	Windows-1252	1252NOR	
ポーランド語	1250	Windows-1250	1250POL	
ポルトガル語	1252	Windows-1252	1252LATIN1	
ロシア語	1251	Windows-1251	1251CYR	
中国語 (簡体文字)	936	GBK	936ZHO	
スペイン語	1252	Windows-1252	1252SPA	
スウェーデン語	1252	Windows-1252	1252SWEFIN	
タイ語	874	TIS-620	874THAIBIN	
中国語 (繁体文字) - 香港	950	Big5-HKSCS	950ZHO_HK	
中国語 (繁体文字) - 台湾	950	Big5	950ZHO_TW	
トルコ語	1254	Windows-1254	1254TRK	1254TRKALT
ウクライナ語	1251	Windows-1251	1251CYR	
西ヨーロッパ言語	1252	Windows-1252	1252LATIN1	

## UNIX プラットフォーム

言語	文字セットラベル	照合	代替照合
アラビア語	ISO_8859-6:1987	UCA	
中央および西ヨーロッパ言語	ISO_8859-2:1987	UCA	
デンマーク語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
オランダ語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
英語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
フィンランド語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
フランス語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
ドイツ語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
ギリシャ語	ISO_8859-7:1987	UCA	
ヘブライ語	ISO_8859-8:1988	UCA	
イタリア語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
日本語	EUC-JP <sup>1</sup>	EUC_JAPAN	
韓国語	EUC-KR	EUC_KOREA	
リトアニア語	(UTF-8を使用)	UCA または UTF8BIN	
ノルウェー語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
ポーランド語	ISO_8859-2:1987	UCA	
ポルトガル語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
ロシア語	ISO_8859-5:1988	UCA	
中国語 (簡体文字)	GB2312	EUC_CHINA	
スペイン語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
スウェーデン語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
タイ語	(UTF-8を使用)	UCA または UTF8BIN	
中国語 (繁体文字) - 香港	Big5-HKSCS	950ZHO_HK	950TWN
中国語 (繁体文字) - 台湾	EUC-TW	EUC_TAIWAN	
中国語 (繁体文字) - 台湾	Big5	950ZHO_TW	
トルコ語	ISO_8859-9:1989	920TRK	
ウクライナ語	ISO_8859-5:1988	UCA	
西ヨーロッパ言語	ISO-8859-15	ISO9LATIN1	ISO1LATIN1

<sup>1</sup> EUC-JP は、SQL Anywhere ラベルである Extended\_UNIX\_Code\_Packed\_Format\_for\_Japanese の代替ラベルです。

## 関連情報

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

### 1.5.4.7 トルコ語文字セットと照合

トルコ語には、**ı**に相当する文字に 2 通りの表記形式があります。

1 つ目の形式は、**I-dot** と呼ばれるもので、次のように表示されます。

**ı, İ**

2 つ目の形式は、**I-no-dot** と呼ばれるもので、次のように表示されます。

**ı, I**

これらの文字は同じ文字の変形として表示される場合でも、トルコ語のアルファベットでは別の文字と見なされます。このような違いに対処するために、トルコ語照合 1254TRK が提供されています。

これらの文字の大文字と小文字の変換に関するトルコ語の規則は、ANSI SQL 標準規則と互換性がありません。たとえば、トルコ語では **ı** の小文字に相当するのは以下の文字です。

**ı**

一方、ANSI 標準ではこれは次の文字になります。

**i**

このような理由から、大文字と小文字を区別しない場合に正しく一致させることができるのは、一致させるテキストがトルコ語か英語/ANSI であるかどうかによります。多くのコンテキストでは、これを区別するだけの十分な情報がないので、そのようなデータベースでは標準外の動作となることがあります。

たとえば、次の文を 1254TRK 照合を使用するデータベースに対して実行するとします。

```
SELECT * FROM syshistory //actual table name is SYSHISTORY
SELECT * FROM fig       //actual name is FIG
```

最初の文はシステムオブジェクトを参照しており、名前の照合には ANSI SQL 変換規則が必要です。2 番目の文はユーザオブジェクトを参照しており、名前の照合にはトルコ語変換規則が必要です。ここで、データベースサーバは、オブジェクトが何であるかを確認できるまでは使用するべき変換規則を判断できず、使用する変換規則が確認できるまではオブジェクトが何であるかを判断できません。この状況においては、システムおよびユーザオブジェクトの両方を満足させるような解決策はありません。

ん。この例では、データベースサーバがトルコ語照合 1254TRK を使用しており、l の小文字が l の大文字と同じであるとは見なされないため、最初の文は失敗します。2 番目の文は成功します。

トルコ語と ANSI 標準には互換性がないため、トルコ語のデータベースに含まれるシステムオブジェクトを参照する場合は、オブジェクト名の大文字と小文字 (そのオブジェクトの作成時に使用された大文字と小文字) を正確に指定する必要があります。上記の最初の文は次のように記述します。

```
SELECT * FROM SYSHISTORY
```

実際は、文字 l のみ大文字と小文字を正しく対応させます。

別の方法として、通常のやり方ではありませんが、次のように文を記述する方法も可能です。

```
SELECT * FROM syshistory //I-no-dot
```

INSERT などのキーワードは、トルコ語データベースでも大文字小文字の区別がありません。SQL Anywhere では、すべてのキーワードが英語の文字のみを使用していると認識しているため、キーワードの一致に ANSI の大文字と小文字の変換規則を使用します。また、組み込み関数などの特定の識別子にこの方法を適用します。ただし、カタログ内に保管されている名前を持つオブジェクトは、上記のように正しい大文字と小文字または文字を使用して指定してください。正しい大文字と小文字を確認するには、システムオブジェクトを定義するシステムビューを調べてください。

このセクションの内容:

[大文字と小文字を区別しないトルコ語データベースのデータ \[611 ページ\]](#)

大文字小文字を区別しないトルコ語データベースには、データを管理するいくつかのルールがあります。

[代替トルコ語照合 1254TRKALT \[612 ページ\]](#)

一部のアプリケーション開発者にとって、トルコ語の文字の問題が重大な問題を引き起こす場合もあります。

### 1.5.4.7.1 大文字と小文字を区別しないトルコ語データベースのデータ

大文字小文字を区別しないトルコ語データベースには、データを管理するいくつかのルールがあります。

たとえば、データ値が次のような場合、

**FIG**

上記のデータへの小文字参照は、次のようになります。

**fig**

同じ l-dot 文字が両方の形式で使用されます。

## 1.5.4.7.2 代替トルコ語照合 1254TRKALT

一部のアプリケーション開発者にとって、トルコ語の文字の問題が重大な問題を引き起こす場合もあります。

正しい解決策は、すべてのオブジェクト参照先で大文字と小文字が正しく対応していることを確認するか、適切な文字 I を使用しているかを確認することです。ただし、トルコ語の規則に合わせず ANSI 規則を優先した方がうまくいく場合もあります

照合 1254TRKALT が提供されています。これは、I-dot と I-no-dot を同等の文字とする以外は 1254TRK と同じです。

この変更に伴う違いをきちんと理解しておく必要があります。1254TRKALT データベースでは、次の文字列の区別がありません。

```
fig
fig
```

これはトルコ語ユーザにとって正しくはありませんが、他のユーザにとっては許容範囲である場合もあります。

2 番目の問題は、ORDER BY を使用する場合に出てきます。次の文字列について考えてみます。

```
ia
1a
1s
is
```

1254TRK データベースでは、文字列の ORDER BY は次のような順序を生成します。

```
1a
1s
ia
is
```

これは、I-no-dot は I-dot より小さいからです。1254TRKALT データベースでは、次のような順序になります。

```
ia
1a
1s
is
```

これは、I-no-dot と I-dot が同等だからです。

## 1.5.5 ログインポリシー

ログインポリシーは、ユーザのデータベース接続を作成するときに適用されるルールセットで構成されます。

ログインポリシーは、ユーザログインのルールのみを管理し、ロールおよび権限とは切り離されています。ログインポリシーは継承されません。データベースのすべてのユーザは、ログインポリシーに割り当てられます。

ログインポリシーは、作成、編集、削除できます。また、ログインポリシーをユーザに割り当てられることができます。

sa\_get\_user\_status システムプロシージャを使用して、ログインポリシーを含め、ユーザの現在のステータスに関する情報を表示します。

このセクションの内容:

### [ルートログインポリシーと継承 \[613 ページ\]](#)

すべての新しいデータベースには、*root* ログインポリシーが含まれています。ルートログインポリシーオプションの値を変更することはできますが、ポリシーは削除できません。

### [ログインポリシーオプションとデフォルト値 \[614 ページ\]](#)

次の表は、ログインポリシーによって管理されるオプションと *root* ログインポリシーのデフォルト値を示します。

### [ログインポリシーの作成 \[618 ページ\]](#)

ルートログインポリシーに基づいてカスタムログインポリシーを作成します。

### [既存のユーザへのログインポリシーの割り当て \[619 ページ\]](#)

ユーザに新しいログインポリシーを割り当てることによって、ユーザに割り当てられたログインポリシーを変更します。

### [ログインポリシーの変更 \[619 ページ\]](#)

ログインポリシーを編集し、そのオプションを変更します。

### [ログインポリシーの削除 \[620 ページ\]](#)

ユーザに割り当てられていないカスタムログインポリシーを削除します。

### [読み込み専用データベースのログインポリシーの管理 \[621 ページ\]](#)

データベースを読み込み専用モードで起動すると、既存するそのデータベースの永続的な状態に基づいてログインポリシーが使用されます。

### [ユーザアカウントの自動ロック解除 \[622 ページ\]](#)

ユーザアカウントは、ログインポリシーで定義された失敗ログインの最大試行回数制限 (`max_failed_login_attempts`) を超過すると、自動的にロックされます。

### 1.5.5.1 ルートログインポリシーと継承

すべての新しいデータベースには、*root* ログインポリシーが含まれています。ルートログインポリシーオプションの値を変更することはできますが、ポリシーは削除できません。

次の場合、ユーザにルートログインポリシーが割り当てられます。

- 新しいユーザを作成し、ログインポリシーを指定していない場合
- アンロードユーティリティ (`dbunload`) を使用して、以前のバージョンの SQL Anywhere で作成されたデータベースを再構築する場合
- アップグレードユーティリティ (`dbupgrad`) または `ALTER DATABASE UPGRADE` 文を使用して、SQL Anywhere バージョン 10 のデータベースをアップグレードする場合

## ログインポリシー設定の継承

root という名前のデフォルトのログインポリシーはデータベースに保存され、すべてのポリシーのデフォルトのオプション値が含まれています。デフォルト以外の設定を使用するには、ルートポリシーを変更するか、新しいポリシーを作成してデフォルトの設定を上書きする内容に変更します。ポリシーは、デフォルト設定を変更して上書きしないかぎり、ルートポリシーのデフォルト設定を継承します。

たとえば、ルートポリシーの max\_connections の値が 5 であるとします。この場合、myPolicy という名前のポリシーを作成して、max\_connections の値を Unlimited に変更します。次に、ユーザを作成し、そのユーザに myPolicy ログインポリシーを割り当てます。ユーザがログインすると、ログインポリシーオプションの設定はルートログインポリシーから継承されますが、max\_connections は Unlimited に設定されたままになります。

ルートポリシーから継承されるデフォルト値について理解しておくことは重要です。これは、ルートポリシーのオプション設定を後で変更すると、デフォルトの設定値に依存しているユーザのポリシーに影響が出るためです。ルートポリシーの値を変更しても、その設定が上書きされたポリシーを割り当てられているユーザには影響しません。

## 関連情報

[二重制御パスワードの設定 \(SQL の場合\) \[1558 ページ\]](#)

### 1.5.5.2 ログインポリシーオプションとデフォルト値

次の表は、ログインポリシーによって管理されるオプションと root ログインポリシーのデフォルト値を示します。

policy-option-name	説明	デフォルト値	適用対象:
auto_unlock_time	MANAGE ANY USER システム権限が付与されていないユーザのアカウントがロックされてから自動的にロック解除されるまでの時間。	Unlimited	MANAGE ANY USER システム権限を持つユーザを除くすべてのユーザ
change_password_dual_control	このオプションの値が ON の場合は、パスワードの設定に 2 名の管理者が必要です。  verify_password_function オプションの設定は、このオプションが ON に設定されている場合は無視されます。これは、パスワードが 2 つの部分に分かれて個別に設定されているためです。検証は実行されません。	OFF	すべてのユーザ
ldap_primary_server	プライマリ LDAP サーバの名前。	(なし)	すべてのユーザ

policy-option-name	説明	デフォルト値	適用対象:
ldap_secondary_server	セカンダリ LDAP サーバの名前。	(なし)	すべてのユーザ
ldap_auto_failback_period	プライマリサーバへの自動フェールバックが試行されるまでの時間 (分単位)。	15 分	すべてのユーザ
ldap_failover_to_std	このオプションは、システムリソースの不足、ネットワークの停止、接続のタイムアウト、または同様のシステム障害が原因でユーザの識別名 (DN) が見つからないために、LDAP サーバでの認証に失敗した場合に、データベースサーバで標準認証を許可するかどうかを制御します。この設定では、LDAP サーバから返された実際の認証の失敗 (ユーザが見つかったが、入力されたパスワードが一致しない場合など) が標準認証へとフェールオーバーされることは許可されません。	ON	すべてのユーザ
ldap_refresh_dn	このオプションが、CREATE LOGIN POLICY または ALTER LOGIN POLICY 文で指定されているときは、現在時刻の値がログインポリシーとともに格納されます。この値は、ユーザ認証によって、ISYSUSER で見つかったユーザの user_dn_cached_at 値と比較されるタイムスタンプです。ポリシー内の値が、ISYSUSER の user_dn_cached_at value 値より新しい場合は、ユーザの識別名 (DN) が検索され、ISYSUSER の user_dn 値が表示されます。  値 NOW は、このポリシーオプションに割り当てられる唯一の有効な値です。これ以外の値はエラーとなります。この値は、協定世界時 (UTC: Coordinated Universal Time) に基づく時間で、データベースサーバのデフォルト形式の文字列として格納されます。	(なし)	すべてのユーザ

policy-option-name	説明	デフォルト値	適用対象:
locked	このオプションの値が ON の場合、ユーザは新しい接続を確立できません。 sa_get_user_status システムプロシージャの reason_locked カラムは、データベースサーバで生成されたユーザのロック理由を示す文字列を返します。	OFF	MANAGE ANY USER システム権限を持つユーザを除くすべてのユーザ
max_connections	このログインポリシーを持つユーザに許可された同時接続の最大数。	Unlimited	SERVER OPERATOR または DROP CONNECTION システム権限を持つユーザを除くすべてのユーザ
max_failed_login_attempts	前回のログイン成功以降のログイン失敗の最大回数。このスレッシュホールドを超えると、このログインポリシーを持つユーザによる新しい接続の確立を自動的に防止します (ロック)。最後のログイン失敗から 1 分が経過すると、SYS_AUTH_DBA_ROLE 互換ロールを持つユーザのロックは解除されます。	Unlimited	すべてのユーザ
max_days_since_login	同一ユーザによる 2 回の連続するログインの間で許容される最大経過日数。このスレッシュホールドを超えると、このログインポリシーを持つユーザによる新しい接続の確立を自動的に防止します (ロック)。	Unlimited	MANAGE ANY USER システム権限を持つユーザを除くすべてのユーザ
max_non_dba_connections	ユーザが確立できる同時接続の最大数。このスレッシュホールドを超えると、このログインポリシーを持つユーザによる新しい接続の確立を自動的に防止します (ロック)。このオプションは、ルートログインポリシーでのみサポートされています。	Unlimited	SERVER OPERATOR または DROP CONNECTION システム権限を持つユーザを除くすべてのユーザ

policy-option-name	説明	デフォルト値	適用対象:
pam_failover_to_std	<p>このオプションの値が ON の場合、PAM ユーザ認証が認証を実行できないときにデータベースがフェイルオーバーして標準認証に切り替えます。</p> <p>標準認証へのフェイルオーバーは、このオプションが ON に設定されていて、次の条件を満たす場合のみ起こります。</p> <ul style="list-style-type: none"> <li>データベースサーバが PAM サポートライブラリをロードできない (ほとんどの UNIX システムでは libpam.so、AIX では libpam.a、Mac OS X では libpam.dylib)。</li> <li>PAM でシステムエラーまたは設定エラーが発生し、ユーザを認証できない。</li> </ul>	ON	すべてのユーザ
pam_service_name	<p>データベースユーザの認証に使用される PAM サービスの名前。この値は、空ではない有効な英数字の ASCII 文字列である必要があります。コンピュータの PAM 設定ファイル (/etc/pam.conf) またはディレクトリ (/etc/pam.d) で定義されている PAM サービスを参照してください。</p>		すべてのユーザ
password_life_time	<p>パスワードの変更が必要となるまでの最大日数。</p>	Unlimited	すべてのユーザ
password_grace_time	<p>ログインは許可されるが、デフォルトの post_login プロシージャから警告が出る、パスワードの有効期限が切れるまでの日数。</p>	0	すべてのユーザ
password_expiry_on_next_login	<p>このオプションの値が ON の場合、次のログイン後に、このログインポリシーを持つユーザのパスワードの有効期限が切れます。</p>	OFF	すべてのユーザ
root_auto_unlock_time	<p>MANAGE ANY USER システム権限が付与されているユーザのアカウントがロックされてから自動的にロック解除されるまでの時間。このオプションは、ルートログインポリシーでのみサポートされています。</p>	1分	MANAGE ANY USER システム権限を持つユーザ

## 1.5.5.3 ログインポリシーの作成

ルートログインポリシーに基づいてカスタムログインポリシーを作成します。

### 前提条件

MANAGE ANY LOGIN POLICY システム権限が必要です。

### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、ログインポリシーを右クリックし、**新規** > **ログインポリシー** をクリックします。
3. ログインポリシー作成ウィザードの指示に従います。

### 結果

新しいログインポリシーが作成されます。

### 次のステップ

新しいログインポリシーをユーザに割り当てます。

### 関連情報

[既存のユーザへのログインポリシーの割り当て \[619 ページ\]](#)

[ログインポリシーの変更 \[619 ページ\]](#)

## 1.5.5.4 既存のユーザへのログインポリシーの割り当て

ユーザに新しいログインポリシーを割り当てることによって、ユーザに割り当てられたログインポリシーを変更します。

### 前提条件

MANAGE ANY USER システム権限が必要です。

### コンテキスト

ログインポリシーは、すべてのユーザに割り当てられています。ユーザからログインポリシーを削除することはできません。ユーザに割り当てられたログインポリシーは変更のみが可能です。

### 手順

1. SQL Central でデータベースに接続します。
2. 左側のウィンドウ枠で、**ユーザ**をクリックします。
3. 右ウィンドウ枠でユーザを右クリックし、**ログインポリシーの設定**をクリックします。
4. **ログインポリシー**リストで、ログインポリシーを選択します。

### 結果

ユーザのログインポリシーが再度割り当てられます。設定は直ちに有効になります。

## 1.5.5.5 ログインポリシーの変更

ログインポリシーを編集し、そのオプションを変更します。

### 前提条件

MANAGE ANY LOGIN POLICY システム権限が必要です。

## コンテキスト

ルートログインポリシーをはじめ、あらゆるログインポリシーを編集できます。

### i 注記

ルートログインポリシーのオプション設定の値を変更する場合、ルートログインポリシーに割り当てられたデフォルト値に依存するすべてのユーザに影響を与えます。これとは逆に、その設定で上書きするカスタムログインポリシーに割り当てられたすべてのユーザは、影響を受けません。

## 手順

1. SQL Central でデータベースに接続します。
2. 左ウィンドウ枠で、[ログインポリシー](#)をクリックします。
3. 右ウィンドウ枠でログインポリシーを右クリックし、[プロパティ](#)をクリックします。  
ルートログインポリシーを変更するには、[ルート](#)を右クリックし、[プロパティ](#)をクリックします。
4. ポリシー値を変更し、[OK](#)をクリックします。

## 結果

ログインポリシーが変更されます。設定は直ちに有効になります。

## 関連情報

[既存のユーザへのログインポリシーの割り当て \[619 ページ\]](#)

[ログインポリシーオプションとデフォルト値 \[614 ページ\]](#)

## 1.5.5.6 ログインポリシーの削除

ユーザに割り当てられていないカスタムログインポリシーを削除します。

## 前提条件

MANAGE ANY LOGIN POLICY システム権限が必要です。

ユーザに割り当てられたままのログインポリシーは削除できません。ユーザを別のログインポリシーに割り当ててから、カスタマイズしたログインポリシーを削除してください。

ルートログインポリシーは削除できません。

## コンテキスト

ルートログインポリシーは削除できませんが、編集は可能です。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[ログインポリシー](#)をクリックします。
3. [ログインポリシー](#)ウィンドウ枠でログインポリシーを右クリックし、[削除](#)をクリックします。
4. [はい](#)をクリックします。

## 結果

ログインポリシーが削除されます。

### 1.5.5.7 読み込み専用データベースのログインポリシーの管理

データベースを読み込み専用モードで起動すると、既存するそのデータベースの永続的な状態に基づいてログインポリシーが使用されます。

割り当てるログインポリシーは、現在のセッションにのみ有効です。

後で読み込み専用モードで起動するデータベースに対してログイン管理が有効な場合、次の制限が適用されます。

- サーバによるログイン管理は、起動前のデータベース状態に基づいて行われます。
- データベースの状態を明示的に変更する文は拒否され、結果はエラーになります。
- データベースサーバは、failed\_login\_attempts や last\_login\_time などの各ユーザに関する動的情報を引き続き維持します。ただし、このような情報は一時的なメモリに格納されるため、データベースを停止すると失われます。データベースは、起動前と同じ状態に戻ります。
- 既存のログイン管理ポリシーによってアカウントがロックされると、ユーザはログインできません。また、ログイン時にパスワードを変更する通常の方法は使用できなくなります。
- 高可用性システムのミラーデータベースまたは読み込み専用スケールアウトシステムのコピーノードとして動作しているためにデータベースが読み込み専用になっている場合、プライマリデータベースで実行された文の結果は読み込み専用データベースに反映されます。また、プライマリサーバで収集された動的情報は読み込み専用データベースに送信され、読み込み専用データベースのために収集された情報とともに一時メモリにマージされます。

## 1.5.5.8 ユーザアカウントの自動ロック解除

ユーザアカウントは、ログインポリシーで定義された失敗ログインの最大試行回数制限 (max\_failed\_login\_attempts) を超過すると、自動的にロックされます。

ロックされた後で、ユーザアカウントのロックを手動で解除するには、MANAGE ANY USER システム権限を付与されたユーザが、ALTER USER ... RESET LOGIN POLICY 文を使用します。ただし、MANAGE ANY USER システム権限を持つすべてのユーザが失敗ログインの最大試行回数によってロックアウトされると、データベースの一部またはすべてのサービスがロックダウンする可能性があります。

自動ロック解除は、失敗したログイン試行によってロックされたアカウントにのみ適用され、他の理由によってロックされたアカウントには適用されません。ユーザのロックされたステータスはログイン時に検証され、そのユーザが指定された自動ロック解除時間と同じかこれを超えている場合、そのユーザはログインを許可され、failed\_login\_attempts カウンタがゼロにリセットされます。

ユーザに付与されたパーミッションに基づき、2つのログインポリシーオプションのいずれかがロック解除の際に検証されます。

MANAGE ANY USER システム権限を持つすべての管理ユーザがログイン試行に失敗することでデータベースからロックアウトされた場合、データベースの一部またはすべてのサービスがロックダウンすることがあります。

このシナリオを回避するために、次の2つのログインポリシーオプションを利用できます。

### root\_auto\_lock\_time

MANAGE ANY USER システム権限を持つユーザの自動ロック解除時間を定義します。このオプションを小さい値 (15分など) に設定します。この値で数時間の上限が強制されたサーバがあります。このオプションは、ルートログインポリシーでのみ設定できます。

### auto\_unlock\_time

他のすべてのユーザの自動ロック解除時間を定義します。デフォルトの時間は UNLIMITED です。つまり、ユーザアカウントは自動的にロック解除されません。自動ロック解除を有効にするには、auto\_unlock\_time オプションを適切な値 (たとえば1時間後に自動ロック解除する場合は60) に設定します。このオプションは、ルートログインポリシーを含む任意のログインポリシーで設定できます。

この値の設定には、MANAGE ANY LOGIN POLICY システム権限が必要です。

## 関連情報

[ログインポリシーオプションとデフォルト値 \[614 ページ\]](#)

## 1.5.6 データベースオプション

SET OPTION 文を使用してデータベースオプションを設定します。

この文の構文は次のとおりです。

```
SET [ EXISTING ] [ TEMPORARY ] OPTION [ userid. | PUBLIC. ]option-name = [ option-value ]
```

ユーザ ID を指定して、特定のユーザに対するオプションを設定できます。PUBLIC ロール名を指定して、オプションを明示的に設定されていないすべてのユーザに対するオプションを設定することもできます。ユーザ ID または PUBLIC を指定しない場合、SET OPTION 文を発行した、現在ログインしているユーザにオプション変更が適用されます。次の例では、PUBLIC オプションに対して値を設定してから、ユーザ Browser に対して別の値を設定します。

```
SET OPTION PUBLIC.isolation_level = 1;
SET OPTION Browser.isolation_level = 2;
```

ユーザ定義オプションを設定することもできますが、ユーザ固有の値を割り当てるには、ユーザ定義オプションに PUBLIC 設定がある必要があります。データベースサーバでは、ユーザ定義オプションへの TEMPORARY 値の設定はサポートされていません。次の例では、ユーザ定義オプションを作成し、DBA ユーザに対して別の値を設定します。

```
SET OPTION PUBLIC.user_defined_option = 'allusers';
SET OPTION DBA.user_defined_option = 'DBAonly';
```

データベースオプションは次の 2 つのスコープレベルで設定できます。

- ロール
- ユーザ

ロールに対するオプション設定は、PUBLIC ロールおよびユーザ拡張ロールの場合にのみ意味があります。ユーザ拡張ロールに対するオプションの設定は、ユーザがログインしている場合にのみ適用されます。ロールを継承するユーザは、そのロールのオプション設定は継承しません。

オプション設定の優先順位は次のとおりです。

1. オプションがユーザに対して明示的に設定されていないかぎり、すべてのユーザは PUBLIC ロールのオプション設定を継承します。
2. 一時的な PUBLIC オプション設定は、永続的な PUBLIC オプション設定よりも優先されます。
3. ユーザオプション設定は、一時的または永続的な PUBLIC ロール設定よりも優先されます。
4. 一時的なユーザオプション設定は、永続的なユーザオプション設定、一時的な PUBLIC オプション設定、永続的な PUBLIC オプション設定よりも優先されます。一時的なユーザオプションは、現在接続しているユーザに対してのみ設定できます。

永続的な設定は、SET OPTION 文を発行するユーザが接続しているデータベースに記録されます。

永続的なオプションをユーザに対して設定すると、対応するオプションも同じように設定されます。永続的な PUBLIC オプションを設定すると、対応する一時的な PUBLIC オプションも同じように設定されます。

一時的な PUBLIC オプション設定が現在の接続とすべての新しい接続に適用されます。通常、オプションの説明に特に明記されていないかぎり、一時的な PUBLIC オプションはその他の既存の接続には適用されません。

一時的なユーザオプション設定は現在の接続にのみ適用されます。

option-value を省略した場合、指定したユーザに対して指定したオプションが削除されます。

一時的または永続的なオプション設定が削除されると、オプション設定は、以前指定した優先順位に従って、次の一時的または永続的な設定に戻されます。次の例では、階層を説明しています。DBA ユーザが次の文を実行するとします。

```
SET OPTION PUBLIC.isolation_level = 0;
SET TEMPORARY OPTION PUBLIC.isolation_level = 1;
SET OPTION isolation_level = 2;
SET TEMPORARY OPTION isolation_level = 3;
SELECT connection_property( 'isolation_level' );
```

現在のユーザ (DBA) の独立性レベルは 3 です。DBA としてログインしている他のユーザは、独立性レベルが永続的に設定されているため、独立性レベルは 2 です。DBA 以外のユーザは、データベースが停止して再起動されるまでは、一時的な独立性レベル 1 です (これらのユーザには独立性レベルは永続的に設定されていないものとします)。

現在のユーザ (DBA) に対する一時的なオプション設定が削除されると、設定は DBA に対する永続的なユーザ設定に戻されます。この内容を次に示します。

```
SET TEMPORARY OPTION isolation_level =;
SELECT connection_property( 'isolation_level' );
```

現在のユーザ (DBA) の独立性レベルは 2 です。ユーザ DBA に対する永続的なオプション設定が削除されると、設定は一時的な PUBLIC 設定に戻されます。この内容を次に示します。

```
SET OPTION isolation_level =;
SELECT connection_property( 'isolation_level' );
```

現在のユーザ (DBA) の独立性レベルは 1 です。PUBLIC に対する一時的なオプション設定が削除されると、設定は永続的な PUBLIC 設定に戻されます。この内容を次に示します。

```
SET TEMPORARY OPTION PUBLIC.isolation_level =;
SELECT connection_property( 'isolation_level' );
```

現在のユーザ (DBA) の独立性レベルは 0 です。

オプションの中には (COMMIT 動作に影響するものなど)、スコープがデータベース全体に及ぶものがあります。これらのオプションを設定するには特定の権限が必要です。isolation\_level などのその他のオプションは、現在の接続にのみ適用でき、特別な権限は必要ありません。

オプション設定への変更がいつ有効になるかは、オプションによって異なります。recovery\_time オプションなどのグローバルデータベースオプションを変更した場合は、次のデータベース起動時から有効になります。他のユーザのデータベースオプションを変更した場合は、そのユーザが次回データベースに再接続したときから有効になります。通常は、現在のユーザ接続に作用するオプションのみが即座に有効になります。たとえば、トランザクションの最中にオプション設定を変更できます。

PUBLIC ロールにのみ設定できるオプションの中には、変更された設定を CONNECTION\_PROPERTY 関数を通じてユーザが参照できなくても、既存の接続に対して即座に有効になるものがあります。こうした動作の例として global\_database\_id オプションがあります。このような理由から、他のユーザがデータベースに接続している間は PUBLIC-only オプションを変更しないでください。

### 警告

カーソルが開いている間はオプション値を変更しないでください。カーソルが開いている間にオプション値を変更した場合、カーソルの中で結果の不整合が生じる可能性があります。たとえば、date\_format オプションを変更した場合、返されるローの一部の形式が古い形式になったり新しい形式になったりすることがあります。新しいオプション値を使用して結果セット内のローが整合性を保って計算されるように、オプション値を変更してからカーソルを開いてください。

### 注記

トルコ語照合を使用しているデータベースや大文字と小文字が区別されるデータベースでは、オプション名の大文字と小文字が正しくないと、SYSOPTION システムビューのクエリや次のようなクエリでは、どのローとも一致しない可能性があります。

```
SELECT * FROM sa_conn_properties( ) WHERE propname = 'BLOCKING';
```

## 例

次の文は、現在ログインしているユーザに対するオプション変更を、ユーザが現在接続しているデータベースで適用します。

```
SET OPTION blocking_timeout = 3;
```

次の文は、現在ログインしているユーザに対するオプション変更を、その接続の間のみ適用します。

```
SET TEMPORARY OPTION blocking_timeout = 3;
```

次の文は、ユーザ Browser に対するオプション変更を、SET OPTION 文を発行するユーザが現在接続しているデータベースで適用します。

```
SET OPTION Browser.blocking_timeout = 3;
```

次の文は、PUBLIC ロールに対する変更を、SET OPTION 文を発行するユーザが現在接続しているデータベースで適用します。この例のオプションの場合、SET OPTION 文を発行するユーザに SET ANY SECURITY OPTION システム権限がある必要があります。

```
SET OPTION PUBLIC.login_mode = 'Standard';
```

次の文は、現在ログインしているユーザに対する ansi\_blanks オプションを、ユーザが現在接続しているデータベースで削除します。現在ログインしているユーザに対する ansi\_blanks オプションは、一時的または永続的な PUBLIC ロール設定に戻されます。

```
SET OPTION ansi_blanks =;
```

このセクションの内容:

### [テンポラリオプション設定の考慮事項 \[626 ページ\]](#)

SET OPTION 文に TEMPORARY キーワードを追加してテンポラリオプションを設定すると、変更の適用期間を変更できます。

### [SQL 文の OPTION 句 \[626 ページ\]](#)

INSERT 文、UPDATE 文、DELETE 文、SELECT 文、UNION 文、EXCEPT 文、INTERSECT 文には、OPTION 句があります。この句を使用すると、有効なすべての PUBLIC またはテンポラリオプション設定より優先されるデータベースオプション設定を、その文でのみ指定できます。

### [データベースオプションの表示方法 \[627 ページ\]](#)

システムプロシージャを使用して、システム定義データベースオプションとユーザ定義データベースオプションを表示できます。

### [オプション設定のモニタリング \[629 ページ\]](#)

オプション設定をモニタするには、sa\_server\_option システムプロシージャと、OptionWatchList プロパティか OptionWatchAction プロパティのどちらかを使用して、特定のデータベースオプションの値が変更されたときのデータベースサーバの動作を指定します。

### [データベースオプションの変更 \(SQL Central\) \[630 ページ\]](#)

データベース、ユーザ、またはロールのデータベースオプションの値を変更します。

### [TDS 通信プロトコルと Command Sequence 通信プロトコルのデータベースオプション設定 \[632 ページ\]](#)

データベースへの接続には、TDS プロトコル (SAP Open Client 接続と jConnect JDBC 接続) と Command Sequence プロトコル (ODBC と Embedded SQL) のどちらも使用できます。

[アルファベット順のデータベースオプションリスト \[632 ページ\]](#)

[さまざまなカテゴリのデータベースオプション \[832 ページ\]](#)

データベースオプションには、互換性、同期、SQL Remote、および Interactive SQL の 4 つのカテゴリがあります。

## 1.5.6.1 テンポラリオプション設定の考慮事項

SET OPTION 文に TEMPORARY キーワードを追加してテンポラリオプションを設定すると、変更の適用期間を変更できます。

通常、オプション値の変更は永続的です。

SET TEMPORARY OPTION 文を使用して、現在のユーザに対するオプションを設定すると、新しいオプション値は現在の接続にかぎり、接続の期間のみ有効になります。テンポラリ設定はサーバメモリにのみ保存され、SYSOPTION システムビューには反映されません。

SET TEMPORARY OPTION 文を使用して PUBLIC オプションを設定すると、新しいオプション値は現在の接続のほか、個別のユーザ設定のない新しい接続でのみ有効になります。その他の既存の接続には適用されません。テンポラリオプションの値は、データベースが停止するまで有効です。テンポラリ設定はサーバメモリにのみ保存され、SYSOPTION システムビューには反映されません。

PUBLIC ロールに対してテンポラリオプションを設定すると、セキュリティ上の利点が得られます。たとえば、login\_mode オプションが有効なときは、データベースは実行中のシステムのログインセキュリティに依存します。このオプションをテンポラリオプション設定として有効にすると、Windows ドメインのセキュリティに依存しているデータベースは、たとえデータベースが停止してローカルコンピュータにコピーされる場合でも、危険にさらされることはありません。この場合、login\_mode オプションによって、たとえばオプションが永久値の Standard に戻ります。このモードでは、統合化ユーザ認証が許可されません。

## 1.5.6.2 SQL 文の OPTION 句

INSERT 文、UPDATE 文、DELETE 文、SELECT 文、UNION 文、EXCEPT 文、INTERSECT 文には、OPTION 句があります。この句を使用すると、有効なすべての PUBLIC またはテンポラリオプション設定より優先されるデータベースオプション設定を、その文でのみ指定できます。

OPTION 句を使用すると、文がマテリアライズドビューを使用する方法とクエリを最適化する方法を指定することもできます。

OPTION 句では、次のデータベースオプションの設定を変更できます。

- isolation\_level
- max\_query\_tasks
- optimization\_goal
- optimization\_level
- optimization\_workload
- user\_estimates

## 関連情報

- [isolation\\_level オプション \[711 ページ\]](#)
- [max\\_query\\_tasks オプション \[731 ページ\]](#)
- [optimization\\_goal オプション \[748 ページ\]](#)
- [optimization\\_level オプション \[749 ページ\]](#)
- [optimization\\_workload オプション \[751 ページ\]](#)
- [user\\_estimates オプション \[823 ページ\]](#)

### 1.5.6.3 データベースオプションの表示方法

システムプロシージャを使用して、システム定義データベースオプションとユーザ定義データベースオプションを表示できます。

#### システム定義オプションの値

次のいずれかの方法で、自分の接続に対するシステム定義オプションのすべての値のリストを取得できます。自分以外の接続に対するシステム定義オプションの値を取得する場合は、SERVER OPERATOR、MONITOR、または DROP CONNECTION のシステム権限が必要です。

##### sa\_conn\_options システムプロシージャ

システム定義オプションの現在の設定をすべてリスト表示するには、sa\_conn\_options システムプロシージャを呼び出します。

```
CALL sa_conn_options;
```

このリストをアルファベット順に並べ替えるには、次の文を実行します。

```
SELECT *  
FROM sa_conn_options( )  
ORDER BY OptionName;
```

結果をフィルタするか、名前以外の要素で並べ替えるには、WHERE 句を使用します。

```
SELECT *  
FROM sa_conn_options( )  
WHERE OptionDescription LIKE '%date%'  
ORDER BY PropNum;
```

##### sa\_conn\_properties システムプロシージャ

現在の接続のシステム定義オプションの設定は、接続プロパティのサブセットとしても取得できます。すべての接続プロパティをリスト表示するには、sa\_conn\_properties システムプロシージャを使用します。

```
CALL sa_conn_properties;
```

このリストをアルファベット順に並べ替えるには、次の文を実行します。

```
SELECT *  
FROM sa_conn_properties( )  
ORDER BY PropName;
```

## SET 文

Interactive SQL では、引数なしで SET 文を実行すると、システム定義接続、データベース、Interactive SQL のオプションの現在の設定がリストされます。

```
SET;
```

1つのシステム定義オプションの現在の値を取得するには、CONNECTION\_PROPERTY システム関数を使用します。

## CONNECTION\_PROPERTY システム関数

次の文を実行すると、ansi\_blanks オプションの値が表示されます。

```
SELECT CONNECTION_PROPERTY ( 'ansi_blanks' );
```

CONNECTION\_PROPERTY システム関数を使用して、ユーザ定義オプションの値を取得することはできません。

Embedded SQL プログラムでは、GET OPTION 文を使用して、Embedded SQL アプリケーション内のオプション (システム定義またはユーザ定義) の値を判断できます。

## PUBLIC ロールにのみ設定できるシステム定義オプション

一部のオプションは PUBLIC ロールにのみ設定が可能で、ユーザや接続には使用できません。このようなデータベースオプションを設定する場合、変更された設定を CONNECTION\_PROPERTY 関数を通じてユーザが参照できなくても、既存の接続に対して即座に有効になります。

PUBLIC ロールにのみ設定できるオプションの現在の値を表示するには、オプション名の SYSOPTION システムビューに対してクエリを実行します。例:

```
SELECT * FROM SYSOPTION WHERE "option" = 'allow_snapshot_isolation'
```

## ユーザ定義オプションの値

SYSOPTIONS システムビューを問合せることにより、ユーザ定義オプションの値を取得できます。

### SYSOPTIONS システムビュー

sa\_conn\_options と sa\_conn\_properties システムプロシージャの結果には、ユーザ設定または PUBLIC 設定よりも優先される、システム定義オプションの接続レベルの設定が含まれます。

ユーザ定義オプションの値は、sa\_conn\_options と sa\_conn\_properties システムプロシージャのどちらからも返されません。代わりに、SYSOPTION または SYSOPTIONS カタログビューを直接問い合わせ、特定のユーザ定義オプション

の値を判断できます。SYSOPTIONS システムビューで次のクエリを実行すると、永続的に設定されたすべてのシステム定義オプションとユーザ定義オプションについて、PUBLIC レベルとユーザレベルの設定がすべて表示されます。

```
SELECT *  
FROM SYSOPTIONS;
```

SET TEMPORARY OPTION 文を使用して特定の接続にサーバ定義オプションを一時的に設定しても、SYSOPTION (または SYSOPTIONS) システムビューは更新されません。接続レベルでオプション設定に加えた一時的な変更は、サーバメモリにのみ保存されます。

## すべてのオプションの値

SQL Central を使用して、テンポラリオプションを除くすべてのオプションの値を取得できます。

### SQL Central

SQL Central でデータベースを選択し、**ファイル > オプション** をクリックします。

## 1.5.6.4 オプション設定のモニタリング

オプション設定をモニタするには、sa\_server\_option システムプロシージャと、OptionWatchList プロパティか OptionWatchAction プロパティのどちらかを使用して、特定のデータベースオプションの値が変更されたときのデータベースサーバの動作を指定します。

### 前提条件

SERVER OPERATOR システム権限が必要です。

### 手順

1. データベースに接続します。
2. OptionWatchList プロパティか OptionWatchAction プロパティのどちらかを指定して sa\_server\_option システムプロシージャを呼び出します。

sa\_server\_option システムプロシージャを使用して、データベースオプションを設定しようとしたときにメッセージを送信するか、エラーを返すことをデータベースサーバに指示します。OptionWatchList プロパティを使用してモニタするオプションのリストを作成し、OptionWatchAction プロパティを使用してモニタされているオプションを設定しようとしたときにデータベースサーバが実行するアクションを指定します。

## 結果

オプションの値がモニタされます。

### 例

次のストアードプロシージャを呼び出して、データベースオプションの `automatic_timestamp`、`float_as_double`、`tsql_hex_constant` をモニタするようデータベースサーバに指示します。

```
CALL dbo.sa_server_option(  
  'OptionWatchList', 'automatic_timestamp,float_as_double,tsql_hex_constant' );
```

次のストアードプロシージャを呼び出して、`OptionWatchList` プロパティで指定されたオプションを設定しようとしたときにエラーを返すようデータベースサーバに指示します。

```
CALL dbo.sa_server_option( 'OptionWatchAction','ERROR' );
```

## 1.5.6.5 データベースオプションの変更 (SQL Central)

データベース、ユーザ、またはロールのデータベースオプションの値を変更します。

### 前提条件

どのユーザも自分の独自のオプションを設定できます。

PUBLIC ロールを含む他のユーザまたはロールに対してデータベースオプションを設定する場合は、オプションで必要になる権限に応じて次のいずれかのシステム権限が必要です。

- SET ANY SYSTEM OPTION
- SET ANY PUBLIC OPTION
- SET ANY SECURITY OPTION
- SET ANY USER DEFINED OPTION

### コンテキスト

データベースのオプションの変更は、PUBLIC ロールのオプションの変更に相当します。

ユーザ定義かどうかにかかわらず、すべてのオプションについて、ユーザ固有の値を割り当てるには PUBLIC の設定が必要です。PUBLIC ロールではオプションの新規作成のみ可能です。

PUBLIC ロールのオプションの値を変更すると、そのオプションの値を明示的に設定したユーザを除くすべてのユーザについて、オプションの値が変更されます。ユーザまたはロール (PUBLIC ロールを除く) のオプションの値を設定すると、PUBLIC ロールによって設定されている値が新しい値に上書きされます。

ユーザまたはロールからオプションを削除すると、そのオプションの PUBLIC 設定が適用されます。PUBLIC ロールからオプションを削除すると、そのオプションがデータベースから削除されます。PUBLIC ロールからオプションを削除するには、先にそれらのオプションのすべてのユーザ設定を削除しておく必要があります。

SQL Central を使用してテンポラリオプションを設定または変更することはできません。テンポラリオプションを設定するには、SET OPTION 文を使用します。

## 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 次のオプションのうちの 1 つを選択してください。

オプション	アクション
PUBLIC ロールのオプションの変更	<ol style="list-style-type: none"> <li>1. 左ウィンドウ枠で<b>ロール</b>をクリックします。</li> <li>2. 右ウィンドウ枠で <i>PUBLIC</i> を右クリックし、<b>オプション</b>をクリックします。</li> </ol>
ロールのオプションの変更	<ol style="list-style-type: none"> <li>1. 左ウィンドウ枠で<b>ロール</b>をクリックします。</li> <li>2. 右ウィンドウ枠で、ロールを右クリックし、<b>オプション</b>をクリックします。</li> </ol> <p><b>i 注記</b></p> <p>ロールに対するオプションの設定は、ユーザ拡張ロールの場合にのみ意味があり、ユーザがログインしている場合にのみ適用されます。ロールを継承するユーザは、オプション設定は継承しません。</p>
ユーザのオプションの変更	<ol style="list-style-type: none"> <li>1. 左ウィンドウ枠で<b>ユーザ</b>をクリックします。</li> <li>2. 右ウィンドウ枠で、ユーザを右クリックし、<b>オプション</b>をクリックします。</li> </ol>

3. ユーザまたはロールのオプションを設定します。指定できる値の候補は、**値**フィールドに表示されます。次のオプションのうちの 1 つを選択してください。

オプション	アクション
オプションの追加	<b>新規</b> をクリックし、オプションの名前と値を指定します。
オプションの変更	<b>値</b> フィールドでオプション値を変更し、 <b>恒久的な設定を行う</b> をクリックします。
オプションの削除	オプションを選択し、 <b>すぐに削除</b> をクリックします。

4. **閉じる**をクリックします。

## 結果

データベース、ユーザ、またはロールのデータベースオプションの値が変更されます。

## 関連情報

[アルファベット順のデータベースオプションリスト \[632 ページ\]](#)

### 1.5.6.6 TDS 通信プロトコルと Command Sequence 通信プロトコルのデータベースオプション設定

データベースへの接続には、TDS プロトコル (SAP Open Client 接続と jConnect JDBC 接続) と Command Sequence プロトコル (ODBC と Embedded SQL) のどちらも使用できます。

TDS と Command Sequence (CmdSeq) の両方を使用しているユーザがいる場合は、ストアードプロシージャを使用して、それらのユーザの初期設定を行うことができます。SQL Anywhere ではこの方法を使用して、デフォルトの SAP Adaptive Server Enterprise の動作に合わせて Open Client と jConnect 接続を設定しています。

初期設定は login\_procedure オプションで制御します。このオプションは、ユーザが接続したときに実行されるストアードプロシージャを指定します。デフォルトの設定では、sp\_login\_environment システムプロシージャが使用されます。この動作は、必要に応じて変更できます。

sp\_login\_environment は、接続が TDS を介して行われているかどうかをチェックします。そうである場合は、sp\_tsq\_environment プロシージャを呼び出して、いくつかのオプションに現在の接続に合った新しいデフォルト値を設定します。

## 関連情報

[login\\_procedure オプション \[720 ページ\]](#)

### 1.5.6.7 アルファベット順のデータベースオプションリスト

このセクションの内容:

[allow\\_nulls\\_by\\_default オプション \[641 ページ\]](#)

NULL か NOT NULL かを指定しないで作成された新規カラムが、NULL 値を含むのを許可するかどうかを制御します。

[allow\\_read\\_client\\_file オプション \[642 ページ\]](#)

クライアントコンピュータ上のファイルの読み込みを許可するかどうかを制御します。

[allow\\_snapshot\\_isolation オプション \[643 ページ\]](#)

スナップショットアイソレーションを有効または無効にします。

[allow\\_write\\_client\\_file オプション \[645 ページ\]](#)

クライアントコンピュータへのファイルの書き込みを許可するかどうかを制御します。

[ansi\\_blanks オプション \[646 ページ\]](#)

文字データがクライアント側でトランケートされる際の動作を制御します。

[ansi\\_close\\_cursors\\_on\\_rollback オプション \[647 ページ\]](#)

WITH HOLD 句で開いたカーソルを、ROLLBACK を実行するときに閉じるかどうかを制御します。

[ansi\\_permissions オプション \[648 ページ\]](#)

DELETE 文と UPDATE 文の権限のチェックを制御します。

[ansi\\_substring オプション \[649 ページ\]](#)

start パラメータまたは length パラメータに負の値が設定された場合の SUBSTRING (SUBSTR) 関数の動作を制御します。

[ansi\\_update\\_constraints オプション \[651 ページ\]](#)

更新可能な範囲を制御します。

[ansinull オプション \[652 ページ\]](#)

NULL 値の解釈を制御します。

[audit\\_log オプション \[654 ページ\]](#)

監査ログのタイプと場所を指定します。

[auditing オプション \[656 ページ\]](#)

データベースでの監査を有効または無効にします。

[auditing\\_options オプション \(システムで使用するために予約済み\) \[658 ページ\]](#)

システムで使用するために予約されています。このデータベースオプションは、sa\_enable\_auditing\_type および sa\_disable\_auditing\_type システムプロシージャによって設定されます。

[auto\\_commit オプション \[658 ページ\]](#)

すべての要求の後にオートコミットを実行します。

[auto\\_commit\\_on\\_create\\_local\\_temp\\_index オプション \[659 ページ\]](#)

ローカルテンポラリテーブルにインデックスを作成する前にデータベースサーバが COMMIT を実行するかどうかを制御します。

[background\\_priority オプション \[廃止予定\] \[660 ページ\]](#)

現在の接続以外の接続のパフォーマンスに対する影響を制限します。

[blob\\_threshold オプション \[SQL Remote\] \[661 ページ\]](#)

Message Agent がロングオブジェクト (BLOB) として処理する値のサイズを制御します。

[blocking オプション \[662 ページ\]](#)

ロック競合に応じる動作を制御します。

[blocking\\_others\\_timeout オプション \[663 ページ\]](#)

現在の接続がロールバックされる前に、現在の接続のローとテーブルのロックを他の接続からブロックできる時間を指定します。

[blocking\\_timeout オプション \[664 ページ\]](#)

トランザクションがロックを獲得するまで、どの程度の期間待機するかを制御します。

[cache\\_estimated\\_value\\_plans オプション \[665 ページ\]](#)

値の推定を使用して、キャッシュされたプランを最適化するかどうかを制御します。

[chained オプション \[666 ページ\]](#)

BEGIN TRANSACTION 文がない場合のトランザクションモードを制御します。

[checkpoint\\_time オプション \[667 ページ\]](#)

データベースサーバがチェックポイントを行わずに実行される最大分数を設定します。

#### [cis\\_option オプション \[668 ページ\]](#)

リモートデータアクセス用のデバッグ情報をデータベースサーバメッセージウィンドウに表示するかどうかを制御します。

#### [cis\\_rowset\\_size オプション \[669 ページ\]](#)

各フェッチに対してリモートサーバから返されるローの数を設定します。

#### [close\\_on\\_endtrans オプション \[670 ページ\]](#)

トランザクションの終了時にカーソルを閉じるかどうかを制御します。

#### [collect\\_statistics\\_on\\_dml\\_updates オプション \[671 ページ\]](#)

INSERT、DELETE、UPDATE などのデータ変更 DML 文の実行中に統計を収集するかどうかを制御します。

#### [compression オプション \[SQL Remote\] \[672 ページ\]](#)

SQL Remote メッセージの圧縮レベルを設定します。

#### [conn\\_auditing オプション \[673 ページ\]](#)

auditing オプションが On に設定されている場合に、接続ごとに監査を有効または無効にします。

#### [connection\\_authentication オプション \[674 ページ\]](#)

認証アプリケーションのデータベースシグネチャに対する、アプリケーションシグネチャの確認に使用する認証文字列を指定します。

#### [connection\\_type データベースオプション \[676 ページ\]](#)

モニタリングアプリケーションからの接続がデータベースまたはデータベースサーバのシャットダウンに影響するかどうかを制御します。

#### [continue\\_after\\_raiserror オプション \[677 ページ\]](#)

RAISERROR 文に応じて動作を制御します。

#### [conversion\\_error オプション \[679 ページ\]](#)

データベースからデータをフェッチするときの、データ型変換障害のレポートを制御します。

#### [cooperative\\_commit\\_timeout オプション \[廃止予定\] \[680 ページ\]](#)

トランザクションログ内の COMMIT エントリがいつディスクに書き込まれるかを管理します。

#### [cooperative\\_commits オプション \[廃止予定\] \[681 ページ\]](#)

コミットがいつディスクに書き込まれるかを制御します。

#### [database\\_authentication オプション \[682 ページ\]](#)

データベースの認証文字列を設定します。

#### [date\\_format オプション \[684 ページ\]](#)

データベースから取り出した日付の形式を設定します。

#### [date\\_order オプション \[686 ページ\]](#)

日付形式の解釈を制御します。

#### [db\\_publisher オプション \[687 ページ\]](#)

データベースパブリッシャのユーザ ID を返します (存在する場合)。

#### [debug\\_messages オプション \[688 ページ\]](#)

DEBUG ONLY 句を含む MESSAGE 文を実行するか否かを制御します。

#### [dedicated\\_task オプション \[689 ページ\]](#)

要求処理タスクを、単一の接続からの要求処理に特化します。

#### [default\\_dbpace オプション \[690 ページ\]](#)

テーブルが作成されるデフォルト DB 領域を変更します。

#### [default\\_timestamp\\_increment オプション \[692 ページ\]](#)

TIMESTAMP または UTC TIMESTAMP がデフォルト値になっているカラムを含むローが挿入または更新されるときに、このカラムの値を一意にしておくためにカラムに追加するマイクロ秒数を指定します。

#### [delayed\\_commit\\_timeout オプション \[693 ページ\]](#)

delayed\_commits オプションが On のときに、アプリケーションが COMMIT を実行する時間から、COMMIT が実際にディスクに書き込まれる時間までの最大遅延を指定します。

#### [delayed\\_commits オプション \[694 ページ\]](#)

COMMIT の後でデータベースサーバからアプリケーションに制御が戻されるタイミングを決定します。

#### [delete\\_old\\_logs オプション \[Mobile Link\]\[SQL Remote\] \[695 ページ\]](#)

トランザクションがレプリケートまたは同期されたときにトランザクションログを削除するかどうかを制御します。

#### [disk\\_sandbox オプション \[696 ページ\]](#)

データベースファイルの読み込み/書き込み操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されるかどうかを制御します。

#### [divide\\_by\\_zero\\_error オプション \[697 ページ\]](#)

ゼロ除算のレポートを制御します。

#### [escape\\_character オプション \(システムで使用するために予約済み\) \[698 ページ\]](#)

システムで使用するために予約されています。

#### [exclude\\_operators オプション \(システムで使用するために予約済み\) \[699 ページ\]](#)

システムで使用するために予約されています。このオプションの設定は変更しないでください。

#### [extended\\_join\\_syntax オプション \[699 ページ\]](#)

複数テーブルのジョインに対して、重複する相関名構文を持つクエリを許可するか、またはエラーとしてレポートするかを指定します。

#### [extern\\_login\\_credentials オプション \[700 ページ\]](#)

リモート接続を行うときに、セッションユーザの外部ログインクレデンシャルを使用するか、既存の (有効な) ユーザの外部ログインクレデンシャルを使用するかどうかを制御します。このオプションは下位互換性のために提供されています。セキュリティ上の理由から、このオプションは指定しないでください。

#### [external\\_remote\\_options オプション \[SQL Remote\] \[701 ページ\]](#)

メッセージリンクパラメータを格納するかどうかを指定します。

#### [fire\\_triggers オプション \[702 ページ\]](#)

データベースにおけるトリガ起動を制御します。

#### [first\\_day\\_of\\_week オプション \[703 ページ\]](#)

何曜日を週の最初にするかを設定します。

#### [for\\_xml\\_null\\_treatment オプション \[704 ページ\]](#)

FOR XML 句を使用するクエリでの NULL 値の処理を制御します。

#### [force\\_view\\_creation オプション \(システムで使用するために予約済み\) \[705 ページ\]](#)

システムで使用するために予約されています。このオプションの設定は変更しないでください。

#### [global\\_database\\_id オプション \[705 ページ\]](#)

DEFAULT GLOBAL AUTOINCREMENT が指定されたカラムの値の範囲を設定します。

#### [http\\_connection\\_pool\\_basesize オプション \[707 ページ\]](#)

データベース接続の公称スレッショルドサイズを指定します。

#### [http\\_connection\\_pool\\_timeout オプション \[708 ページ\]](#)

未使用の接続を接続プールに保持できる時間の上限を指定します。

#### [http\\_session\\_timeout オプション \[709 ページ\]](#)

非アクティブ状態の HTTP セッションが維持されるデフォルトのタイムアウト時間 (分単位) を指定します。

#### [integrated\\_server\\_name オプション \[710 ページ\]](#)

統合化ユーザ認証 (ログイン) 用の Windows ユーザグループメンバーシップの検索に使用するドメインコントローラサーバの名前を指定します。

#### [isolation\\_level オプション \[711 ページ\]](#)

ロック独立性レベルを制御します。

#### [java\\_class\\_path option \[713 ページ\]](#)

クラスの検索先となる一連の追加ディレクトリまたは JAR ファイルを指定します。

#### [java\\_location オプション \[714 ページ\]](#)

データベースの Java VM のパスを指定します。

#### [java\\_main\\_userid オプション \(廃止予定\) \[715 ページ\]](#)

このオプションは推奨されなくなりました。

#### [java\\_vm\\_options オプション \[715 ページ\]](#)

Java VM の起動時にデータベースサーバが使用するコマンドラインオプションを指定します。

#### [log\\_deadlocks オプション \[716 ページ\]](#)

デッドロックレポートのオン/オフを制御します。

#### [login\\_mode オプション \[717 ページ\]](#)

データベースに対する標準、統合化、Kerberos、LDAP、PAM の各ユーザ認証の使用を制御します。

#### [login\\_procedure オプション \[720 ページ\]](#)

ユーザがデータベースログインまたは Web サービスから接続するときに呼び出されるストアードプロシージャを指定します。

#### [materialized\\_view\\_optimization オプション \[722 ページ\]](#)

クエリへの応答を効率化するためにオブティマイザがマテリアライズドビューを使用する方法を指定します。

#### [max\\_client\\_statements\\_cached オプション \[724 ページ\]](#)

クライアントでキャッシュされる文の数を制御します。

#### [max\\_connections オプション \[725 ページ\]](#)

データベースに対して許可する同時接続数を制御します。

#### [max\\_cursor\\_count オプション \[727 ページ\]](#)

接続で一度に使用できるカーソルの最大数を制限するリソースガバナーを制御します。

#### [max\\_parallel\\_statements オプション \[728 ページ\]](#)

BEGIN PARALLEL WORK 文の内部で指定され、データベースサーバがいつでも並列実行できる文の最大数を指定します。

#### [max\\_plans\\_cached オプション \[729 ページ\]](#)

キャッシュに格納される実行プランの最大数を指定します。

#### [max\\_priority オプション \[730 ページ\]](#)

接続の最高優先度レベルを制御します。

#### [max\\_query\\_tasks オプション \[731 ページ\]](#)

データベースサーバがクエリの並列処理に使用できるサーバタスクの最大数を指定します。

#### [max\\_recursive\\_iterations オプション \[732 ページ\]](#)

再帰共通テーブル式が反復できる最大回数を制限します。

#### [max\\_statement\\_count オプション \[733 ページ\]](#)

接続で一度に使用できる準備文の最大数を制限するリソースガバナーを制御します。

#### [max\\_temp\\_space オプション \[734 ページ\]](#)

1つの接続で使用できるテンポラリファイル領域の最大サイズを設定します。

#### [min\\_password\\_length オプション \[736 ページ\]](#)

新しいパスワードの最小長をデータベースに設定します。

#### [min\\_role\\_admins オプション \[737 ページ\]](#)

ロールの管理に必要な管理者の最小数を設定します。

#### [ml\\_remote\\_id オプション \[Mobile Link\] \[738 ページ\]](#)

Mobile Link 同期システム内のリモートデータベースのリモート ID を設定します。

#### [nearest\\_century オプション \[739 ページ\]](#)

文字列から日付への変換で、2桁の年の解釈を制御します。

#### [non\\_keywords オプション \[740 ページ\]](#)

個々の予約済みキーワードを無効にして、識別子として使用できるようにします。

#### [odbc\\_describe\\_binary\\_as\\_varbinary オプション \[742 ページ\]](#)

SQL Anywhere ODBC ドライバの BINARY カラムの記述方法を制御します。

#### [odbc\\_distinguish\\_char\\_and\\_varchar オプション \[743 ページ\]](#)

SQL Anywhere ODBC ドライバの CHAR カラムの記述方法を制御します。

#### [oem\\_string オプション \[744 ページ\]](#)

データベースファイルのヘッダページ内にあるユーザ指定の情報を保存します。

#### [on\\_charset\\_conversion\\_failure オプション \[746 ページ\]](#)

文字の変換中にエラーが発生した場合の動作を制御します。

#### [on\\_tsq\\_error オプション \[747 ページ\]](#)

ストアドプロシージャのエラー処理を制御します。

#### [optimization\\_goal オプション \[748 ページ\]](#)

クエリ処理の最適化の対象を、最初のローを迅速に返すこと、または完全な結果セットを返すコストを最小限に抑えることのどちらかに指定します。

#### [optimization\\_level オプション \[749 ページ\]](#)

クエリオプティマイザが SQL 文のアクセスプランの検索に費やす作業量を制御します。

#### [optimization\\_workload オプション \[751 ページ\]](#)

クエリ処理において、更新と読み込みを組み合わせた負荷に対して最適化するか、または大部分が読み込みベースの負荷 (OLAP) に対して最適化するかを決定します。

#### [parameterization\\_level オプション \[752 ページ\]](#)

クライアント文の自動パラメータ化を制御します。

#### [pinned\\_cursor\\_percent\\_of\\_cache オプション \[753 ページ\]](#)

カーソルを固定するために使用できるキャッシュの割合を指定します。

#### [post\\_login\\_procedure オプション \[754 ページ\]](#)

ユーザが接続したときにアプリケーションが表示する必要のあるメッセージが結果セットに含まれるプロシージャを指定します。

#### precision オプション [756 ページ]

10 進法計算での結果の最大桁数を指定します。

#### prefetch オプション [757 ページ]

クライアントアプリケーションで使用できるようになる前に、ローがクライアント側にフェッチされるかどうかを制御します。

#### preserve\_source\_format オプション [759 ページ]

プロシージャ、トリガ、ビュー、イベントハンドラの元のソース定義をシステムファイルに保存するかどうかを制御します。

#### prevent\_article\_pkey\_update オプション [Mobile Link] [760 ページ]

Mobile Link パブリケーションに関連するテーブルのプライマリキーカラムの更新を制御します。

#### priority オプション [761 ページ]

接続からの要求を実行する優先度レベルを設定します。

#### progress\_messages オプション [762 ページ]

データベースサーバからクライアントに進行メッセージを送信するかどうかを制御します。

#### qualify\_owners オプション [SQL Remote] [764 ページ]

SQL Remote によってレプリケートされる SQL 文で、修飾されたオブジェクト名を使用するかどうかを指定します。

#### query\_mem\_timeout オプション [764 ページ]

要求にメモリが付与されるまでの最大待機時間をミリ秒で設定します。

#### quote\_all\_identifiers オプション [SQL Remote] [765 ページ]

SQL Remote によってレプリケートされる SQL 文で、引用符で囲んだ識別子を使用するかどうかを指定します。

#### quoted\_identifier オプション [766 ページ]

二重引用符内の文字列の解釈を制御します。

#### read\_past\_deleted オプション [767 ページ]

独立性レベル 1 または 2 でコミットされていない削除におけるデータベースサーバ動作を制御します。

#### recovery\_time オプション [768 ページ]

データベースサーバがシステム障害からリカバリするのに要する最長時間を分単位で設定します。

#### remote\_idle\_timeout オプション [769 ページ]

Web サービスのクライアントプロシージャとファンクションで許容される休止時間 (秒数) を制御します。

#### replication\_error オプション [SQL Remote] [770 ページ]

SQL エラーが発生したときに Message Agent で呼び出すスタアドプロシージャを指定できます。

#### replication\_error\_piece オプション [SQL Remote] [771 ページ]

replication\_error オプションとともに機能し、SQL Remote のレプリケーション中に SQL エラーが発生した場合に Message Agent から呼び出される LONG VARCHAR スタアドプロシージャを指定できるようにします。

#### request\_timeout オプション [772 ページ]

1 つの要求を実行できる最大時間を設定します。このオプションを使用すると、接続が大量のサーバリソースを長時間にわたり消費することを防止できます。

#### reserved\_connections オプション [773 ページ]

データベースが標準接続に予約する必要がある同時接続の最小数を制御します。このオプションは、データベースサーバが HTTP/HTTPS 接続を受け入れる場合に役立ちます。

#### reserved\_keywords オプション [775 ページ]

1 つ以上のキーワードを予約キーワードに変更します。

#### [return\\_date\\_time\\_as\\_string オプション \[776 ページ\]](#)

クエリが実行されたときに、DATE、TIME、または TIMESTAMP の値がクライアントアプリケーションに渡される方法を制御します。

#### [rollback\\_on\\_deadlock オプション \[778 ページ\]](#)

デッドロック発生時のトランザクションの処理方法を制御します。

#### [row\\_counts オプション \[779 ページ\]](#)

データベースが開いているときに、クエリ内のローの数がデータベースによって常にカウントされるかどうかを指定します。

#### [save\\_remote\\_passwords オプション \[SQL Remote\] \[780 ページ\]](#)

メッセージリンクに入力されたパスワードを保存します。

#### [scale オプション \[780 ページ\]](#)

計算結果が最大精度でトランケートされる場合の、小数点以下の最小桁数を指定します。

#### [secure\\_feature\\_key オプション \(廃止予定\) \[781 ページ\]](#)

-sf データベースオプションを使用して保護された接続に対して機能を有効にできます。

#### [sort\\_collation オプション \[783 ページ\]](#)

ORDER BY 式に対して SORTKEY 関数の暗黙の使用を許可します。

#### [sql\\_flagger\\_error\\_level オプション \[784 ページ\]](#)

指定した規格にはない SQL への応答を制御します。

#### [sql\\_flagger\\_warning\\_level オプション \[785 ページ\]](#)

指定した規格にはない SQL への応答を制御します。

#### [sr\\_date\\_format オプション \[SQL Remote\] \[787 ページ\]](#)

データベースから取り出した DATE 値の形式を設定します。

#### [sr\\_time\\_format オプション \[SQL Remote\] \[788 ページ\]](#)

データベースから取り出した TIME 値の形式を設定します。

#### [sr\\_timestamp\\_format オプション \[SQL Remote\] \[790 ページ\]](#)

データベースから取り出した TIMESTAMP 値の形式を設定します。

#### [sr\\_timestamp\\_with\\_time\\_zone\\_format オプション \[SQL Remote\] \[792 ページ\]](#)

データベースから取り出した TIMESTAMP WITH TIME ZONE 値の形式を設定します。

#### [st\\_geometry\\_asbinary\\_format オプション \[794 ページ\]](#)

ジオメトリからバイナリへの空間データ値の変換方法を制御します。

#### [st\\_geometry\\_astext\\_format オプション \[795 ページ\]](#)

ジオメトリからテキストへの空間データ値の変換方法を制御します。

#### [st\\_geometry\\_asxml\\_format オプション \[796 ページ\]](#)

ジオメトリから XML への空間データ値の変換方法を制御します。

#### [st\\_geometry\\_describe\\_type オプション \[797 ページ\]](#)

空間データの記述方法を制御します。

#### [st\\_geometry\\_interpolation オプション \[798 ページ\]](#)

空間計算が円弧を含むときに内挿法を制御します。

#### [st\\_geometry\\_on\\_invalid オプション \[799 ページ\]](#)

ジオメトリが基本的な検証に失敗した場合 (1 ポイントの LineString、リングが閉じていない多角形、空間参照系の境界を越えたジオメトリなど) の動作を制御します。

#### [string\\_rtruncation オプション \[801 ページ\]](#)

文字列がトランケートされた場合にエラーを発生させるかどうかを決定します。

#### [subscribe\\_by\\_remote オプション \[SQL Remote\] \[802 ページ\]](#)

SUBSCRIBE BY 値が NULL または空の文字列である場合の解釈を制御します。

#### [subsume\\_row\\_locks オプション \[803 ページ\]](#)

データベースサーバがテーブル用に個別のローロックをいつ取得するかを制御します。

#### [suppress\\_tds\\_debugging オプション \[804 ページ\]](#)

TDS デバッグ情報をデータベースサーバメッセージウィンドウに表示するかどうかを決定します。

#### [synchronize\\_mirror\\_on\\_commit オプション \[805 ページ\]](#)

非同期モードまたは非同期フルページモードにおいて、データベースの変更がミラーサーバへ送信されたことがどの時点で保証されるかを制御します。

#### [tds\\_empty\\_string\\_is\\_null オプション \[806 ページ\]](#)

TDS 接続で空の文字を NULL として返すか、ブランク文字 1 文字を含む文字列として返すかを制御します。

#### [temp\\_space\\_limit\\_check オプション \[807 ページ\]](#)

接続によって使用されるテンポラリファイル領域のサイズを確認し、要求された領域サイズが接続に許容されるサイズよりも大きい場合は要求は失敗になります。

#### [time\\_format オプション \[808 ページ\]](#)

データベースから取り出した時刻の表示形式を設定します。

#### [time\\_zone\\_adjustment オプション \[809 ページ\]](#)

1 つの接続でタイムゾーンの調整を修正できます。

#### [time\\_zone オプション \[810 ページ\]](#)

データベースでタイムゾーンの計算に使用するタイムゾーンを指定します。

#### [timestamp\\_format オプション \[811 ページ\]](#)

データベースから取り出したタイムスタンプの形式を設定します。

#### [timestamp\\_with\\_time\\_zone\\_format オプション \[813 ページ\]](#)

データベースから取り出した TIMESTAMP WITH TIME ZONE 値の形式を設定します。

#### [truncate\\_timestamp\\_values オプション \[815 ページ\]](#)

TIMESTAMP 値の精度を制限します。

#### [trusted\\_certificates\\_file オプション \[817 ページ\]](#)

データベースサーバが LDAP サーバのクライアントとして機能するときに、信頼できる認証局の証明書のリストを含むファイルを指定します。

#### [tsql\\_outer\\_joins オプション \[818 ページ\]](#)

Transact-SQL 外部ジョイン演算子の \*= と =\* を文やビューで使用可能にするかどうかを指定します。

#### [tsql\\_variables オプション \[819 ページ\]](#)

@ 符号を、Embedded SQL ホスト変数名のプレフィクスとして使用できるかどうかを制御します。

#### [updatable\\_statement\\_isolation オプション \[820 ページ\]](#)

isolation\_level オプションが Readonly-statement-snapshot に設定されている場合に更新可能な文に適用する独立性レベルを指定します。

#### [update\\_statistics オプション \[821 ページ\]](#)

接続から統計ガバナーにクエリのフィードバックを送信できるかどうかを制御します。

#### [user\\_estimates オプション \[823 ページ\]](#)

クエリの述部に含まれるユーザ選択性推定をクエリオプティマイザが尊重するか無視するかを制御します。

#### [uuid\\_has\\_hyphens オプション \[824 ページ\]](#)

ユニークな識別子の値を文字列に変換するときのフォーマットを設定します。

#### [verify\\_all\\_columns オプション \[SQL Remote\] \[825 ページ\]](#)

ローカルデータベースで発行した更新を含むメッセージがすべてのカラム値を含んで送信されるかどうかを制御します。

#### [verify\\_password\\_function オプション \[826 ページ\]](#)

パスワード規則を実装します。

#### [verify\\_threshold オプション \[SQL Remote\] \[828 ページ\]](#)

更新がレプリケートされるときにどのカラムが確認されるかを制御します。

#### [wait\\_for\\_commit オプション \[829 ページ\]](#)

データが操作されるときに、いつ外部キー整合性をチェックするかを決定します。

#### [webservice\\_namespace\\_host オプション \[830 ページ\]](#)

DISH サービスの XML ネームスペース仕様内のホスト名を指定します。

#### [webservice\\_sessionid\\_name オプション \[831 ページ\]](#)

セッション管理が使用されているかどうかを判断するために Web サーバが使用する項目を再定義します。

## 1.5.6.7.1 allow\_nulls\_by\_default オプション

NULL か NOT NULL かを指定しないで作成された新規カラムが、NULL 値を含むのを許可するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

## 備考

allow\_nulls\_by\_default オプションは、Transact-SQL との互換性を保つために実装されています。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.2 allow\_read\_client\_file オプション

クライアントコンピュータ上のファイルの読み込みを許可するかどうかを制御します。

## 指定可能な値

On、Off

## デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい (現在の接続のみ)、SET ANY SECURITY OPTION あり	いいえ

## 備考

READ\_CLIENT\_FILE 関数を使用する場合など、クライアントコンピュータからファイルを読み込む場合は、このオプションを有効にしてください。

Tabular Data Stream (TDS) プロトコルについては、クライアントファイルの読み込みがサポートされていません。

## 関連情報

[isql\\_allow\\_read\\_client\\_file オプション \[Interactive SQL\] \[1012 ページ\]](#)

[allow\\_write\\_client\\_file オプション \[645 ページ\]](#)

[isql\\_allow\\_write\\_client\\_file オプション \[Interactive SQL\] \[1013 ページ\]](#)

### 1.5.6.7.3 allow\_snapshot\_isolation オプション

スナップショットアイソレーションを有効または無効にします。

#### 指定可能な値

On、Off

#### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	いいえ	いいえ

## 備考

このオプションは、データベースのスナップショットアイソレーションを有効にするかどうかを指定します。このオプションを On に設定すると、トランザクションがスナップショットアイソレーションを使用した場合、データベースサーバは更新されたローの元のバージョンをテンポラリファイルに記録します。

トランザクションが実行中である場合は、allow\_snapshot\_isolation オプションの設定を変更しても、新しい設定が直ちに有効になるわけではありません。このオプションの設定を Off から On に変更しても、その時点で実行中のトランザクションが完了した後でなければスナップショットは使用できません。このオプションの設定を On から Off に変更した場合は、未完了のスナップショットが完了してからバージョン情報の収集が停止され、新たなスナップショットは開始されません。

データベースについて現在のスナップショットアイソレーションの設定を確認するには、SnapshotIsolationState データベースプロパティの値を問い合わせます。

```
SELECT DB_PROPERTY ( 'SnapshotIsolationState' );
```

### 例

次の文は、データベースのスナップショットアイソレーションを有効にします。

```
SET OPTION PUBLIC.allow_snapshot_isolation = 'On';
```

## 関連情報

[isolation\\_level オプション \[711 ページ\]](#)

[updatable\\_statement\\_isolation オプション \[820 ページ\]](#)

## 1.5.6.7.4 allow\_write\_client\_file オプション

クライアントコンピュータへのファイルの書き込みを許可するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい (現在の接続のみ)、SET ANY SECURITY OPTION あり	いいえ

### 備考

WRITE\_CLIENT\_FILE 関数を使用する場合など、クライアントコンピュータにファイルを書き込む場合は、このオプションを有効にしてください。

Tabular Data Stream (TDS) プロトコルについては、クライアントファイルの書き込みがサポートされていません。

### 関連情報

[isql\\_allow\\_write\\_client\\_file オプション \[Interactive SQL\] \[1013 ページ\]](#)

[allow\\_read\\_client\\_file オプション \[642 ページ\]](#)

[isql\\_allow\\_read\\_client\\_file オプション \[Interactive SQL\] \[1012 ページ\]](#)

## 1.5.6.7.5 ansi\_blanks オプション

文字データがクライアント側でトランケートされるときの動作を制御します。

### 指定可能な値

On、Off

### デフォルト

Off

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('On') によって設定されます。

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

### 備考

ansi\_blanks オプションが効力を持つのは、データベースが文字列の比較で後続ブランクを無視し、フェッチした文字列を文字配列に埋め込む場合に限られます。これは、N の値が M 以上の場合に、データ型 CHAR(N) の値が C char(M) 変数に読み込まれるたびトランケーションエラーを強制的に発生させます。ansi\_blanks が Off に設定されていると、トランケーションエラーは、1 文字以上のブランク以外の文字がトランケートされた場合にのみ発生します。

Embedded SQL で ansi\_blanks オプションが On に設定されている場合、データ型 DT\_STRING の値を挿入するときには、sqlen フィールドを値が保存されているバッファの長さ (最低でも値の長さに末尾の NULL 文字を加えた長さ) に設定する必要があります。ansi\_blanks が Off の場合、長さは NULL 文字の位置でのみ決定されます。ansi\_blanks オプションの値は、接続が確立されたときに決定されます。接続が確立された後で ansi\_blanks オプションの値を変更しても、sqlen Embedded SQL の動作には影響しません。

ブランク埋め込みが有効になっているデータベースでは、このオプションの設定によって、フェッチする式が CHAR または NCHAR であり (VARCHAR または NVARCHAR ではない)、ホスト変数 char または nchar (VARCHAR または NVARCHAR ではない) に格納される場合に、トランケーション警告をクライアントに送信するかどうかが決まります。これらの条件が該当し、ホスト変数が小さすぎるために、フェッチされた式の最大長までブランクが埋め込まれることにより、変数に式

を格納できなくなる場合は、トランケーション警告が発生し、フェッチされた式の最大長まで空白が埋め込まれた場合に式の格納に必要な最小バイト数がインジケータに挿入されます。式が CHAR(N) または NCHAR(N) である場合、返される値の文字セット変換と文字長セマンティクスが考慮され、インジケータが N 以外の値に設定されることがあります。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

### 1.5.6.7.6 ansi\_close\_cursors\_on\_rollback オプション

WITH HOLD 句で開いたカーソルを、ROLLBACK を実行するときに閉じるかどうかを制御します。

#### 指定可能な値

On、Off

#### デフォルト

Off

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

#### 備考

ドラフトの SQL/3 標準では、トランザクションをロールバックするときにすべてのカーソルが閉じている必要があります。デフォルトでは、ロールバックした場合、データベースサーバは、WITH HOLD 句なしで開いたカーソルだけを閉じます。このオプションを使うと、すべてのカーソルを強制的に閉じることができます。

close\_on\_endtrans オプションは、ansi\_close\_cursors\_on\_rollback オプションを上書きします。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[close\\_on\\_endtrans オプション \[670 ページ\]](#)

### 1.5.6.7.7 ansi\_permissions オプション

DELETE 文と UPDATE 文の権限のチェックを制御します。

#### 指定可能な値

On、Off

#### デフォルト

On

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

#### 備考

ansi\_permissions を On にすると、DELETE 文と UPDATE 文に対する ANSI/ISO SQL 標準の権限要求がチェックされます。Adaptive Server Enterprise では、このオプションのデフォルト値は Off です。次の表はこの違いを説明しています。

SQL 文	ansi_permissions を Off にした状態で必要な権限	ansi_permissions を On にした状態で必要な権限
UPDATE	値が設定されるカラムに対する UPDATE 権限	値が設定されるカラムに対する UPDATE 権限 WHERE 句に出現するすべてのカラムに対する SELECT 権限 SET 句の右側のすべてのカラムに対する SELECT 権限
DELETE	そのテーブルに対する DELETE 権限	そのテーブルに対する DELETE 権限 WHERE 句に出現するすべてのカラムに対する SELECT 権限

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

### 1.5.6.7.8 ansi\_substring オプション

start パラメータまたは length パラメータに負の値が設定された場合の SUBSTRING (SUBSTR) 関数の動作を制御します。

#### 指定可能な値

On、Off

#### デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

ansi\_substring オプションを On に設定した場合、SUBSTRING 関数は ANSI/ISO SQL 標準と同じ動作をします。開始オフセットが負または 0 の場合は、文字列の左側が文字以外で埋められているかのように扱われ、このときに負の長さが指定されるとエラーになります。

このオプションを Off に設定した場合、SUBSTRING 関数の動作は以前のリリースの SQL Anywhere と同じになります。負の開始オフセットは、文字列の末尾からのオフセットになり、負の長さは、開始オフセットから左に向かって指定の部分文字列の終端までの長さになります。また、開始オフセット 0 を使用した場合、開始オフセット 1 と同じ結果となります。

このオプションの設定は、BYTE\_SUBSTR 関数の動作には影響しません。SUBSTRING 関数では、正以外の start オフセットまたは負の length を使用しないようにしてください。可能なかぎり、SUBSTRING 関数の代わりに LEFT 関数または RIGHT 関数を使用してください。

### 例

次の例は、ansi\_substring オプションの設定によって SUBSTRING 関数の戻り値がどのように変わるかを示しています。

```
SUBSTRING( 'abcdefgh',-2,4 );
ansi_substring = Off ==> 'gh' // substring starts at second-last character
ansi_substring = On  ==> 'a'  // takes the first 4 characters of
                               // ???abcdefgh and discards all ?

SUBSTRING( 'abcdefgh',4,-2 );
ansi_substring = Off ==> 'cd'
ansi_substring = On  ==> value -2 out of range for destination

SUBSTRING( 'abcdefgh',0,4 );
ansi_substring = Off ==> 'abcd'
ansi_substring = On  ==> 'abc'
```

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.9 ansi\_update\_constraints オプション

更新可能な範囲を制御します。

### 指定可能な値

Off、Cursors、Strict

### デフォルト

Cursors

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このソフトウェアでは、ANSI/ISO SQL 標準で許可されていない更新を許可する複数の拡張がサポートされています。これらの拡張機能を利用して、強力かつ効率的に更新を行うことができます。ただし、予期しない動作が発生することがあります。ユーザのアプリケーションがこれらの言語拡張機能の動作を予期して設計されていない場合、この動作によって、更新内容の消失などの異常事態が発生することがあります。

ansi\_update\_constraints オプションは、更新が、ANSI/ISO SQL 標準で許可される更新に制限されるかどうかを制御します。

このオプションを Strict に設定すると、次の更新は許可されません。

- JOINS を含んだカーソルの更新
- ORDER BY 句に表示されるカラムの更新
- FROM 句は、UPDATE 文では使用できません。

このオプションを Cursors に設定した場合は、カーソルにだけ同じ制限が加えられます。カーソルが FOR UPDATE または FOR READ ONLY で開かれていない場合は、データベースサーバは ANSI/ISO SQL 標準に基づいて、更新可能かどうかを選択します。ansi\_update\_constraints オプションが Cursors または Strict の場合、ORDER BY 句を含むカーソルは、FOR READ ONLY をデフォルトにします。それ以外の場合は、引き続き FOR UPDATE をデフォルトにします。

ODBC、JDBC、ADO.NET、OLE DB では、文の更新可能性は明示的な READ ONLY です。

Embedded SQL では、SQL プリプロセッサの -m HISTORICAL オプションが指定されていないかぎり、文の更新可能性は明示的な READ ONLY です。

デフォルトでは、ストアプロシージャのカーソルは明示的な FOR UPDATE または READ ONLY ではありません。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.10 ansinull オプション

NULL 値の解釈を制御します。

### 指定可能な値

On、Off

### デフォルト

On

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('Off') によって設定されます。

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、主に Transact-SQL (Adaptive Server Enterprise) との互換性を保つために実装されています。ansinull オプションは、NULL 定数を持つ比較述部の結果に影響します。また、NULL 値でグループ化されたクエリに対して発行される警告にも影響します。

ansinull を On にすると、ANSI 3 値的論理が WHERE 句、HAVING 句、または On 状態におけるすべての述部比較で使用されます。= または != を使用した NULL との比較はすべて unknown と評価されます。

ansinull を Off に設定すると、データベースサーバは次の 4 つの条件に対して 2 値的論理を使用します。

```
expr = NULL
```

```
expr != NULL
```

```
expr = @var // @var はプロシージャ変数またはホスト変数
```

```
expr != @var
```

いずれの場合も、述部は true または false と評価され、unknown と評価されることはありません。このような比較では、NULL 値は各ドメインで特別値として処理され、2 つの NULL 値の等号 (=) 比較は true になります。式 `expr` は、相対的に単純な式で、カラム、変数、リテラルのみを参照し、サブクエリや関数を許可しない必要があります。

ansinull を On に設定した場合、NULL 値が少なくとも 1 つ含まれている式の集合関数の評価では、COUNT(\*) を除き、警告 (SQLSTATE '01003') が生成されることがあります。

ansinull を Off に設定した場合、この警告は表示されません。

### i 注記

- WHERE、HAVING、または ON 句が含まれているすべての SQL 文が、ansinull オプションの影響を受けます。また、そうした文の内側にあるすべての式も ansinull オプションの影響を受けます。
- Adaptive Server Enterprise 12.5 では、ansinull が Off に設定された状態における NULL パターン文字列を持つ LIKE 述部の動作が変更されています。SQL Anywhere では、LIKE 述部は従来どおり ansinull の設定に影響を受けません。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.11 audit\_log オプション

監査ログのタイプと場所を指定します。

### 指定可能な値

```
[ TRANSLOG
 | SYSLOG
 | FILE ( filename_prefix='path-and-filename'; [ target-parameter [;...] ] )
]
target-parameter :
  target-parameter-name = target-parameter-value[;...]
target-parameter-name
{
  | max_size;
  | num_files;
  | flush_on_write;
  | compressed;
}
```

### パラメータ

パラメータ名	値
<i>filename_prefix</i>	パス付きまたはパスなしのログファイル名のプレフィクス。すべてのログファイルには、.etd という拡張子が付きます。フルパスを指定しない場合、データベースが存在するディレクトリをルートディレクトリとして使用します。このパラメータは必須です。
<i>max_size</i>	ファイルの最大サイズ (バイト単位)。デフォルトは 0 で、これはファイルサイズに上限がないことを意味し、ディスク領域が許す限り大きくなっていきます。指定したサイズに達すると、新しいファイルが開始されます。
<i>num_files</i>	イベントトレース情報が書き込まれるファイルの数。 <i>max_size</i> が設定されている場合にのみ、この設定を使用します。すべてのファイルが指定した最大サイズに達すると、データベースサーバは最も古いファイルを上書きします。
<i>flush_on_write</i>	記録されるイベントが発生するたびにディスクバッファをフラッシュするかどうかを制御するブール値。デフォルトは ON です。このパラメータをオンにすると、多くのトレースイベントが記録される場合、データベースサーバのパフォーマンスが低下することがあります。
<i>compressed</i>	ディスク領域を節約するためのログファイルの圧縮を制御するブール値。デフォルトは OFF です。

## デフォルト

空の文字列

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×

## 備考

このオプションは、監査イベントを書き込む場所を指定します。audit\_log オプションを指定しない場合、または空の文字列を指定する場合、デフォルトでは監査イベントはトランザクションログ (TRANSLOG) に記録されます。

監査が有効なときにトランザクションログをオフにするには、audit\_log を FILE または SYSLOG に設定する必要があります。audit\_log をトランザクションログに設定すると、トランザクションログをオフにすることはできません (ファイルが指定されるまで)。

audit\_log を SYSLOG に設定すると、イベントはオペレーティングシステムのイベントトレースログに記録されます。イベントトレースログは Windows と UNIX で異なります。

- Windows オペレーティングシステムでは、イベントトレースログは [Windows イベントログ](#) です。イベントは、SQLANY で始まるソース名を持ち、イベントビューアーユーティリティで、[▶ イベントビューアー \(ローカル\) ▶ Windows ログ ▶ アプリケーション](#) ▶ とナビゲートすることで表示できます。
- UNIX オペレーティングシステムでは、イベントトレースログは syslog 機能です。

重複するターゲット名は使用できません。FILE、TRANSLOG、SYSLOG が 2 回指定されると、エラーが返されます。

ディスクサンドボックスが有効になっている場合、audit\_log オプションで参照されるファイルはアクセス可能な場所に存在している必要があります。

audit\_log の値が無効な場合、または監査ログのターゲットを開始できない場合、エラーが返され、以前の audit\_log の値をリストアします。以前の値をリストアできない場合、audit\_log オプションはトランザクションログをデフォルトにして、データベースサーバメッセージウィンドウにメッセージが表示されます。ただし、以前の audit\_log オプションの値は保持されます。

ファイルへのログインが失敗する場合、監査イベントが FILE ターゲットにない可能性があります。ログインエラーが発生すると、データベースサーバメッセージウィンドウにメッセージが表示され、最初に発生したエラーを示します。

audit\_log オプションを機能させるには、auditing オプションを On に設定し、sa\_enable\_auditing\_type システムプロシージャを使用して、どのタイプの情報を監査するのかを指定してください。

FILE ターゲットを指定すると、データベースは **audit** という名前の内部的なトレースイベントセッションを使用し、指定したファイルに監査イベントを記録します。内部的なトレースイベントセッションは変更できません。監査で使用するトレースイベントセッションは、ユーザトレースイベントセッションとは次のように異なります。

- filename\_prefix パラメータでフルパスを指定しない場合、データベースが存在するディレクトリをルートディレクトリとして使用します。
- flush\_on\_write パラメータはデフォルトで ON に設定されます。

#### 例

次の文は、監査ログをデータベースのトランザクションログに設定します。

```
SET OPTION PUBLIC.audit_log = 'TRANSLOG'
```

次の文は、監査ログをフィルターターゲットに設定します。

```
SET OPTION PUBLIC.audit_log = 'FILE(filename_prefix=c:¥audit)'
```

## 関連情報

[データベースアクティビティの監査 \[1609 ページ\]](#)

[イベントトレースデータ \(ETD\) 診断ログファイルの内容の表示 \[969 ページ\]](#)

[auditing オプション \[656 ページ\]](#)

[イベントトレースデータ \(ETD\) ファイル管理ユーティリティ \(dbmanageetd\) \[1104 ページ\]](#)

## 1.5.6.7.12 auditing オプション

データベースでの監査を有効または無効にします。

### 指定可能な値

ON、OFF

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×

## 備考

このオプションは、監査のオンとオフを切り替えます。

監査とは、データベース内のイベントに関する詳細を記録することをいいます。監査では、パフォーマンスコストがかかるセキュリティ機能がいくつか提供されています。

監査ログをトランザクションログに設定した場合、次の制約があります。

- データベースの監査をオンにした場合、トランザクションログの使用を停止することはできません。この場合、トランザクションログを停止するためには、監査をオフにする必要があります。
- 監査をトランザクションログに設定しているデータベースを読み込み専用モードで起動することはできません。

auditing オプションを On に設定し、監査オプションを指定しない場合は、すべてのタイプの監査情報が記録されます。また、監査する情報のタイプを制御するには、sa\_enable\_auditing\_type システムプロシージャと sa\_disable\_auditing\_type システムプロシージャを使用してください。

### 例

次の文は、データベースの監査をオンにします。

```
SET OPTION PUBLIC.auditing = 'On';
```

## 関連情報

[データベースアクティビティの監査 \[1609 ページ\]](#)

[audit\\_log オプション \[654 ページ\]](#)

### 1.5.6.7.13 auditing\_options オプション (システムで使用するために予約済み)

システムで使用するために予約されています。このデータベースオプションは、sa\_enable\_auditing\_type および sa\_disable\_auditing\_type システムプロシージャによって設定されます。

### 1.5.6.7.14 auto\_commit オプション

すべての要求の後にオートコミットを実行します。

#### 指定可能な値

On、Off

#### デフォルト

Off

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	いいえ	いいえ	いいえ
一時的な設定が許可されているかどうか	いいえ	はい (現在の接続のみ)	いいえ

#### 備考

このオプションを On に設定すると、データベースサーバはすべての要求の後に自動的にコミットします。このオプションは、1 つの接続に対してのみ一時的に設定できます。

アプリケーションで特定のドライバ API を使用してオートコミットを有効にすると、接続先のデータベースサーバがバージョン 17 の場合、SQL Anywhere JDBC、ODBC、ADO.NET、OLE DB ドライバによって自動的に auto\_commit オプションが On に設定されます。バージョン 16 以前のデータベースサーバの場合、ドライバによって、クライアント側でオートコミットを処理するように戻されます。デフォルトでは、上記のドライバではオートコミットが有効です。

### i 注記

JDBC、ODBC、ADO.NET、OLE DB などの API を使用している場合、auto\_commit サーバオプションを直接設定しないでください。オートコミットを有効化または無効化するには、API 固有のメカニズムを使用してください。たとえば、ODBC では SQLSetConnectAttr を使用して SQL\_ATTR\_AUTOCOMMIT 接続属性を設定します。API を使用する場合、ドライバでオートコミットの現在の設定を追跡できます。

### i 注記

同じ名前の Interactive SQL オプションが設定されないように、Interactive SQL セッションからデータベースオプションを設定するには、BEGIN ブロックを使用します。

```
BEGIN
    SET TEMPORARY OPTION AUTO_COMMIT = 'ON';
END;
```

データベースオプションの新しい設定を確認するには、次の Interactive SQL コマンドを使用します。

```
SET;
```

### i 注記

auto\_commit オプションは chained オプションとは異なります。auto\_commit オプションを On に設定すると、データベースサーバはすべての要求の後にコミットします。chained オプションを Off に設定すると、データベースサーバは各文の後にコミットします。ストアドプロシージャを実行するときには、次の区別が非常に重要です。chained オプションを Off に設定すると、プロシージャ内にある個別の各文の実行後にコミット要求が生成されます。auto\_commit オプションを On に設定すると、プロシージャ全体の実行が終了した後で単一のコミット要求が生成されます。オートコミットが必要である場合は、chained オプションよりも auto\_commit オプションを使用するほうが適切です。

## 1.5.6.7.15 auto\_commit\_on\_create\_local\_temp\_index オプション

ローカルテンポラリテーブルにインデックスを作成する前にデータベースサーバが COMMIT を実行するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションを On に設定すると、ローカルテンポラリテーブルにインデックスを作成する前にデータベースサーバが COMMIT を実行します。COMMIT が失敗した場合は、操作が失敗するかロールバックされます。このオプションを Off に設定すると (デフォルト)、COMMIT は実行されず、インデックスに依存しているプロシージャをアトミック操作 (トリガ、UPDATE 文、INSERT 文、DELETE 文、FROM 句など) の一部として使用できるようになります。

## 1.5.6.7.16 background\_priority オプション [廃止予定]

現在の接続以外の接続のパフォーマンスに対する影響を制限します。

## 指定可能な値

On、Off

## デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり

	PUBLIC ロール	現在のユーザ	他のユーザ
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションを On に設定すると、要求は Background 優先度レベルで実行されます。Off に設定すると、要求は priority オプションで指定した優先度レベルで実行されます。

クエリ内並列処理は、background\_priority が On に設定されている接続には使用されません。

## 関連情報

[priority オプション \[761 ページ\]](#)

[max\\_priority オプション \[730 ページ\]](#)

## 1.5.6.7.17 blob\_threshold オプション [SQL Remote]

Message Agent がロングオブジェクト (BLOB) として処理する値のサイズを制御します。

### 指定可能な値

整数 (バイト)

### デフォルト

256

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

blob\_threshold オプションより長い値は、BLOB としてレプリケートされます。つまり、細かく分割して各部分をレプリケートしてから、SQL 変数を使用し、受信者サイトでその分割された部分を連結して再構成します。

各 SQL 文をメッセージの中に収める必要があるため、このオプションの値にメッセージサイズ (デフォルトは 50 KB) よりも大きいサイズを設定しないでください。

## 1.5.6.7.18 blocking オプション

ロック競合に応じる動作を制御します。

### 指定可能な値

On、Off

### デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり

	PUBLIC ロール	現在のユーザ	他のユーザ
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

blocking オプションを On に設定すると、あるトランザクションが取得しようとしたロックが既存のロックと競合している場合、競合するすべてのロックが解放されるか、blocking\_timeout に設定されたタイムアウト時間が経過するまで、そのトランザクションは待機状態となります。blocking\_timeout ミリ秒以内にロックが解放されない場合は、待機しているトランザクションにエラーが返されます。blocking を Off に設定した場合は、トランザクションが競合するロックを取得しようすると、エラーが返されます。

特定の操作 (マテリアライズドビューの再表示など) を行う間、blocking 接続プロパティの値を一時的に Off に設定できますが、それによってエラーが発生する場合があります。

## 関連情報

[blocking\\_timeout オプション \[664 ページ\]](#)

### 1.5.6.7.19 blocking\_others\_timeout オプション

現在の接続がロールバックされる前に、現在の接続のローとテーブルのロックを他の接続からブロックできる時間を指定します。

このオプションを使用すると、優先度の低いタスクが指定時間を超えて他の接続をブロックすることを防止できます。

#### 指定可能な値

整数 (ミリ秒単位)

#### デフォルト

0

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	×	×	×
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションを 0 に設定すると、現在の接続で他の接続を無制限にブロックできます。

このオプションで指定された時間 (ミリ秒) が経過しても現在の接続で別の接続がブロックされていると、現在の接続がロールバックされます。現在の接続が要求中である場合は、要求が中断されます。接続がロールバックされると、エラーが返されます。要求を中断する必要がある場合は、エラーが 2 回返されることがあります。接続がアイドルの場合は、すぐにロールバックされます。ロールバックされた場合、アプリケーションで失敗または遅延後の再試行を準備する必要があります。

### i 注記

LOCK TABLE `table-name` WITH HOLD で作成したロックは、ロールバックされます。blocking\_others\_timeout エラーのために操作がロールバックされたことを示すエラーが接続で受信されると、その接続が切断されるまですべての要求でエラーが返されます。

## 関連情報

[スキーマロックの自動解放 \(Interactive SQL\) \[1032 ページ\]](#)

[blocking\\_timeout オプション \[664 ページ\]](#)

## 1.5.6.7.20 blocking\_timeout オプション

トランザクションがロックを獲得するまで、どの程度の期間待機するかを制御します。

## 指定可能な値

整数 (ミリ秒単位)

## デフォルト

0

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

blocking オプションを On に設定すると、トランザクションが取得しようとしたロックが既存のロックと競合している場合は、blocking\_timeout に設定された時間 (ミリ秒単位) が経過するまで、競合するロックが解放されるのを待ちます。blocking\_timeout ミリ秒以内にロックが解放されない場合は、待機しているトランザクションにエラーが返されます。

このオプションを 0 に設定すると、ロックを獲得しようとしているすべてのトランザクションは、競合するトランザクションがロックを解放するまで待機します。

## 関連情報

[blocking オプション \[662 ページ\]](#)

[blocking\\_others\\_timeout オプション \[663 ページ\]](#)

[request\\_timeout オプション \[772 ページ\]](#)

## 1.5.6.7.21 cache\_estimated\_value\_plans オプション

値の推定を使用して、キャッシュされたプランを最適化するかどうかを制御します。

## 指定可能な値

On、Off

## デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

On に設定すると、値の推定を使用してキャッシュされたプランを最適化します。値の推定は、プランがキャッシュされたときに文で提供される実際の値の影響を受けます。Off に設定すると、提供される実際の値にかかわらず、キャッシュされたプランを最適化します。

### i 注記

Off の場合、ソフトウェアのバージョン 16 およびそれ以前のバージョンと同じ動作をします。

## 1.5.6.7.22 chained オプション

BEGIN TRANSACTION 文がない場合のトランザクションモードを制御します。

## 指定可能な値

On、Off

## デフォルト

On

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('Off') によって設定されます。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

Transact-SQL トランザクションモードを制御します。非連鎖モード (chained = Off) では、明示的な BEGIN TRANSACTION 文が実行されてトランザクションを開始しないかぎり、各文が個々にコミットされます。連鎖モード (chained = On) では、データ検索文または修正文の前にトランザクションが暗黙的に開始されます。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

### 1.5.6.7.23 checkpoint\_time オプション

データベースサーバがチェックポイントを行わずに実行される最大分数を設定します。

#### 指定可能な値

整数

#### デフォルト

60

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

## 備考

このオプションは、いつチェックポイントを実行すべきかを決定するために、`recovery_time` オプションと一緒に使用します。変更を有効にするには、データベースサーバを停止し、再起動する必要があります。

## 関連情報

[チェックポイントログ \[294 ページ\]](#)

[データベースサーバがチェックポイントのタイミングを決定する方法 \[293 ページ\]](#)

[recovery\\_time オプション \[768 ページ\]](#)

[-gc データベースサーバオプション \[413 ページ\]](#)

[-m データベースサーバオプション \[447 ページ\]](#)

## 1.5.6.7.24 cis\_option オプション

リモートデータアクセス用のデバッグ情報をデータベースサーバメッセージウィンドウに表示するかどうかを制御します。

## 指定可能な値

0、7

## デフォルト

0

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、リモートデータアクセスを使用する場合に、リモートデータベースでのクエリの実行方法に関する情報をデータベースサーバメッセージウィンドウに表示するかどうかを制御します。このオプションを 7 に設定すると、データベースサーバメッセージウィンドウにデバッグ情報が表示されます。このオプションを 0 (デフォルト) に設定すると、データベースサーバメッセージウィンドウにリモートデータアクセスのデバッグ情報は表示されません。

リモートトレーシングを有効にすると、データベースサーバのメッセージウィンドウにトレーシング情報が表示されます。この出力をファイルに記録するには、データベースサーバの起動時に -o サーバオプションを指定します。

## 関連情報

[-o データベースサーバオプション \[452 ページ\]](#)

## 1.5.6.7.25 cis\_rowset\_size オプション

各フェッチに対してリモートサーバから返されるローの数を設定します。

### 指定可能な値

整数

### デフォルト

50

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、リモートサーバへの新しい接続が確立されたときに有効になります。

このオプションは、ODBC を使用してリモートデータベースサーバに接続する場合の ODBC FetchArraySize 値を設定します。

## 1.5.6.7.26 close\_on\_endtrans オプション

トランザクションの終了時にカーソルを閉じるかどうかを制御します。

## 指定可能な値

On、Off

## デフォルト

On

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('Off') によって設定されます。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり

	PUBLIC ロール	現在のユーザ	他のユーザ
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

close\_on\_endtrans が On に設定されている場合、カーソルが WITH HOLD で開かれていないかぎり、カーソルはトランザクションがコミットされるたびに閉じます。トランザクションがロールバックするときの動作は ansi\_close\_cursors\_on\_rollback オプションの設定内容によって制御されます。

close\_on\_endtrans が Off に設定されている場合は、ansi\_close\_cursors\_on\_rollback オプションの設定に関係なく、またカーソルが WITH HOLD で開かれているかどうかにも関係なく、コミットまたはロールバックのどちらでもカーソルは閉じません。

このオプションを Off に設定すると、Adaptive Server Enterprise と互換性のある動作になります。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[ansi\\_close\\_cursors\\_on\\_rollback オプション \[647 ページ\]](#)

## 1.5.6.7.27 collect\_statistics\_on\_dml\_updates オプション

INSERT、DELETE、UPDATE などのデータ変更 DML 文の実行中に統計を収集するかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

データベースサーバは通常の文の実行中に統計情報を更新し、収集した統計を使用してカラム統計の自己チューニングを行います。collect\_statistics\_on\_dml\_updates オプションを Off に設定すると、INSERT、DELETE、UPDATE などのデータ変更 DML 文の実行中における統計の更新が無効となります。

通常の状況ではこのオプションを Off にする必要はありません。ただし、大量のデータが頻繁に変更される環境では、update\_statistics を On に設定している場合、同時にこのオプションを Off にすることで、パフォーマンスが改善することがあります。

collect\_statistics\_on\_dml\_updates オプションと update\_statistics オプションが異なる点は、update\_statistics オプションを On に設定した場合、ある述部を満たすローの実際の数とその述部を満たすと予想されるローの数が比較されてから、推定値が更新されることです。collect\_statistics\_on\_dml\_updates オプションを On に設定した場合は、挿入、更新、または削除されたローの値に基づいてカラム統計が修正されます。

## 関連情報

[update\\_statistics オプション \[821 ページ\]](#)

## 1.5.6.7.28 compression オプション [SQL Remote]

SQL Remote メッセージの圧縮レベルを設定します。

### 指定可能な値

-1 ~ 9 の整数

## デフォルト

6

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

値には次の意味があります。

-1

メッセージをバージョン 5 フォーマットで送信します。バージョン 5 の Message Agent では、バージョン 6 以降の Message Agent で送信されたメッセージを読むことはできません。自分のシステムにあるすべての Message Agent がバージョン 6 以降にアップグレードされるまでは、必ず圧縮オプションを -1 に設定してください。

0

圧縮しません。

1 ~ 9

圧縮率を高めます。大きい圧縮率でメッセージを作成すると、小さい圧縮率を使用した場合より時間がかかります。

### 1.5.6.7.29 conn\_auditing オプション

auditing オプションが On に設定されている場合に、接続ごとに監査を有効または無効にします。

## 指定可能な値

On、Off

## デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	×	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい (現在の接続のみ)	×

## 備考

conn\_auditing オプションの設定は、ログインプロシージャに設定されている (login\_procedure データベースオプションで指定します) 場合のみ有効です。conn\_auditing を On に設定すると、その設定の監査がオンになります。ただし、auditing オプションも On に設定しないかぎり、監査情報は記録されません。接続が監査されているかどうかを確認するには、次のプロシージャを実行します。

```
SELECT CONNECTION_PROPERTY ( 'conn_auditing' );
```

## 関連情報

[auditing オプション \[656 ページ\]](#)

[login\\_procedure オプション \[720 ページ\]](#)

## 1.5.6.7.30 connection\_authentication オプション

認証アプリケーションのデータベースシグネチャに対する、アプリケーションシグネチャの確認に使用する認証文字列を指定します。

## 指定可能な値

文字列

## デフォルト

空の文字列

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	いいえ	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

## 備考

このオプションは、OEM 認証をサポートする SQL Anywhere データベースサーバの OEM Edition を使用している場合にのみ有効になります。

認証されたアプリケーションは、接続の確立後すぐに、すべての接続に対して connection\_authentication データベースオプションを設定する必要があります。シグネチャが検証された場合、接続が認証され、SQL の権限によって強制される制限を超えて、接続の動作の制限がなくなります。シグネチャが検証されないと、接続は、非認証アプリケーションによって許可されているアクションに制限されます。

connection\_authentication オプションは、現在の接続の間だけ、TEMPORARY キーワードを使用して設定してください。次の SQL 文は、接続を認証します。

```
SET TEMPORARY OPTION connection_authentication =  
  'company = company-name;  
  application = application-name;  
  signature = application-signature';
```

company-name と application-name は、database\_authentication オプションにより指定された値に一致する必要があります。application-signature は、SAP から取得したアプリケーションシグネチャです。

会社名に引用符やアポストロフィなどの特殊文字が含まれている場合は、文字列の中で特殊文字を 2 つ続けて記述します。

### 例

次の例では、特殊文字を含む認証文字列を指定します。

```
SET TEMPORARY OPTION connection_authentication=  
  'Company = Joe''s Garage;  
  Application = Joe''s Program;  
  Signature = 0fa55157edb8e14d818e...';
```

## 関連情報

[OEM Edition の認証アプリケーション \[354 ページ\]](#)

[database\\_authentication オプション \[682 ページ\]](#)

### 1.5.6.7.31 connection\_type データベースオプション

モニタリングアプリケーションからの接続がデータベースまたはデータベースサーバのシャットダウンに影響するかどうかを制御します。

#### 指定可能な値

Event、Internal、Standard、Monitor

デフォルトの値は次のとおりです。

##### Standard

CmdSeq、TDS、HTTP、および HTTPS 接続の場合。

##### Event

イベントハンドラ用に作成された接続の場合。

##### Internal

その他すべてのタイプの接続の場合。

#### デフォルト

Standard

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	×	×	×
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

CmdSeq、TDS、HTTP、および HTTPS 接続の場合、サポートされる値は **Standard** と **Monitor** です。

次のいずれかを使用する場合、**Monitor** に設定されている接続は、データベースまたはデータベースサーバをシャットダウンするための条件が満たされたかどうかを判別するときに無視されます。

- データベースサーバメッセージウィンドウのシャットダウンボタン。
- -y オプションを指定しない dbstop ユーティリティ。
- UNCONDITIONALLY 句のない STOP DATABASE 文または STOP SERVER 文。

データベースが自動で停止するように設定している場合、connection\_type が Monitor に設定されている接続は、データベースを自動で停止するための条件が満たされているかどうかを判別するときに無視されます。

### i 注記

セッション ID が関連付けられていない HTTP 接続および HTTPS 接続は、接続タイプが **Monitor** に設定されていない場合でも、データベースまたはデータベースサーバをシャットダウンするための条件が満たされているかどうかを判別するときに常に無視されます。

セッション ID が関連付けられている HTTP 接続および HTTPS 接続は、接続タイプが **Monitor** に設定されている場合、データベースまたはデータベースサーバをシャットダウンするための条件が満たされているかどうかを判別するときのみ無視されます。

### 例

データベースまたはデータベースサーバをシャットダウンするための条件が満たされているかどうかを判別するときに、データベースサーバで接続を無視するには、次の文を実行します。

```
SET TEMPORARY OPTION connection_type = 'Monitor';
```

## 1.5.6.7.32 continue\_after\_raiserror オプション

RAISERROR 文に応じて動作を制御します。

### 指定可能な値

On、Off

### デフォルト

On

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続によって設定されます ('On')。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

RAISERROR 文はプロシージャ内で使用され、エラー生成を実行します。このオプションを Off に設定すると、RAISERROR 文に到達するたびにプロシージャまたはトリガの実行が停止します。

このオプションは BEGIN...END 文の TRY ブロック内では無視されます。

continue\_after\_raiserror オプションを On に設定すると、RAISERROR 文は実行終了エラーの信号を送信しなくなります。代わりに、RAISERROR ステータスコードとメッセージが格納され、プロシージャが完了すると直前の RAISERROR が返されます。RAISERROR を引き起こしたプロシージャが他のプロシージャから呼び出された場合、最も外側のプロシージャが終了するまで RAISERROR は返されません。

中間レベルの RAISERROR のステータスとコードは、プロシージャが終了すると失われます。リターン時に RAISERROR と一緒にエラーが生じた場合は、新しいエラー情報が戻され、RAISERROR 情報は失われます。アプリケーションは異なる実行ポイントで @@error グローバル変数を検査して、中間 RAISERROR ステータスを問い合わせることができます。

continue\_after\_raiserror オプションの設定は RAISERROR 文の後の動作を制御することを目的とし、on\_tsq\_error オプションが Conditional (デフォルト) に設定されている場合のみ使用します。on\_tsq\_error オプションを Stop または Continue に設定した場合、その設定は continue\_after\_raiserror の設定より優先されます。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[on\\_tsq\\_error オプション \[747 ページ\]](#)

### 1.5.6.7.33 conversion\_error オプション

データベースからデータをフェッチするときの、データ型変換障害のレポートを制御します。

#### 指定可能な値

On、Off

#### デフォルト

On

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

#### 備考

このオプションは、データがデータベースからフェッチされる時、またはデータベースに挿入される時にデータ型変換障害が発生した場合、データベースがエラー (conversion\_error が On) と警告 (conversion\_error が Off) のどちらとしてレポートするかを制御します。

conversion\_error を On に設定すると、SQLE\_CONVERSION\_ERROR エラーまたは SQLE\_OVERFLOW\_ERROR エラーが生成されます。このオプションを Off に設定すると、警告 SQLE\_CANNOT\_CONVERT が生成されます。

変換エラーが警告のみでレポートされた場合、変換できなかった値の代わりに NULL 値が使用されます。Embedded SQL では、インジケータ変数はエラーを出したカラムに -2 を設定します。

#### 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.34 cooperative\_commit\_timeout オプション [廃止予定]

トランザクションログ内の COMMIT エントリがいつディスクに書き込まれるかを管理します。

### i 注記

バージョン 11 の時点で、コミット動作が自動的にチューニングされるためこのオプション設定は無視されます。

### 指定可能な値

整数 (ミリ秒単位)

### デフォルト

250

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションは、cooperative\_commits が On に設定されている場合にかぎり意味を持ちます。データベースサーバは、ディスクに書き込む前に、指定されたミリ秒数だけ、他の接続がログのページを埋めるのを待ちます。デフォルト設定は 250 ミリ秒です。

### 関連情報

[cooperative\\_commits オプション \[廃止予定\] \[681 ページ\]](#)

[delayed\\_commit\\_timeout オプション \[693 ページ\]](#)

## 1.5.6.7.35 cooperative\_commits オプション [廃止予定]

コミットがいつディスクに書き込まれるかを制御します。

### i 注記

バージョン 11 の時点で、コミット動作が自動的にチューニングされるためこのオプション設定は無視されます。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

cooperative\_commits を Off に設定した場合、COMMIT はデータベースサーバで受信されたときにディスクに書き込まれ、その後アプリケーションは継続が許可されます。

cooperative\_commits を On (デフォルト) に設定し、他にアクティブな接続がある場合は、データベースサーバは COMMIT をすぐにはディスクに書き込みません。アプリケーションは cooperative\_commit\_timeout オプションで設定した最大長になるまで、他にそのページに含めるものがないか待ってから、COMMIT をディスクに書き込みます。

cooperative\_commits を On に設定し、cooperative\_commit\_timeout の設定を大きくすると、ディスク I/O の数が減少することによってデータベースサーバの全体的なスループットが向上しますが、各接続のターンアラウンドタイムが長くなります。

cooperative\_commits と delayed\_commits の両方を On に設定している場合、cooperative\_commit\_timeout に設定された時間がページの書き込みなしで経過すると、アプリケーションが (コミットが実行されたかのように) 再開され、残りの時間

(delayed\_commit\_timeout の値から cooperative\_commit\_timeout の値を差し引いた長さ) は delayed\_commits 間隔として使用されます。その後、ページは満杯になっていなくてもディスクに書き込まれます。

## 関連情報

[delayed\\_commits オプション \[694 ページ\]](#)

### 1.5.6.7.36 database\_authentication オプション

データベースの認証文字列を設定します。

#### 指定可能な値

文字列

#### デフォルト

空の文字列

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	いいえ	いいえ

#### 備考

このオプションは、OEM 認証をサポートする SQL Anywhere データベースサーバの OEM Edition を使用している場合にのみ有効になります。

このオプションを有効にするには、データベースサーバを再起動する必要があります。

データベースが認証されると、正しい認証シグネチャを指定する接続だけが、データベースへの操作を実行できます。認証されない接続は、読み込み専用モードで操作します。認証されたデータベースを使用するには、SQL Anywhere OEM Edition を使用してください。

データベースを認証するには、データベースの `database_authentication` オプションを設定します。

```
SET OPTION PUBLIC.database_authentication =  
'company = company-name;  
application = application-name;  
signature = database-signature';
```

`company-name` と `application-name` の引数は、シグネチャの取得時に SAP に提供した値です。`database-signature` は、SAP から受け取ったデータベースシグネチャです。

会社名に引用符やアポストロフィなどの特殊文字が含まれている場合は、文字列の中で特殊文字を 2 つ続けて記述します。

データベースサーバが認証データベースをロードするとき、データベースサーバメッセージウィンドウに、認証された会社とアプリケーションについて説明するメッセージが表示されます。このメッセージを調べて、`database_authentication` オプションが有効になっているかどうかを確認できます。メッセージの形式は次のとおりです。

```
このデータベースは次の使用を目的としてライセンスされています。  
アプリケーション: application-name  
会社: company-name
```

SQL スクリプトファイルに認証の文を格納し、長いシグネチャを繰り返し入力することを回避できます。認証の文をファイル `%SQLANY17%\scripts\authenticate.sql` に格納すると、この文は、データベースの作成、再構築、または更新を行うたびに適用されます。

#### 例

```
SET OPTION PUBLIC.database_authentication =  
'company = MyCompany;  
application = MyApp;  
signature = 0fa55157edb8e14d818e';
```

## 関連情報

[OEM Edition の認証アプリケーション \[354 ページ\]](#)

[認証データベースのアップグレード \[362 ページ\]](#)

[connection\\_authentication オプション \[674 ページ\]](#)

## 1.5.6.7.37 date\_format オプション

データベースから取り出した日付の形式を設定します。

### 指定可能な値

文字列

### デフォルト

YYYY-MM-DD

現在のユーザのテンポラリ設定が ODBC ドライバ、JDBC ドライバ、OLE DB ドライバによって設定されます ('yyyy-mm-dd')。

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続によって設定されます ('YYYY-MM-DD')。

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

形式は次の記号を組み合わせた文字列です。

シンボル	説明
YY	2 桁の年
YYYY	4 桁の年
MM	2 桁の月
MMM[m...]	月を示す略式文字
D	曜日を示す 1 桁の数字 (1 = 日曜、7 = 土曜)
DD	2 桁の日

シンボル	説明
DDD[d...]	曜日を示す略式文字
JJJ	通し日数 (1 ~ 366)

各記号は、その形式を適用した日付のデータで置き換えられます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'mmm' は 3 文字の月名を示しています。

文字データを表す記号 (MMM など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、すべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、すべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字がデータベースサーバにより選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。

## i 注記

日付形式の順序が変更されるように date\_format の設定を変更する場合は、同じ変更が反映されるように date\_order オプションも変更し、同様に date\_order を変更する場合は date\_format も変更してください。

## 例

次の表は、2008 年 4 月 14 日 (月) に実行された次の文からの出力と、date\_format の設定を示します。

```
SELECT CAST( CURRENT DATE AS VARCHAR );
```

date_format	CURRENT DATE
yyyy/mm/dd/ddd	2008/04/14/mon
yyyy/Mm/Dd/ddd	2008/4/14/mon
jji	105
mmm yyyy	apr 2008
Mmm yyyy	Apr 2008
mm-yyyy	04-2008

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[date\\_order オプション \[686 ページ\]](#)

[time\\_format オプション \[808 ページ\]](#)

[timestamp\\_format オプション \[811 ページ\]](#)

[timestamp\\_with\\_time\\_zone\\_format オプション \[813 ページ\]](#)

## 1.5.6.7.38 date\_order オプション

日付形式の解釈を制御します。

### 指定可能な値

MDY、YMD、DMY

### デフォルト

YMD

現在のユーザのテンポラリ設定が ODBC ドライバ、JDBC ドライバ、OLE DB ドライバによって設定されます ('ymd')。

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続によって設定されます ('MDY')。

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

### 備考

年、月、日のデフォルトの順序は、ISO 8601 日付フォーマットに対応しています。たとえば、"06-01-07" は、2006 年 1 月 7 日として解釈されます。

日付部分の解釈のために、さまざまな順番を指定できます。たとえば、"06-01-07" を 2007 年 6 月 1 日として解釈させるには、日付の順序を "MDY" と指定します。

#### **i** 注記

date\_order オプションのデフォルト設定を変更する場合、その変更を反映するように date\_format オプションと timestamp\_format オプションも変更できます。

値によって、日付 10/11/12 をどう解釈するかが決まります。

日付順	意味
MDY	October 11 2012
YMD	November 12 2010
DMY	November 10 2012

nearest\_century オプションを使用して、文字列から日付への変換で、2 桁の年の解釈を制御します。

次の SQL クエリは、date\_order と nearest\_century のデフォルト設定を使用して、2010-11-12 を返します。

```
SELECT CAST( CAST( '10/11/12' AS DATE ) AS VARCHAR(15) );
```

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[date\\_format オプション \[684 ページ\]](#)

[timestamp\\_format オプション \[811 ページ\]](#)

### 1.5.6.7.39 db\_publisher オプション

データベースパブリッシャのユーザ ID を返します (存在する場合)。

#### 指定可能な値

-1、NULL、または任意のユーザ ID

#### デフォルト

-1

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	×	×	×

## 備考

このオプション設定はすぐに有効になります。

パブリッシャがすでに設定されている場合は、このオプションを -1 または NULL にリセットしてから別のユーザ ID を設定する必要があります。

パブリッシャを変更すると、CURRENT PUBLISHER 特別値の値が更新されます。

また、GRANT PUBLISH 文を使用して db\_publisher の値を設定したり、REVOKE PUBLISH 文を使用して db\_publisher の値をリセットしたりすることもできます。

## 1.5.6.7.40 debug\_messages オプション

DEBUG ONLY 句を含む MESSAGE 文を実行するか否かを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションにより、指定した DEBUG ONLY 句がある MESSAGE 文を含むストアードプロシージャとトリガ内のデバッグメッセージの動作を制御できます。このオプションはデフォルトで Off に設定されており、MESSAGE 文が実行された場合にデバッグメッセージは表示されません。debug\_messages を On に設定することにより、すべてのストアードプロシージャとトリガでデバッグメッセージが有効になります。

### i 注記

debug\_messages オプションを Off に設定している場合、DEBUG ONLY メッセージはパフォーマンスへの影響が小さいため、通常は運用システム上のストアードプロシージャに残すことができます。ただし、頻繁に実行される場所では控え目に使用してください。そうしないと、パフォーマンスが多少低下することがあります。

## 1.5.6.7.41 dedicated\_task オプション

要求処理タスクを、単一の接続からの要求処理に特化します。

### 指定可能な値

On、Off

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	いいえ	いいえ	不可。
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい (現在の接続のみ)、SET ANY SYSTEM OPTION あり	いいえ

## 備考

dedicated\_task 接続オプションを On に設定すると、要求処理タスクは、その接続の要求処理専用となります。このオプションを有効にして接続を事前に確立しておくことで、データベースサーバが応答しなくなった場合にデータベースサーバの状態に関する情報を収集できます。

## 例

```
DROP USER admin;
CREATE USER ADMIN IDENTIFIED BY sql;
GRANT SET ANY SYSTEM OPTION TO admin;
CREATE OR REPLACE PROCEDURE dba.admin_login_procedure ()
BEGIN
    SET TEMPORARY OPTION dedicated_task='on';
END;
GRANT EXECUTE ON dba.admin_login_procedure TO ADMIN;
SET OPTION admin.login_procedure = 'dba.admin_login_procedure';
```

## 1.5.6.7.42 default\_dbspace オプション

テーブルが作成されるデフォルト DB 領域を変更します。

### 指定可能な値

文字列

### デフォルト

空の文字列

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

各データベースについて、システム (メイン) DB 領域に加えて最大 12 の DB 領域を作成できます。DB 領域を指定しないでテーブルを作成すると、このオプションに指定された DB 領域が使用されます。このオプションを設定しない場合、空の文字列に設定した場合、または system に設定した場合は、システム DB 領域が使用されます。

テンポラリテーブルまたはインデックスは、default\_dbspace オプションの設定に関係なく常に TEMPORARY DB 領域に配置されます。ベーステーブルの作成時に IN 句を指定した場合は、default\_dbspace オプションで指定する DB 領域ではなく、IN 句で指定した DB 領域が使用されます。

すべてのテーブルをシステム DB 領域以外のロケーションに作成している場合は、システム DB 領域はチェックポイントログとシステムテーブルの保存にのみ使用されます。この設定は、パフォーマンス上の理由からチェックポイントログを他のデータベースオブジェクトとは別のディスクに保存する場合に便利です。チェックポイントログを別のディスクに保存するには、すべての CREATE TABLE 文で DB 領域を指定するか、このオプションの設定を変更してからテーブルを作成します。

### 例

次の例では、まず MyLibrary という名前の新しい DB 領域を作成します。次に、その DB 領域をデフォルト DB 領域に設定し、テーブル LibraryBooks をシステム DB 領域ではなく MyLibrary DB 領域に保存します。

```
CREATE DBSPACE MyLibrary
AS 'c:\¥¥dbfiles¥¥library.db';
SET OPTION default_dbspace = 'MyLibrary';
CREATE TABLE LibraryBooks (
    title CHAR(100),
    author CHAR(50),
    isbn CHAR(30),
);
```

## 関連情報

[追加の DB 領域の考慮事項 \[270 ページ\]](#)

[DB 領域の作成 \[272 ページ\]](#)

[-ds データベースオプション \[525 ページ\]](#)

## 1.5.6.7.43 default\_timestamp\_increment オプション

TIMESTAMP または UTC TIMESTAMP がデフォルト値になっているカラムを含むローが挿入または更新されるときに、このカラムの値を一意にしておくためにカラムに追加するマイクロ秒数を指定します。

### 指定可能な値

整数 (1 ~ 1000000)

### デフォルト

1

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

SQL Anywhere では、TIMESTAMP 値の精度が小数第 6 位であるため、2 つの同一の TIMESTAMP 値または TIMESTAMP WITH TIME ZONE 値を区別するために、デフォルトでは 1 マイクロ秒 (0.000001 秒) が追加されます。

このデータベースオプションの設定は、DEFAULT CURRENT TIMESTAMP 属性で宣言された TIMESTAMP カラムには適用されません。

DEFAULT TIMESTAMP 属性または DEFAULT UTC TIMESTAMP 属性を持つカラムを含む複数の行が更新され、カラム自体は明示的に更新されない場合は、更新される各ローの一意の値がカラムに与えられます。たとえば、DEFAULT UTC TIMESTAMP 属性を持つ TIMESTAMP WITH TIME ZONE タイプのカラムを含む 3 つのローが更新されるとします。これらのローは '2011-04-01 12:47:52.724000+00:00' に更新されますが、ローごとに一意の UTC TIMESTAMP 値が追加されます。

```
2011-04-01 12:47:52.724000+00:00
2011-04-01 12:47:52.724001+00:00
2011-04-01 12:47:52.724002+00:00
```

Microsoft Access などのソフトウェアの中には、TIMESTAMP 値を小数点第 3 位までトランケートしてしまうため、正しい比較を行う上で問題になるものもあります。互換性を確保するために、truncate\_timestamp\_values オプションを On に設定し、データベースサーバが格納する小数点以下の桁数を指定できます。

Mobile Link 同期でこのオプションを設定しようとする場合は、1 回目の同期を実行する前に設定してください。

## 関連情報

[truncate\\_timestamp\\_values オプション \[815 ページ\]](#)

### 1.5.6.7.44 delayed\_commit\_timeout オプション

delayed\_commits オプションが On のときに、アプリケーションが COMMIT を実行する時間から、COMMIT が実際にディスクに書き込まれる時間までの最大遅延を指定します。

## 指定可能な値

整数 (ミリ秒単位)

## デフォルト

500

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、delayed\_commits が On に設定されている場合にかぎり意味を持ちます。トランザクションログ内の COMMIT エントリがいつディスクに書き込まれるかを管理します。delayed\_commits を On に設定すると、データベースサ

サーバは、`delayed_commit_timeout` オプションで指定されたミリ秒間待ち、他の接続によって1つのログページを埋めてから、現在のページ内容をディスクに書き込みます。

## 関連情報

[delayed\\_commits オプション \[694 ページ\]](#)

[cooperative\\_commit\\_timeout オプション \[廃止予定\] \[680 ページ\]](#)

[cooperative\\_commits オプション \[廃止予定\] \[681 ページ\]](#)

### 1.5.6.7.45 `delayed_commits` オプション

COMMIT の後でデータベースサーバからアプリケーションに制御が戻されるタイミングを決定します。

#### 指定可能な値

On、Off

#### デフォルト

Off

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

#### 備考

Off は、ISO の COMMIT 動作に準拠しています。

On に設定すると、データベースサーバは COMMIT のトランザクションログエントリがディスクに書き込まれるのを待たずに、直ちに COMMIT 文に応答します。Off に設定すると、アプリケーションは COMMIT がディスクに書き込まれるまで待たなければなりません。

このオプションが On の場合、ログページが満杯となったとき、delayed\_commit\_timeout オプションに設定された時間が経過したときのうち、いずれか早い時点でログがディスクに書き込まれます。トランザクションがコミットされた後であっても、データベースサーバが COMMIT に応答してからページがディスクに書き込まれる前にシステム障害が発生すると、トランザクションが失われる可能性がわずかながらあります。delayed\_commits を On に設定し、delayed\_commit\_timeout オプションを高い値にすると、応答時間は早くなりますが、リカバリ中にコミットされたトランザクションが失われるリスクが若干高くなります。

cooperative\_commits と delayed\_commits の両方を On に設定した場合、ページが書き込まれないまま cooperative\_commit\_timeout 時間が経過すると、アプリケーションが (コミットが機能していたかのように) 再開され、残りの時間 (delayed\_commit\_timeout - cooperative\_commit\_timeout) は delayed\_commits 時間として使用されます。その後、ページがいっぱいでもページが書き込まれます。

## 関連情報

[cooperative\\_commit\\_timeout オプション \[廃止予定\] \[680 ページ\]](#)

[cooperative\\_commits オプション \[廃止予定\] \[681 ページ\]](#)

[delayed\\_commit\\_timeout オプション \[693 ページ\]](#)

## 1.5.6.7.46 delete\_old\_logs オプション [Mobile Link][SQL Remote]

トランザクションがレプリケートまたは同期されたときにトランザクションログを削除するかどうかを制御します。

### 指定可能な値

On、Off、Delay、n days (日数)

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

SQL Anywhere Mobile Link クライアントと SQL Remote がこのオプションを使用します。デフォルト設定は Off です。このオプションを On に設定すると、古いトランザクションログ内のすべての変更についてレプリケーションまたは同期が完了した時点で、古いトランザクションログが削除されます。Delay に設定すると、トランザクションログの最終変更のタイムスタンプで作成日が現在の日付であることが示されている場合、すべての変更が送信され、受信が確認されても、そのトランザクションログは削除されません。n days に設定すると、n 日より古いログが削除されます。無効な値は Off として解釈され、エラーや警告は報告されません。

### 例

delete\_old\_logs オプションが 10 日に設定されているリモートデータベースに対して、1月 18 日に dbmlsync を実行すると、dbmlsync は、1月 7 日以前に作成されたオフライントランザクションログを削除します。リモートデータベースでは、このオプションを次のように設定します。

```
SET OPTION delete_old_logs = '10 days';
```

## 関連情報

[トランザクションログ \[279 ページ\]](#)

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

## 1.5.6.7.47 disk\_sandbox オプション

データベースファイルの読み込み/書き込み操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されるかどうかを制御します。

## 指定可能な値

On、Off

## デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×

## 備考

このオプションを On に設定すると、データベースファイルの読み込み/書き込み操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されます。設定はセッション間で維持されます。

## 関連情報

[ディスクサンドボックス \[1604 ページ\]](#)

[-sbx データベースサーバオプション \[470 ページ\]](#)

[-sbx データベースオプション \[534 ページ\]](#)

## 1.5.6.7.48 divide\_by\_zero\_error オプション

ゼロ除算のレポートを制御します。

## 指定可能な値

On、Off

## デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、ゼロ除算をエラーとしてレポートするかどうかを指示します。このオプションを On に設定すると、ゼロ除算は SQLSTATE '22012' のエラーになります。

このオプションを Off に設定すると、ゼロ除算はエラーになりません。代わりに、NULL が返されます。

## 1.5.6.7.49 escape\_character オプション (システムで使用するために予約済み)

システムで使用するために予約されています。

## デフォルト

On

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('Off') によって設定されます。

## 備考

このオプションの設定は変更しないでください。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

### 1.5.6.7.50 `exclude_operators` オプション (システムで使用するために予約済み)

システムで使用するために予約されています。このオプションの設定は変更しないでください。

### 1.5.6.7.51 `extended_join_syntax` オプション

複数テーブルのジョインに対して、重複する相関名構文を持つクエリを許可するか、またはエラーとしてレポートするかを指定します。

#### 指定可能な値

On、Off

#### デフォルト

On

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションを On に設定すると、データベースサーバで外部ジョインの NULL 入力側で重複した相関名を使用することが可能となります。同じ相関名で指定されたすべてのテーブルまたはビューは、テーブルまたはビューの同じインスタンスとして解釈されます。

次の FROM 句は、重複する相関名を使用したジョインが SQL Anywhere でどのように解釈されるかを示しています。C1 と C2 は探索条件を表します。

```
( R left outer join T on ( C1 ), T join S on ( C2 ) )
```

このオプションを On に設定すると、このジョインは次のように解釈されます。

```
( R left outer join T on ( C1 ) ) join S on ( C2 )
```

このオプションを Off に設定すると、SQLCODE -137 が生成されます。

### i 注記

重複する相関名を削除した結果を確認する場合、2 番目の引数を ANSI に設定した REWRITE 関数を使うと、書き直した文を表示できます。

## 1.5.6.7.52 extern\_login\_credentials オプション

リモート接続を行うときに、セッションユーザの外部ログインクレデンシャルを使用するか、既存の (有効な) ユーザの外部ログインクレデンシャルを使用するかどうかを制御します。このオプションは下位互換性のために提供されています。セキュリティ上の理由から、このオプションは指定しないでください。

### 指定可能な値

Login\_user、Effective\_user

### デフォルト

Effective\_user

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×

## 備考

このオプションは、リモートデータアクセス接続が実行されるときに、セッションユーザの外部ログインクレデンシャルを使用するか、現在実行しているユーザの外部ログインクレデンシャルを使用するかかどうかを指定します。セッションユーザは接続しているユーザで、現在実行しているユーザはスタアドプロシージャコールによって異なる場合があります。

### Login\_user

Login\_user を指定すると、現在実行しているユーザが何かに関係なく、リモートデータアクセス接続が実行されるときに、セッションユーザの外部ログインクレデンシャルがデータベースサーバで常に使用されます。

### Effective\_user

Effective\_user を指定すると、現在実行しているユーザの設定がデータベースサーバで優先されます。リモート接続は、現在実行しているユーザの外部ログインクレデンシャルに基づいて行われます。Effective\_user を指定した場合、同じローカル接続に対して複数の（現在実行しているユーザごとに 1 つずつ）リモート接続が行われることがあります。

## 1.5.6.7.53 external\_remote\_options オプション [SQL Remote]

メッセージリンクパラメータを格納するかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

SQL Remote はこのオプションを使用して、メッセージリンクパラメータをデータベースに格納するのか (Off) 外部に格納するのか (On) を示します。

## 1.5.6.7.54 fire\_triggers オプション

データベースにおけるトリガ起動を制御します。

## 指定可能な値

On、Off

## デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい (現在の接続のみ)、SET ANY SYSTEM OPTION あり	×

## 備考

On に設定すると、トリガが起動されます。Off に設定すると、参照整合性トリガ (カスケード更新や削除など) を含め、トリガは起動されません。-gf オプションはこのオプションよりも優先され、fire\_triggers の設定に関係なくすべてのトリガの起動がオフになります。

Adaptive Server Enterprise トランザクションログのアクションは、トリガによって実行されるアクションも含め、すべて SQL Anywhere にレプリケートされるので、このオプションは、Adaptive Server Enterprise から SQL Anywhere にデータをレプリケートする場合に意味を持ちます。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[-gf データベースサーバオプション \[417 ページ\]](#)

## 1.5.6.7.55 first\_day\_of\_week オプション

何曜日を週の最初にするかを設定します。

### 指定可能な値

1、2、3、4、5、6、7

### デフォルト

7

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

値には次の意味があります。

値	意味
1	月曜日
2	火曜日
3	水曜日
4	木曜日
5	金曜日
6	土曜日
7	日曜日

デフォルトは 7 で、1 週間が日曜日から始まることを意味します。

このオプションで指定する値は、平日の値を取得するときに DATEPART 関数の結果に影響します。週の最初の曜日は、T-SQL の SET 文で DATEFIRST オプションを使用して変更することもできます。

このオプションで指定する値は、DOW 関数の結果には影響しません。たとえば、週の最初の曜日を月曜日に設定しても、DOW 関数は月曜日の値として 2 を返します。

## 1.5.6.7.56 for\_xml\_null\_treatment オプション

FOR XML 句を使用するクエリでの NULL 値の処理を制御します。

### 指定可能な値

Empty、Omit

### デフォルト

Omit

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

FOR XML 句を含むクエリを実行する場合、for\_xml\_null\_treatment オプションが NULL 値の処理方法を決定します。デフォルトでは、NULL 値を含む要素と属性は、結果から省略されます。このオプションを Empty に設定すると、値が NULL の場合に空の要素または属性が生成されます。

### 1.5.6.7.57 force\_view\_creation オプション (システムで使用するために予約済み)

システムで使用するために予約されています。このオプションの設定は変更しないでください。

#### 警告

force\_view\_creation オプションは、reload.sql スクリプトでのみ使用されます。このオプションは、アンロードユーティリティ (dbunload) でのみ使用され、明示的に設定されません。

### 1.5.6.7.58 global\_database\_id オプション

DEFAULT GLOBAL AUTOINCREMENT が指定されたカラムの値の範囲を設定します。

レプリケーション環境でユニークなプライマリキーを生成する場合に使用します。

## 指定可能な値

負でない整数

## デフォルト

2147483647

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

## 備考

このオプションで指定した値は開始値となります。DEFAULT GLOBAL AUTOINCREMENT が指定されたカラムでは、DEFAULT GLOBAL AUTOINCREMENT カラムに値がないテーブルにローを挿入すると、データベースサーバによってそのカラムの値が自動的に生成されます。この値は、global\_database\_id の値とカラムの分割サイズによって決まります。

global\_database\_id をデフォルト値に設定すると、DEFAULT GLOBAL AUTOINCREMENT が無効になります。この場合、デフォルトとして NULL が生成されます。

現在のデータベースのオプション値を検索するには、次の文を使用します。

```
SELECT DB_PROPERTY( 'GlobalDBID' );
```

この機能は、特に、レプリケーション環境でユニークなプライマリキーを生成するために使用します。

### 例

次の例は、データベース ID 番号を 100 に設定します。

```
SET OPTION PUBLIC.global_database_id = '100';
```

## 1.5.6.7.59 http\_connection\_pool\_basesize オプション

データベース接続の公称スレッショルドサイズを指定します。

### 指定可能な値

整数 (0 ~ 1000)

### デフォルト

10

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

### 備考

このオプションの変更は、次のページ循環時に 5 秒以内に有効になります。http\_connection\_pool\_basesize スレッショルド内の http\_connection\_pool\_timeout を超過した未使用のデータベース接続は削除されます。接続プールのサイズが http\_connection\_pool\_basesize スレッショルドを超えると、より積極的な速さで接続が削除されます。

値 0 を指定すると、オプションを設定してから 5 秒以内に HTTP 接続プールがパージされ、HTTP 接続プールが無効になります。

未使用の余分な接続はタイムアウト間隔の半分以内に削除され、残りの接続のうち半分は、タイムアウト間隔の 4 分の 1 以内に使用されない場合は削除される、ようになります。データベース内の接続プールの運用効率を判断するために、次のデータベースプロパティが追加されています。

- HttpConnPoolCachedCount
- HttpConnPoolHits
- HttpConnPoolMisses
- HttpConnPoolSteals

## 関連情報

[http\\_connection\\_pool\\_timeout オプション \[708 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

### 1.5.6.7.60 http\_connection\_pool\_timeout オプション

未使用の接続を接続プールに保持できる時間の上限を指定します。

#### 指定可能な値

整数 (1 ~ 86400)

#### デフォルト

60

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

#### 備考

プールサイズが `http_connection_pool_basesize` を超えると、この値に基づいた増加率で、余分な接続が削除されます。

このオプションの変更は、次のページ循環時に 5 秒以内に有効になります。`http_connection_pool_basesize` スレッシュホールド内の `http_connection_pool_timeout` を超過した未使用のデータベース接続は削除されます。接続プールのサイズが `http_connection_pool_basesize` スレッシュホールドを超えると、より積極的な速さで接続が削除されます。

未使用の余分な接続はタイムアウト間隔の半分以上以内に削除され、残りの接続のうち半分は、タイムアウト間隔の 4 分の 1 以内に使用されない場合は削除されるなどがあります。データベース内の接続プールの運用効率を判別するために、次のデータベースプロパティが追加されています。

- HttpConnPoolCachedCount
- HttpConnPoolHits
- HttpConnPoolMisses
- HttpConnPoolSteals

## 関連情報

[http\\_connection\\_pool\\_basesize オプション \[707 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

### 1.5.6.7.61 http\_session\_timeout オプション

非アクティブ状態の HTTP セッションが維持されるデフォルトのタイムアウト時間 (分単位) を指定します。

#### 指定可能な値

整数 (1 ~ 525600)

#### デフォルト

30

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このデータベースオプションを別の値に設定すると、以降のすべての HTTP セッションに適用される新しいデフォルトのセッションタイムアウトが設定されます。sa\_set\_http\_option システムプロシージャを呼び出して特定の HTTP セッション内で SessionTimeout を設定すると、このデフォルトを HTTP セッションで上書きできます。

## 1.5.6.7.62 integrated\_server\_name オプション

統合化ユーザ認証 (ログイン) 用の Windows ユーザグループメンバーシップの検索に使用するドメインコントローラサーバの名前を指定します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×

## 備考

このオプションを使用すると、SET ANY SECURITY OPTION システム権限を持つユーザは、統合化ログインに Windows ユーザグループを使う場合に、グループメンバーシップの検索に使用するドメインコントローラサーバの名前を指定できます。デフォルトでは、データベースサーバを実行しているコンピュータがグループメンバーシップの確認に使用されます。

## 例

次の例では、グループメンバーシップがコンピュータ server-1 で検証されることを指定します。

```
SET OPTION PUBLIC.integrated_server_name = '¥¥¥¥server-1';
```

## 関連情報

[Windows ユーザグループの統合化ログイン \[130 ページ\]](#)

## 1.5.6.7.63 isolation\_level オプション

ロック独立性レベルを制御します。

### 指定可能な値

0、1、2、3、Snapshot、Statement-snapshot、Readonly-statement-snapshot

### デフォルト

0

現在のユーザのテンポラリ設定が JDBC ドライバ (0) によって設定されます。

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 (1) によって設定されます。

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、次のようにロック独立性レベルを制御します。

0

ダーティリード、繰り返し不可能読み出し、幻ローを許可します。

1

ダーティリードを防ぎます。繰り返し不可能読み出しと幻ローを許可します。

2

ダーティリードと繰り返し不可能読み出しを防ぎます。幻ローを許可します。

3

直列化可能。ダーティリード、繰り返し不可能読み出し、幻ローを防ぎます。

### Snapshot

トランザクションが最初のローの読み込みまたは更新を行った時点から、コミットされたデータのスナップショットを使用します。

### Statement-snapshot

個々の文について、データベースから最初のローが読み込まれた時点から、コミットされたデータのスナップショットを使用します。繰り返し不可能読み出しと幻ローは、1つのトランザクション内で発生することはありますが、1つの文の中で発生することはありません。

### Readonly-statement-snapshot

読み込み専用の文についてのみ、データベースから最初のローが読み込まれた時点から、コミットされたデータのスナップショットを使用します。繰り返し不可能読み出しと幻ローは、1つのトランザクション内で発生することはありますが、1つの文の中で発生することはありません。更新可能な文については、`updatable_statement_isolation` オプションに指定された独立性レベル (0 (デフォルト)、1、2、3 のいずれか) を使用します。

このオプションを Snapshot、Statement-snapshot、または Readonly-statement-snapshot に設定する場合は、`allow_snapshot_isolation` オプションを On に設定してください。

Snapshot、Statement-snapshot、Readonly-statement-snapshot のいずれかの独立性レベルでクエリを実行した場合には、データベースのコミットされた状態のスナップショットからデータが取得されます。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこのオプションに対するテンポラリ設定や PUBLIC 設定よりも優先させることができます。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[allow\\_snapshot\\_isolation オプション \[643 ページ\]](#)

[updatable\\_statement\\_isolation オプション \[820 ページ\]](#)

## 1.5.6.7.64 java\_class\_path option

クラスの検索先となる一連の追加ディレクトリまたは JAR ファイルを指定します。

### 指定可能な値

文字列

デフォルト

空の文字列

スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

### 備考

データベース内の Java で使用するすべての Java クラスと JAR ファイルはデータベース内にインストールする必要があります。データベースを別のコンピュータまたはオペレーティングシステムに移動すると、Java クラスと JAR ファイルも同時に移動されます。また、クラスと JAR ファイルをデータベース内にインストールすると、データベースサーバのクラスローダでクラスとリソースをデータベースからフェッチできるという利点もあります。その結果、データベース内の Java を使用している各接続で、これらのクラスの独自のインスタンスと、クラス内の静的変数の独自のコピーを使用できます。

ただし、クラスまたは JAR ファイルをシステムクラスローダによってロードする必要がある場合は、java\_class\_path データベースオプションまたは -cp dbeng17/dbsrv17 サーバオプションで指定できます。どちらのオプションを指定しても、Java VM の起動用にデータベースサーバによって構築されるクラスパスにクラスと JAR ファイルが追加されます。java\_class\_path データベースオプションは、サーバが複数のデータベースを実行していて、それぞれのデータベースにシステムクラスローダでロードする必要のある異なるセットの JAR とディレクトリがあるときに便利です。-cp データベースサーバオプションは、サーバ上のすべてのデータベースが同一のクラスまたは JAR ファイルを必要とするときに便利です。

## 関連情報

[-cp データベースサーバオプション \[396 ページ\]](#)

### 1.5.6.7.65 java\_location オプション

データベースの Java VM のパスを指定します。

#### 指定可能な値

文字列

#### デフォルト

空の文字列

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

#### 備考

デフォルトでは、このオプションには空の文字列が設定されます。この場合、データベースサーバは、JAVA\_HOME 環境変数と Java VM のパスおよびその他のロケーションを検索します。JavaVM データベースプロパティを使用すると、java\_location オプションが設定されていない場合にデータベースサーバがどの Java VM を使用するのかをクエリできます。

## 関連情報

[java\\_vm\\_options オプション \[715 ページ\]](#)

## 1.5.6.7.66 java\_main\_userid オプション (廃止予定)

このオプションは推奨されなくなりました。

## 1.5.6.7.67 java\_vm\_options オプション

Java VM の起動時にデータベースサーバが使用するコマンドラインオプションを指定します。

### 指定可能な値

文字列

デフォルト

空の文字列

スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

### 備考

このオプションを使用すると、java\_location オプションで指定された Java VM を起動するときにデータベースサーバが使用するオプションを指定できます。これらの追加のオプションを使用して、デバッグ目的に Java VM を設定したり、UNIX プラットフォーム上のサービスとして実行したりできます。32 ビットモードではなく 64 ビットモードで Java VM を使用するために、その他のオプションが必要になることもあります。

## 例

次の例は、データベースサーバがサービスとして起動され、ユーザがログアウトしなければならない場合に、`java_vm_options` オプションを使用して UNIX での Java VM の実行を維持します。

```
SET OPTION PUBLIC.java_vm_options = '-Xrs';
```

次の例では、Java VM に対して HP-UX 上で 64 ビットモードで実行するように指示します。

```
SET OPTION PUBLIC.java_vm_options = '-d64';
```

## 関連情報

[java\\_location オプション \[714 ページ\]](#)

## 1.5.6.7.68 log\_deadlocks オプション

デッドロックレポートのオン/オフを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

## 備考

このオプションを On に設定した場合、データベースサーバは、内部バッファ内のデッドロックに関する情報をログに記録します。バッファのサイズは、10000 バイトに固定されています。デッドロック情報は sa\_report\_deadlocks ストアドプロシージャを使用すると表示できます。このオプションを Off に設定すると、バッファの内容は保持されません。

デッドロックが発生すると、そのデッドロックに関わった接続のみの情報がレポートされます。接続のレポート順序は、どの接続がどの行を待っているかに基づきます。スレッドデッドロックの場合、すべての接続の情報がレポートされます。

デッドロックレポートがオンになっていると、デッドロックが発生した場合に Deadlock システムイベントを使用してアクションを実行することもできます。

sa\_server\_option システムプロシージャで DeadlockLogging プロパティを使用して、このオプションの設定を変更することもできます。

## 関連情報

[システムイベント \[954 ページ\]](#)

[チュートリアル: 接続のブロックとデッドロックの診断 \(プロファイラ の場合\) \[1424 ページ\]](#)

## 1.5.6.7.69 login\_mode オプション

データベースに対する標準、統合化、Kerberos、LDAP、PAM の各ユーザ認証の使用を制御します。

### 指定可能な値

次のうち 1 つまたは複数指定できます。Standard、Integrated、Kerberos、LDAPUA、PAMUA、CloudAdmin、Mixed (廃止予定)

### デフォルト

Standard

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	いいえ	いいえ

## 備考

このオプションは、Standard、Integrated、Kerberos、LDAP、PAM、CloudAdmin のうちのユーザ認証を許可するかを指定します。以下のログインモードのうち 1 つまたは複数指定できます (大文字と小文字は区別されません)。

### i 注記

Integrated、Kerberos、LDAPUA、PAMUA は、永続的な login\_mode として指定しないでください。これを設定すると、ユーザがデータベースのコピーを取得した場合にデータベースに不正にアクセスできるようになります。

#### Standard

標準ユーザ認証を許可します。この値はデフォルトの設定です。標準ユーザ認証を使用する接続にはユーザ ID とパスワードの両方が含まれ、統合化または Kerberos の接続パラメータは使用されません。

#### Integrated

統合化ユーザ認証を許可します。

#### Kerberos

Kerberos ユーザ認証を許可します。

#### LDAPUA

LDAP (Lightweight Directory Access Protocol) ユーザ認証ログインを許可します。LDAP ユーザ認証を使用する接続にはユーザ ID とパスワードの両方が含まれ、統合化または Kerberos の接続パラメータは使用されません。

ldap\_failover\_to\_std オプションが ON に設定されている場合、ユーザのハッシュパスワードが変更されると、LDAPUA を使用してユーザがデータベースにログインしたときにデータベースの SYSUSER テーブルが更新されます。

LDAPUA を使用している場合、ログインポリシーのパスワード制御ルールは無視されます。

**PAMUA** PAM (Pluggable Authentication Modules) ユーザ認証ログインを許可します。pam\_failover\_to\_std オプションが ON に設定されている場合、ユーザのハッシュパスワードが変更されると、PAMUA を使用してユーザがデータベースにログインしたときにデータベースの SYSUSER テーブルが更新されます。

PAMUA を使用している場合、ログインポリシーのパスワード制御ルールは無視されます。

#### CloudAdmin

このログインモードは、クラウドの内部で使用します。

#### Mixed (廃止予定)

この値は、Standard、Integrated を指定した場合と同じです。

複数のログインモードを指定すると、データベースサーバでは、そのすべてのモードが使用可能となります。

### 警告

標準ユーザ認証を許可しないように login\_mode データベースオプションを設定すると、接続が、統合化、Kerberos、LDAP、または PAM のログインマッピングを付与されているユーザのみに制限されます。標準のデータベースユーザ ID とパスワードを使用して接続しようとすると、エラーが発生します。ただし、MANAGE ANY USER 権限を持つユーザは、このルールの唯一の例外です。

複数の値はカンマ区切りリストで指定します。このリストに空白スペースを含めることはできません。たとえば、次の設定は標準ログインと統合化ログインの両方を許可します。

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

### 警告

データベースファイルがセキュリティで保護されておらず、権限のないユーザによってコピーできる場合は、一時的な PUBLIC login\_mode オプションを使用してください (統合化、Kerberos、LDAP、および PAM のユーザ認証の場合)。ファイルのコピーする場合、統合化、Kerberos、LDAPUA、および PAMUA のログインはデフォルトではサポートされません。

### 例

統合化ユーザ認証のみを有効にします (標準、Kerberos、LDAP、および PAM のユーザ認証は失敗します)。

```
SET OPTION PUBLIC.login_mode = 'Integrated';
```

標準ユーザ認証と Kerberos ユーザ認証を有効にします (統合化、LDAP、および PAM のユーザ認証は失敗します)。

```
SET OPTION PUBLIC.login_mode = 'Standard,Kerberos';
```

標準、統合化、Kerberos、LDAP、および PAM の各ユーザ認証を有効にします。

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated,Kerberos,LDAPUA,PAMUA';
```

## 関連情報

[Windows 統合化ログイン \[124 ページ\]](#)

[Kerberos ユーザ認証 \[149 ページ\]](#)

[セキュリティ: ユーザ ID \[1601 ページ\]](#)

[セキュリティ: ログインモードを使用したデータベースのセキュリティ保護 \[162 ページ\]](#)

[-al データベースサーバオプション \[384 ページ\]](#)

## 1.5.6.7.70 login\_procedure オプション

ユーザがデータベースログインまたは Web サービスから接続するときに呼び出されるストアプロシージャを指定します。

### 指定可能な値

文字列

### デフォルト

sp\_login\_environment

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい (現在の接続のみ)、SET ANY SECURITY OPTION あり	×

### 備考

デフォルトでは sp\_login\_environment ストアドプロシージャが呼び出されます。このログインプロシージャは、すべてのチェックが完了し、接続が有効であることが確認された後で呼び出されます。login\_procedure オプションで指定されたプロシージャは、イベント接続では実行されませんが、通常のログインおよび Web サービス接続では実行されます。

カスタムログインプロシージャを作成するには、新しいストアプロシージャを作成し、login\_procedure をこの新しいプロシージャの所有者と名称に設定します。

このカスタムプロシージャは、TDS 接続が発生し、該当するデータベースオプションを設定するタイミングで dbo.sp\_tsq\_environment ストアドプロシージャを使用して検出するため、実行時の一部のポイントで dbo.sp\_login\_environment ストアドプロシージャを呼び出す必要があります。この操作が失敗した場合、TDS ベースの接続は切断されることがあります。dbo.sp\_login\_environment または dbo.sp\_tsq\_environment ストアドプロシージャのどちらかを修正するのはよいことはありません。

ユーザ定義のログインプロシージャから SQLSTATE 08WA0 のパスワード有効期限切れエラーメッセージを発生させ、パスワードの有効期限が切れていることをユーザに通知することができます。エラーの通知により、アプリケーションはエラーを確認し、有効期限の切れたパスワードを処理できます。ただし、パスワードの有効期限を実装する場合はログインポリシーを使用し、パスワードの有効期限切れのメッセージを返すログインプロシージャは使用しないでください。

NewPassword=\* 接続パラメータを使用する場合は、このエラーを通知して、クライアントライブラリが新しいパスワードの入力を要求するプロンプトを表示できるようにする必要があります。プロシージャで SQLSTATE '28000' (無効なユーザ ID またはパスワード) または SQLSTATE '08WAO' (パスワードの有効期限切れ) が通知されるか、RAISERROR のエラーが発生すると、ログインは失敗し、エラーがユーザに返されます。その他のエラーを通知するか、別のエラーが発生した場合は、ユーザログインは成功し、メッセージがデータベースサーバメッセージログに書き込まれます。

## 例

次の例は、データベース接続の合計数を制限する方法を示しています。

```
CREATE PROCEDURE DBA.login_check( )
BEGIN
    DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';
    // Allow a maximum of 3 concurrent connections
    IF( DB_PROPERTY( 'ConnCount' ) > 3 ) THEN
        SIGNAL INVALID_LOGON;
    ELSE
        CALL dbo.sp_login_environment;
    END IF;
END
go
GRANT EXECUTE ON DBA.login_check TO PUBLIC
go
SET OPTION PUBLIC.login_procedure='DBA.login_check'
go
```

次の例は、ユーザの失敗した接続の数が 30 分間で 3 回よりも多くなった場合に、接続試行をブロックする方法を示しています。ブロック期間中にブロックされた試行は、すべて無効パスワードエラーを受け取り、ログに失敗として記録されます。DBA がログを解析するために、ログは十分な時間保持されます。

```
CREATE TABLE DBA.ConnectionFailure(
    pk INT PRIMARY KEY DEFAULT AUTOINCREMENT,
    user_name CHAR(128) NOT NULL,
    tm TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
)
go
CREATE INDEX ConnFailTime ON DBA.ConnectionFailure(
    user_name, tm )
go
CREATE EVENT ConnFail TYPE ConnectFailed
HANDLER
BEGIN
    DECLARE usr CHAR(128);
    SET usr = event_parameter( 'User' );
    // Put a limit on the number of failures logged.
    IF (SELECT COUNT(*) FROM DBA.ConnectionFailure
        WHERE user_name = usr
        AND tm >= DATEADD( minute, -30, CURRENT_TIMESTAMP )
        ) < 20
    THEN
        INSERT INTO DBA.ConnectionFailure( user_name )
            VALUES( usr );
        COMMIT;
        // Delete failures older than 7 days.
        DELETE DBA.ConnectionFailure
        WHERE user_name = usr
        AND tm < DATEADD( day, -7, CURRENT_TIMESTAMP );
        COMMIT;
    END IF;
END
go
CREATE PROCEDURE DBA.login_check( )
BEGIN
```

```

DECLARE usr CHAR(128);
DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';
SET usr = CONNECTION_PROPERTY( 'userid' );
// Block connection attempts from this user
// if 3 or more failed connection attempts have occurred
// within the past 30 minutes.
IF ( SELECT COUNT( * ) FROM DBA.ConnectionFailure
      WHERE user_name = usr
      AND tm >= DATEADD( minute, -30,
                        CURRENT_TIMESTAMP ) ) >= 3 THEN
    SIGNAL INVALID_LOGON;
ELSE
    CALL dbo.sp_login_environment;
END IF;
END
go
GRANT EXECUTE ON DBA.login_check TO PUBLIC
go
SET OPTION PUBLIC.login_procedure='DBA.login_check'
go

```

次の例は、ユーザのパスワードの有効期限が切れていることを知らせるエラーを通知する方法を示しています。ただし、パスワードの有効期限切れの通知を実装するには、ログインポリシーを使用する方が適しています。

```

CREATE PROCEDURE DBA.check_expired_login( )
BEGIN
    DECLARE PASSWORD_EXPIRED EXCEPTION FOR SQLSTATE '08WA0';
    IF( condition-to-check-for-expired-password ) THEN
        SIGNAL PASSWORD_EXPIRED;
    ELSE
        CALL dbo.sp_login_environment;
    END IF;
END;

```

## 関連情報

[セキュリティ: パスワード \[1602 ページ\]](#)

[ログインポリシー \[613 ページ\]](#)

[post\\_login\\_procedure オプション \[754 ページ\]](#)

[NewPassword \(NEWPWD\) 接続パラメータ \[94 ページ\]](#)

## 1.5.6.7.71 materialized\_view\_optimization オプション

クエリへの応答を効率化するためにオプティマイザがマテリアライズドビューを使用する方法を指定します。

### 指定可能な値

*Disabled, Fresh, Stale, N { Minute[s] | Hour[s] | Day[s] | Week[s] | Month[s] }*

## デフォルト

Stale

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

materialized\_view\_optimization オプションでは、オプティマイザが古いマテリアライズドビューを使用できる状況を指定できます。

マテリアライズドビュー内のデータは、そのビューが参照するベーステーブルのデータが更新されることによって古くなります。マテリアライズドビューの再表示頻度を設定するときには、データの古さをどの程度まで許容できるかを考慮してください。また、再表示の処理中はビューがクエリに応答できないため、ビューの再表示に要する時間も考慮します。さらに、データベースの現在の状態が反映されていない可能性がある結果をデータベースサーバから返すことを許容するかどうかも考慮します。このオプションは、以下のいずれかに設定できます。

### Disabled

クエリの最適化にマテリアライズドビューを使用しません。

### Fresh

古くなっていない (基本となるテーブル内のデータが前回の再表示以降に変更されていない) 場合のみマテリアライズドビューを使用します。

### Stale

古くなっていてもマテリアライズドビューを使用します。この値はデフォルトの設定です。

**N { Minute[s] | Hour[s] | Day[s] | Week[s] | Month[s] }**

古いマテリアライズドビューが指定された時間内に再表示されている場合にかぎり、新しいマテリアライズドビューと古いマテリアライズドビューを使用します。指定する値は分単位とし、2<sup>31</sup> 分未満であることが必要です。データベースサーバは、1 週間を 7 日、1 か月を 30 日と見なします。

クエリがマテリアライズドビューを直接参照する場合は、古くなっているかどうかにかかわらず、そのビューが使用されます。この場合、materialized\_view\_optimization オプションは効力を持ちません。

## 1.5.6.7.72 max\_client\_statements\_cached オプション

クライアントでキャッシュされる文の数を制御します。

### 指定可能な値

整数 (0 ~ 100)

### デフォルト

10

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

クライアントで文をキャッシュすると、同一の SQL 文が複数回準備されたときに、データベース要求と文の準備が減少します。同じ SQL 文の準備と削除が繰り返し行われると、クライアントは、文がデータベースサーバに残された状態で、その文をキャッシュします。この文がアプリケーションによって削除された後も、継続してキャッシュします。文のキャッシュにより、データベースサーバは、文の削除や再準備などの余分な作業を節約できます。スキーマの変更、データベースオプション設定の変更、DROP VARIABLE 文の実行が生じると、準備文は自動的に削除され、その SQL 文が次に実行されるときに再び準備されます。これにより、不正な動作の原因となり得るような、キャッシュされた文が再使用されることのないようにします。

このオプションは、準備 (キャッシュ) されたまま維持できる文の最大数を指定します。キャッシュされた文は、max\_statement\_count リソースガバナナでは考慮されません。

このオプションの設定は、Embedded SQL、ODBC、OLE DB、ADO.NET、SQL Anywhere JDBC ドライバを使用して作成された接続に適用されます。SAP Open Client、jConnect、HTTP 接続には適用されません。

このオプションに 0 を設定すると、クライアントの文のキャッシュが無効になります。この値を増加すると、アプリケーションが同じ SQL 文について 10 回以上にわたって準備と削除を繰り返す場合に、パフォーマンスが向上する可能性があります。たとえば、25 の SQL 文をスルーするループを処理するアプリケーションについて考えてみます。ループをスルーすることに準

備と削除を繰り返し、これらの SQL 文のそれぞれにまったく同じ文が含まれているときは、このオプションを 25 に設定すると、パフォーマンスが向上します。

このオプションの値を増加させると、クライアントにおけるメモリの使用量が増加し、データベースサーバに対するキャッシュ要求が高まります。キャッシュされた文が大量になった場合で、スキーマの変更やオプション設定の変更によって文を再使用できなくなると、その接続において、文のキャッシュは自動的に無効になります。文のキャッシュが自動的に無効にされると、クライアントは再び定期的に文のキャッシュを実行して処理方法を再評価し、文のキャッシュを再び有効にすることに効果があるかどうかを判断します。

クライアントでの文のキャッシュは、次の場合に予期しない結果をもたらす可能性があります。

1. 文を準備し、記述します。その記述は結果のない文を返します。

```
CREATE OR REPLACE FUNCTION test() RETURNS INT BEGIN RETURN 1;
END;
CALL test();
```

2. DDL 文によって同じ文のテキスト (この例では CALL 文) が同じ接続に結果セットを返すようになる場合。次に例を示します。

```
CREATE OR REPLACE PROCEDURE test() BEGIN SELECT 2;
END;
CALL test();
```

クライアントでの文のキャッシュが有効になっているときに、2 番目の CALL test() 文が誤って記述されます。

tracetime.pl Perl スクリプトを使用してログが分析される場合は、要求ログが取得される間、max\_client\_statements\_cached オプションを 0 に設定して、クライアントでの文のキャッシュを無効にする必要があります。

## 関連情報

[max\\_statement\\_count オプション \[733 ページ\]](#)

## 1.5.6.7.73 max\_connections オプション

データベースに対して許可する同時接続数を制御します。

### 指定可能な値

整数 (0 またはそれ以上)、NULL

## デフォルト

ネットワークデータベースサーバに同時接続できるデフォルトの限界数は、パーシートライセンスにより許容される最大接続数と同じです。またはパーコアライセンスの場合は、32766 接続です。

パーソナルサーバに同時接続できるデフォルトの限界数は 10 で、そのうち少なくとも 3 つは標準接続に使用される必要があります。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	いいえ	いいえ

## 備考

max\_connections オプションは、データベースに対して許可する同時接続数を指定します。制限に達すると、次のようになります。

- データベースへの接続数が指定した制限を下回るまで、後続の標準接続試行は失敗します。
- その後の HTTP/HTTPS 接続試行は、接続数が指定した制限を下回るか、キューに入れられた接続がタイムアウトするまでキューに入れられます。  
HTTP/HTTPS キューに接続が入っている間、HTTP/HTTPS 接続の代わりに標準接続を行うことはできません。標準接続用にデータベース接続の一部が予約されている場合以外は、標準接続を行うとしているユーザは無限に待機することとなります。reserved\_connections オプションを使用することで、接続を予約します。

接続の限界数に達した場合でも、DROP CONNECTION または SERVER OPERATOR システム権限を持つ DBA ユーザは、データベースへの標準接続を 1 つ作成し、必要に応じて接続を削除できます。

## 例

- 次の例では、ネットワークデータベースサーバで実行されているデータベースの同時接続の最大数を 1000 に設定します。

```
SET OPTION PUBLIC.max_connections=1000;
```

- 次の例では、パーソナルデータベースサーバで実行されているデータベースの同時接続の最大数を 3 に設定します。

```
SET OPTION PUBLIC.max_connections=3;
```

## 関連情報

[データベース接続の制限 \[239 ページ\]](#)

### 1.5.6.7.74 max\_cursor\_count オプション

接続で一度に使用できるカーソルの最大数を制限するリソースガバナーを制御します。

#### 指定可能な値

整数

#### デフォルト

50

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい (現在の接続のみ)、SET ANY SYSTEM OPTION あり	×

#### 備考

このリソースガバナーを使用すると、接続ごとにユーザが使用できるカーソルの数を、DBA が制限できます。接続の制限を超える操作が行われると、リソースのガバナーを超過していることを示すエラーを出力します。

接続でストアードプロシージャを実行する場合、そのプロシージャはプロシージャの所有者の権限で実行されます。ただし、プロシージャが使用するリソースは、現在の接続に割り当てられています。

オプションを 0 (ゼロ) に設定すると、リソース制限を削除できます。

## 1.5.6.7.75 max\_parallel\_statements オプション

BEGIN PARALLEL WORK 文の内部で指定され、データベースサーバがいつでも並列実行できる文の最大数を指定します。

### 指定可能な値

整数

### デフォルト

0 (データベースサーバは選択した任意の数の文を並列実行します)。

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

max\_parallel\_statements オプションを 1 に設定すると、BEGIN PARALLEL WORK 文内の文の並列実行が無効化されます。

max\_parallel\_statements オプションでは、sa\_server\_option システムプロシージャにおける、データベースサーバオプション -gtc および -gta の設定と、ProcessorAffinity オプションの設定が考慮されます。

#### 例

次の例では、max\_parallel\_statements オプションを 4 に設定します。

```
SET TEMPORARY OPTION max_parallel_statements = 4;
```

### 関連情報

[CREATE INDEX 文の一覧または LOAD TABLE 文の一覧の同時実行によるパフォーマンスの向上 \[1410 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

[-gta データベースサーバオプション \[431 ページ\]](#)

## 1.5.6.7.76 max\_plans\_cached オプション

キャッシュに格納される実行プランの最大数を指定します。

### 指定可能な値

整数

### デフォルト

20

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションは、各接続でキャッシュされるプランの最大数を指定します。オプティマイザは、ストアードプロシージャ、ファンクション、トリガの中で実行されるクエリ、INSERT、UPDATE、DELETE の各文の実行プランをキャッシュします。ある接続でストアードプロシージャ、ストアード関数、またはトリガに含まれる文が複数回実行された後、オプティマイザは、その文の再使用可能なプランを構築します。

再利用可能なプランでは、選択性推定やリライト最適化にホスト変数の値は使用されません。この動作の結果として、文が再び最適化された場合に、再利用可能なプランのコストが高くなる可能性があります。再使用可能なプランのコストが文に最適と思われるコストに近いとき、オプティマイザはそのプランをプランキャッシュに追加します。

このキャッシュは、CREATE TABLE や DROP TABLE など、テーブルスキーマを変更する文が実行されたときにクリアされません。宣言されたテンポラリテーブルを参照する文はキャッシュされません。

このオプションに 0 を設定すると、プランのキャッシュが無効になります。

## 1.5.6.7.77 max\_priority オプション

接続の最高優先度レベルを制御します。

### 指定可能な値

Critical、High、Above Normal、Normal、Below Normal、Low、Background

### デフォルト

normal

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい (現在の接続のみ)、SET ANY SYSTEM OPTION あり	×

### 備考

別の優先度レベルをスケジューリングすると、要求の優先度レベルに関係なく、すべての要求に CPU 時間が割り当てられます。優先度レベルが高い要求ほど、多くの時間が割り当てられます。

### 関連情報

[background\\_priority オプション \[廃止予定\] \[660 ページ\]](#)

[priority オプション \[761 ページ\]](#)

## 1.5.6.7.78 max\_query\_tasks オプション

データベースサーバがクエリの並列処理に使用できるサーバタスクの最大数を指定します。

### 指定可能な値

整数

### デフォルト

0

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

### 備考

max\_query\_tasks オプションは、クエリ処理で使用できる並列処理の最大レベルを設定します。このオプションは、クエリの並列処理に使用できるデータベースサーバタスクの数を設定します。デフォルト値は 0 です。この場合、データベースサーバは必要と判断した数のタスクを使用します。max\_query\_tasks オプションに 0 以外の値を指定した場合は、1 つのクエリについて使用可能なタスクの最大数が設定されます。max\_query\_tasks オプションを 1 に設定すると、クエリ内並列処理が無効になります。

すべての要求に対してデータベースサーバが使用できるタスクの数は、起動時に -gn オプションで設定されたしきい値によって制限されます。この数は、サーバがサービスを提供しているすべてのデータベースと接続に対してグローバルな最大数です。1 つの要求に使用されるタスクの数も、データベースサーバで利用できる論理プロセッサの数によって制限されます。たとえば、-gtc オプションでプロセッサの同時実行性を 1 に設定した場合、クエリ内並列処理は無効となります。

クエリ内並列処理が有効である場合は、特定の条件を満たす SELECT 文がクエリ内並列処理によって処理されます。このクエリのアクセスプランでは、クエリ内並列処理が使用されたことが交換演算子によって示されます。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこのオプションに対するテンポラリ設定や PUBLIC 設定よりも優先させることができます。

## 関連情報

[スレッド \[330 ページ\]](#)

[マルチプログラミングレベルのデータベースサーバ設定 \[334 ページ\]](#)

[-gn データベースサーバオプション \[421 ページ\]](#)

[-gt データベースサーバオプション \[430 ページ\]](#)

[-gtc データベースサーバオプション \[432 ページ\]](#)

## 1.5.6.7.79 max\_recursive\_iterations オプション

再帰共通テーブル式が反復できる最大回数を制限します。

### 指定可能な値

整数

### デフォルト

100

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

指定の反復回数内で計算が完了しなかった場合、再帰共通テーブル式の計算はアボートされてエラーが生成されます。再帰サブクエリは、反復が発生するたびに必要なリソースの総量が幾何学的に増加することがあります。このオプションを設定して、無限回帰が検出される前に費やされる時間とリソースの量を制限し、再帰共通テーブル式が意図どおりに機能できるようにしてください。

このオプションに 0 を設定すると、再帰共通テーブル式が無効になります。

## 1.5.6.7.80 max\_statement\_count オプション

接続で一度に使用できる準備文の最大数を制限するリソースガバナーを制御します。

### 指定可能な値

整数

### デフォルト

50

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい (現在の接続のみ)、SET ANY SYSTEM OPTION あり	×

### 備考

準備文を使用するアプリケーションは、準備文が不要になった後で明示的に削除されなかった場合、"準備文のリソースガバナーが制限を超えています" というエラーを受け取ることがあります。max\_statement\_count オプションはリソースガバナーであり、これを使用することにより、DBA は 1 つの接続で使用される準備文の数を制限できます。接続の制限を超える操作が行われると、リソースのガバナーを超過していることを示すエラーを出力します。

接続でストアードプロシージャを実行する場合、そのプロシージャはプロシージャの所有者の権限で実行されます。ただし、プロシージャが使用するリソースは、現在の接続に割り当てられています。

データベースサーバには、接続によって作成される準備文ごとにデータ構造体が用意されます。アプリケーションがデータベースサーバに準備文が不要になったことを通知するか、接続が切断されるまで、これらの構造体は解放されません。接続ごとの準備文の数を減らすためには、DROP STATEMENT 要求に相当する処理を実行する必要があります。次の表は、SQL Anywhere でサポートされる API で実行できる文を示しています。

インタフェース	文
ADO	RecordSet.Close
ADO.NET	SADataReader.Close または SADataReader.Dispose
Embedded SQL	DROP STATEMENT
Java	resultSet.Close、Statement.Close
ODBC	SQLFreeStmt( hstmt, SQL_DROP ) または SQLFreeHandle( SQL_HANDLE_STMT, hstmt )

### i 注記

Java および .NET では、文を明示的に削除する必要があります。言語ルーチンからはステートメントリソースの割り付けを解除するサーバ呼び出しが発行されないため、ガーベジコレクションを利用してこのクリーンアップを実行しないでください。また、ガーベジコレクションルーチンがいつ実行されるかは保証されません。

サーバにおいて接続のいずれかの時点でデフォルトの最大数を超える準備文のサポートが必要になった場合は、`max_statement_count` をデフォルトより大きな値に設定する必要があります。ただし、アクティブな準備文が多くなると、サーバメモリの使用量が増加します。`max_statement_count` オプションを 0 (ゼロ) に設定し、準備ステートメントリソースガバナーを無効にすることもできますが、この方法は推奨できません。このようにすると、アプリケーションが準備文を適切に解放しない場合、データベースサーバがメモリ不足によって異常終了しやすくなります。

## 1.5.6.781 max\_temp\_space オプション

1 つの接続で使用できるテンポラリファイル領域の最大サイズを設定します。

### 指定可能な値

Integer [ *k* | *m* | *g* | *p* ]

### デフォルト

0

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり	はい、SET ANY SYSTEM OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	はい (現在の接続のみ)、SET ANY SYSTEM OPTION あり	×

## 備考

このオプションでは、1つの接続で使用できるテンポラリファイル領域の最大サイズを指定できます。このテンポラリファイル領域の最大サイズを超えると、要求は失敗します。max\_temp\_space オプションが効力を持つためには、temp\_space\_limit\_check オプションを On (デフォルト) に設定する必要があります。

デフォルト値の 0 は、1つの接続が要求できるテンポラリファイル領域のサイズに制限がないことを意味します。0 以外の値は、1つの接続で使用できるテンポラリファイル領域のバイト数を示します。単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ k、m、g のいずれかを使用してください。p を使用すると、値は使用できるテンポラリファイル領域の総量に対する割合を表します。

テンポラリファイル領域を要求する接続では、データベースサーバが max\_temp\_space オプションの設定に対する制限をチェックし、要求がテンポラリファイル領域の最大サイズを超えていないことを確認します。接続が最大サイズを超えるテンポラリファイル領域を要求すると、その要求は失敗し、エラー SQLSTATE\_TEMP\_SPACE\_LIMIT が生成されます。

### 例

次の文は接続に対して 1 GB の制限を設定します。

```
SET OPTION PUBLIC.max_temp_space = '1g';
```

次の 2 つの文は、いずれも接続に対して 1 MB の制限を設定します。

```
SET OPTION PUBLIC.max_temp_space = 1048576;
```

```
SET OPTION PUBLIC.max_temp_space = '1m';
```

次の文は、テンポラリファイル領域の使用を全体の 5 % に制限します。

```
SET OPTION PUBLIC.max_temp_space = '5p';
```

## 関連情報

[temp\\_space\\_limit\\_check オプション \[807 ページ\]](#)

## 1.5.6.7.82 min\_password\_length オプション

新しいパスワードの最小長をデータベースに設定します。

### 指定可能な値

#### 整数

値はバイト数で指定します。シングルバイト文字セットの場合、この値は文字数と同じです。

### デフォルト

6

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×

### 備考

このオプションを使用すると、データベース管理者は、セキュリティを強化するために、新しいパスワードすべてに最小長を設定できます。既存のパスワードは影響を受けません。パスワードは、最大長が 255 バイトで、大文字と小文字が区別されません。

#### 例

新しいパスワードの最小長を 6 バイトに設定します。

```
SET OPTION PUBLIC.min_password_length = 6;
```

## 関連情報

[セキュリティ: パスワード \[1602 ページ\]](#)

[verify\\_password\\_function オプション \[826 ページ\]](#)

### 1.5.6.7.83 min\_role\_admins オプション

ロールの管理に必要な管理者の最小数を設定します。

#### 指定可能な値

整数 (0 ~ 10)

#### デフォルト

1

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	×	×
一時的な設定が許可されているかどうか	×	×	×

#### 備考

このオプション設定はすぐに有効になります。

ロールの作成、削除、取り消し、ユーザパスワードの NULL への変更、ユーザの手動ロック、またはログインポリシーの条件に違反したことによるユーザの自動ロックをする場合、データベースサーバはロール管理者の数が、指定の最小数を下回らないようにします。期限切れパスワードを持つロール管理者、ロックアウトされていないロール管理者を含む、NULL でないパスワードを持つロール管理者のみが数に含まれます。

このオプションの値の変更は、各ロールが少なくとも指定されたロール管理者数を持っている場合にのみ許可されます。ユーザからロールを取り消している間に、管理者数が min\_role\_admins オプションの値を下回った場合は、文からエラーが返されます。

### i 注記

そのユーザのログインポリシーに対して *locked* 属性を ON に設定することで、ユーザを手動でロックします。これにより、ユーザはデータベースに対して新しい接続を作成できなくなります。

ユーザを自動ロックできるログインポリシーの条件には、`max_failed_login_attempts` または `max_days_since_login` を超えることが含まれます。

### 例

次の例は、`min_role_admins` オプションを 2 に設定します。

```
SET OPTION PUBLIC.MIN_ROLE_ADMINS = 2;
```

## 関連情報

[ロール \[1455 ページ\]](#)

## 1.5.6.7.84 ml\_remote\_id オプション [Mobile Link]

Mobile Link 同期システム内のリモートデータベースのリモート ID を設定します。

### 指定可能な値

Mobile Link による同期時にデータベースをユニークに識別する任意の値

### デフォルト

空の文字列

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり

	PUBLIC ロール	現在のユーザ	他のユーザ
一時的な設定が許可されているかどうか	x	x	x

## 備考

Mobile Link 同期では、PUBLIC ロール設定のみを使用します。

このオプションは、Mobile Link 同期システム内のリモートデータベースのリモート ID を割り当てます。リモート ID は、同期システム内でユニークでなければなりません。

リモート ID を手動で設定し、その後リモートデータベースを再作成した場合は、再作成したリモートデータベースに古いリモートデータベースとは異なる名前を付けるか、ml\_reset\_sync\_state ストアドプロシージャを使用して、統合データベース内でリモートデータベースのステータス情報をリセットします。

### 警告

リモート ID は最初の同期を実行する前に変更すると安全です。リモート ID を後で変更する場合は、変更する直前に同期処理の実行を完了しておくようにしてください。同期処理を行わないでリモート ID を変更すると、一部のデータが失われ、データベースの整合性が保たれなくなる可能性があります。

### 例

次の SQL 文では、リモート ID を HR001 に設定します。

```
SET OPTION PUBLIC.ml_remote_id = 'HR001'
```

## 1.5.6.7.85 nearest\_century オプション

文字列から日付への変換で、2 桁の年の解釈を制御します。

### 指定可能な値

整数 (0 ~ 100)

### デフォルト

50

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、文字列から日付またはタイムスタンプに変換するときに、2 桁の年の処理を制御します。

nearest\_century 設定は、ロールオーバーポイントとして動作する数値です。この値より小さい 2 桁の年は 20yy に変換され、この値以上の年は 19yy に変換されます。

従来の SQL Anywhere では、年に 1900 を加算していました。Adaptive Server Enterprise では基準年を使用するので、yy が 50 より小さい場合は 20yy になります。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.86 non\_keywords オプション

個々の予約済みキーワードを無効にして、識別子として使用できるようにします。

## 指定可能な値

文字列

## デフォルト

空の文字列

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

## 備考

特定のキーワードが予約済みキーワードとして認識されないようにするには、それらのキーワードのカンマ区切りリストを指定します。このオプションは、古いバージョンの製品で作成されたアプリケーションが新しいキーワードの追加によって壊されないようにするための手段となります。キーワードとして認識されるようになった識別子がデータベース内にある場合は、その識別子に対するすべての参照を二重引用符、角括弧、または逆引用符 (`...`) で囲むか、non\_keywords オプションを使用してキーワードを無効にすることができます。

ある単語が単語のカンマ区切りリストで受け入れられるかどうかは、次のルールにより決まります。

- その単語が予約語候補のリストに表示されている必要があります。
- その単語が予約語 SET、OPTION、または OPTIONS のいずれかではあってはなりません。

予約語候補のリストは、sa\_reserved\_words システムプロシージャにより生成されます。"予約語" および "予約キーワード" という用語は、このマニュアルでは相互に置き換え可能な方法で使用されています。

次の文を記述すると、TRUNCATE と SYNCHRONIZE がキーワードとして認識されません。

```
SET OPTION non_keywords = 'TRUNCATE, SYNCHRONIZE';
```

このオプションで新規設定を行うと、前に設定しているものに置き換わります。

次の文は以前の設定内容をすべてクリアします。

```
SET OPTION non_keywords =;
```

このオプションの副次的影響として、無効化されたキーワードを必要とする SQL 文が実行できなくなり、それらの文は構文エラーとなります。たとえば、SELECT キーワードを無効化すると、SELECT 文が実行できなくなります。

個々のキーワードを有効化するには、reserved\_keywords オプションを使用します。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[reserved\\_keywords オプション \[775 ページ\]](#)

[reserved\\_keywords オプション \[775 ページ\]](#)

## 1.5.6.7.87 odbc\_describe\_binary\_as\_varbinary オプション

SQL Anywhere ODBC ドライバの BINARY カラムの記述方法を制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションを使用すると、すべての BINARY カラムと VARBINARY カラムを、BINARY としてまたは VARBINARY として、アプリケーションに対して記述するかどうかを選択できます。デフォルトでは、SQL Anywhere ODBC ドライバは BINARY と VARBINARY の両方のカラムを SQL\_BINARY として記述します。このオプションを On に設定すると、ODBC ドライバは BINARY と VARBINARY カラムを SQL\_VARBINARY として記述します。このオプションの設定にかかわらず、BINARY カラムと VARBINARY カラムを区別することはできません。

BINARY カラムは常に 0 埋め込みで、VARBINARY カラムはそうではない Delphi アプリケーションを使用する場合は、このオプションを On に設定することをお奨めします。このオプションを On にしてすべてのカラムが可変長データ型として扱われるようにすると、Delphi のパフォーマンスが向上します。

## 1.5.6.7.88 odbc\_distinguish\_char\_and\_varchar オプション

SQL Anywhere ODBC ドライバの CHAR カラムの記述方法を制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

接続を開くと、SQL Anywhere ODBC ドライバは、このオプションの設定を使用して CHAR カラムの記述方法を決定します。このオプションを Off (デフォルト) に設定すると、CHAR カラムは SQL\_VARCHAR として記述されます。このオプションを On に設定すると、CHAR カラムは SQL\_CHAR として記述されます。VARCHAR カラムは、常に SQL\_VARCHAR として記述されます。

## 1.5.6.7.89 oem\_string オプション

データベースファイルのヘッダページ内にあるユーザ指定の情報を保存します。

### 指定可能な値

文字列 (最大 128 バイト)

### デフォルト

空の文字列

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

### 備考

データベースファイルのヘッダページ内の情報を保存しておき、アプリケーションから直接ファイルを読み込んで情報を抽出できます。このページはシステム DB 領域ファイルのヘッダに保存されます。OEM 文字列に対して 128 バイトを超える値を指定すると、エラーが返されます。

スキーマバージョン、アプリケーション名、アプリケーションのバージョンなどの情報は、保存しておく役に立つことがあります。また、アプリケーションがデータベースファイルを使用する前に検証のために読み込む文字列を保存することによって、アプリケーションはデータベースを起動しなくても OEM 文字列によってデータベースファイルがそのアプリケーションに関連付けられているかどうかを確認できます。つまり、アプリケーションを設計するときに、OEM 文字列を使用してデータベースファイルがそのアプリケーション用であることを検証するプロセスを組み込むことができます。さらに、ユーザに表示するメタデータを抽出することもできます。

oem\_string をシステム DB 領域ファイルのヘッダ内に設定するには、次の文を実行します。

```
SET OPTION PUBLIC.oem_string=user-specified-string;
```

user-specified-string の値は、ISYSOPTION システムテーブルとシステム DB 領域ファイルのヘッダの両方に保存されます。この文字列が SET OPTION 文に渡されるときには変換が行われられないため、文字列を必要な文字セットで定義して

から SET OPTION 文に指定する必要があります。文字列を必要な文字セットに変換するには、CSCONVERT 関数を使用します。

oem\_string の値を問い合わせるには、以下の方法があります。

- oem\_string 接続プロパティを使用します。

```
SELECT CONNECTION_PROPERTY( 'oem_string' );
```

- SYSOPTION システムビューを使用します。

```
SELECT setting FROM SYSOPTION WHERE "option" = 'oem_string';
```

oem\_string ディレクトリには 2 つのサンプルプログラムが含まれています。

- dboem.cpp は、OEM 文字列を抽出してデータベースサーバメッセージウィンドウに出力する方法を示す C プログラムです。
- dboem.pl は、OEM 文字列を抽出して PERL スクリプト内の stdout に出力する方法を示します。

### 警告

アプリケーションからデータベース内の OEM 文字列に直接書き込みを行うことはできません。そのようにすると、データベースのヘッダページが破損します。

Windows では、サーバがこのデータベースファイルを読み込んだ場合、アプリケーションからこのファイルを直接読み込むことはできません。データベースサーバは、このファイルに排他ロックを設定します。これに対し、サポートされる UNIX プラットフォームでは、読み込みパーミッションのあるアプリケーションであれば、このファイルをいつでも直接読み込むことができます。ただし、OEM 文字列に対する変更は、直ちにファイルに反映されないことがあります。チェックポイントを発行すると、データベースサーバがページ 0 をディスクにフラッシュするため、現在の OEM 文字列値がファイルに反映されます。

万一、OEM 文字列の変更から次のチェックポイントまでの間にデータベースサーバが障害を起こした場合、新しい OEM 文字列値がファイルヘッダに反映されないことがあります。新しい OEM 文字列値はデータベースのリカバリ後に正しく設定されます。

### 例

次の例は、データベースファイルに関する情報を示す OEM 文字列を暗号化し、データベースヘッダファイルに保存します。

```
BEGIN
  DECLARE @v VARCHAR(100);
  SET @v = BASE64_ENCODE( ENCRYPT( 'database version 10', 'abc' ) );
  EXECUTE IMMEDIATE 'SET OPTION PUBLIC.oem_string = ''' || @v || '''';
END;
```

OEM 文字列の値を取得するには、次の文を実行します。

```
SELECT DECRYPT(
  BASE64_DECODE(
    CONNECTION_PROPERTY( 'oem_string' ) ), 'abc' )
```

## 1.5.6.7.90 on\_charset\_conversion\_failure オプション

文字の変換中にエラーが発生した場合の動作を制御します。

### 指定可能な値

Ignore、Warning、Error

### デフォルト

Ignore

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

文字の変換中にエラーが発生した場合の動作を次のように制御します。

#### Ignore

エラーも警告も表示しません。

#### 警告

置換と不正な文字列を警告としてレポートします。不正な文字列は変換されません。

#### エラー

置換と不正な文字列をエラーとしてレポートします。

クライアントとデータベース間で文字セットの変換が必要な場合、不正な文字が検出されたり文字の置換が使用されたりしたときに、それを無視するか、警告を返すか、エラーを返すかをこのオプションで制御します。

シングルバイトからシングルバイトへのコンバータは、置換や不正な文字列をレポートすることはできず、Ignore に設定する必要があります。

このオプションは、損失を伴う変換がクライアントで発生した場合の動作は制御しません。たとえば、クライアントからの SQL 文は CHAR データベース文字セットに格納されているか、変換される必要があります。Unicode のクライアントアプリケーション

ンが SQL 文を準備し、その文に CHAR データベース文字セットで表現できない文字が含まれているとします。この場合、置換文字が代わりに使用されます。ただし、損失を伴う変換がクライアントで発生したため、データベースサーバではこの変換が発生したことがわかりません。

## 1.5.6.7.91 on\_tsq\_error オプション

ストアードプロシージャのエラー処理を制御します。

### 指定可能な値

#### Stop

エラーの検出と同時に実行を停止します。

#### Conditional

プロシージャが ON EXCEPTION RESUME を使用していて、エラーのすぐ後ろの文がエラーを処理する場合は継続します。それ以外の場合は終了します。

#### Continue

次に続く文に関係なく実行は継続されます。エラーが複数ある場合は、ストアードプロシージャに最初に出現したエラーがプロシージャの完了後に返されます。

### デフォルト

Conditional

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('Continue') によって設定されます。

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、ストアードプロシージャと Transact-SQL バッチのエラー処理を制御します。このオプションは BEGIN...END 文の TRY ブロック内では効果がなく、エラー処理の動作は TRY ブロック内で定義されます。

on\_tsq\_error の設定 Conditional と Continue は、いずれも Adaptive Server Enterprise との互換性を確保するために使用します。Continue が Adaptive Server Enterprise 動作を最も忠実にシミュレートします。エラーがより早期にレポートされるようにするには、Transact-SQL の新しいストアードプロシージャを作成するときに Conditional 設定を使用します。

このオプションを Stop または Continue に設定した場合は、その設定が continue\_after\_raisererror オプションの設定より優先されます。ただし、このオプションを Conditional (デフォルト) に設定した場合は、RAISERROR 文の後の動作は continue\_after\_raisererror オプションによって決まります。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[continue\\_after\\_raisererror オプション \[677 ページ\]](#)

## 1.5.6.7.92 optimization\_goal オプション

クエリ処理の最適化の対象を、最初のローを迅速に返すこと、または完全な結果セットを返すコストを最小限に抑えることのできるに指定します。

### 指定可能な値

First-row または All-rows

### デフォルト

All-rows

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

optimization\_goal オプションは、SQL データ操作言語 (DML) 文を応答時間に対して最適化するか、リソースの総消費量に対して最適化するかを制御します。

このオプションを All-rows (デフォルト) に設定すると、クエリが最適化され、予測される合計検索時間が最短になるアクセスプランを選択します。SAP PowerBuilder DataWindow アプリケーションなど、処理の前に結果セット全体が必要になるアプリケーションでは、optimization\_goal を All-rows に設定するのが適切です。All-rows の設定では、カーソルが開いたときに結果全体が実体化されるため、insensitive (ODBC の静的) カーソルにも適しています。また、結果セットのスクロールを目的とするスクロール (ODBC キーセット駆動型) カーソルにも適しています。

このオプションを First-row に設定すると、オプティマイザは、クエリの結果の最初のローをフェッチするまでの時間を短縮するアクセスプランを選択します。この場合、検索にかかる合計時間は長くなることがあります。また、通常オプティマイザでは、可能であれば結果の実体化を必要とするアクセスプランは使用しないで、最初のローを返すまでの時間を短縮します。この設定では、オプティマイザは、明示的なソートの操作を必要とするアクセスプランではなく、クエリの ORDER BY 句を満たすインデックスを使用するアクセスプランを採用します。

クエリの FROM 句の FASTFIRSTROW テーブルヒントを使用すると、特定のクエリの最適化目標を First-row に設定できます。この場合、optimization\_goal の設定を変更する必要はありません。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこのオプションに対するテンポラリ設定や PUBLIC 設定よりも優先させることができます。

## 1.5.6.7.93 optimization\_level オプション

クエリオプティマイザが SQL 文のアクセスプランの検索に費やす作業量を制御します。

## 指定可能な値

0-15

## デフォルト

9

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

optimization\_level オプションは、オプティマイザが SQL データ操作言語 (DML) の最適化に費やす作業量を制御します。このオプションは、オプティマイザが SELECT ブロックに対して考慮する代替ジョイン方式の最大数を制御します。optimization\_level の設定が高いほど、オプティマイザが考慮するジョイン方式の最大数が大きくなります。

このオプションを 0 に設定すると、オプティマイザは実行のために考慮する最初のアクセスプランを選択し、事実上、代替プランのコストベースの比較を避けることとなります。さらに、レベル 0 では、ネストされたクエリのセマンティックな最適化が一部無効になります。このオプションが 0 よりも大きい値に設定されると、オプティマイザは代替方式を評価し、予想コストが最も低いものを選択します。このオプションがデフォルトの 9 よりも大きい値に設定されると、オプティマイザは代替方式をより積極的に検索し、その結果、最適化フェーズで経過する時間がはるかに長くなる可能性があります。

代表的なシナリオでは、アプリケーションが DML 文に対してより速い OPEN 時間を必要とする場合、このオプションは一時的に低いレベル (0、1、2 など) に設定されます。文が複雑であってもクエリの実行時間は非常に短いので、オプティマイザによって選択された特定のアクセスプランはあまり重要ではないことがわかっています。optimization\_level の PUBLIC 設定をデフォルトから変更することはお奨めできません。

optimization\_level オプションの設定の結果は、optimization\_goal と optimization\_workload オプションの設定とは無関係です。

単純な DML 文 (特定の行を識別する WHERE 句に等号条件を含んだ単一ブロック、単一テーブルのクエリ) はヒューリスティックに最適化されるため、コストベースのオプティマイザをすべてバイパスします。単純な DML 文の最適化は、optimization\_level オプションの設定からは影響を受けません。オプティマイザバイパスメカニズムによって最適化された要求の数は、QueryBypassed 接続プロパティとして使用できます。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこのオプションに対するテンポラリ設定や PUBLIC 設定よりも優先させることができます。

## 関連情報

[接続プロパティの一覧 \[839 ページ\]](#)

## 1.5.6.7.94 optimization\_workload オプション

クエリ処理において、更新と読み込みを組み合わせた負荷に対して最適化するか、または大部分が読み込みベースの負荷 (OLAP) に対して最適化するかを決定します。

### 指定可能な値

Mixed、OLAP

### デフォルト

Mixed

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

### 備考

optimization\_workload オプションは、更新と読み取りが組み合わさった負荷、または主に読み取りベースの負荷のどちらの処理向けにクエリが最適化されるかを制御します。

このオプションが Mixed (デフォルト) に設定されている場合、オプティマイザは短い挿入、更新、削除と、実行時間の長い読み取り専用クエリを組み合わせた負荷に対して適切なクエリ最適化アルゴリズムを選択します。

このオプションが OLAP に設定されている場合、オプティマイザは、実行時間の長いクエリの大部分とバッチ更新を組み合わせた負荷に対して適切なアルゴリズムを選択します。特に、オプティマイザは、クラスタドハッシュ Group By クエリ実行アルゴリズムを使用するように選択する場合があります。

オプションが OLAP に設定されている場合、クラスタドハッシュ Group By アルゴリズムが有効になります。このオプションを Mixed (デフォルト) に設定すると、このアルゴリズムは無効になります。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこのオプションに対するテンポラリ設定や PUBLIC 設定よりも優先させることができます。

## 関連情報

[optimization\\_level オプション \[749 ページ\]](#)

### 1.5.6.7.95 parameterization\_level オプション

クライアント文の自動パラメータ化を制御します。

#### 指定可能な値

Off、Simple、Forced

#### デフォルト

Simple

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

#### 備考

クライアントアプリケーションで SQL 文の準備が整うと、データベースサーバは文を準備する前に、SQL テキスト内の定数リテラルをパラメータ用プレースホルダと置き換えることがあります。その結果、SQL 文はより汎用的になります。パラメータ化された定数値のみが異なる後続の SQL 文が実行されると、その SQL 文はパラメータ化された文と一致します。汎用化された SQL 文には、次のような 2 つのパフォーマンス上の利点があります。

- パラメータ化された文がクライアント接続によってキャッシュされると、後続の SQL 文の準備を回避できる可能性があります。
- データベースサーバがパラメータ化された文の実行プランをキャッシュすると、以降の SQL 文の最適化を回避できる可能性があります。

1 番目の利点では、クライアント文のキャッシュが有効である必要があります。2 番目の利点では、クライアント文のキャッシュとサーバのプランのキャッシュの両方が有効である必要があります。

パラメータ化はクライアントアプリケーションに対して透過的なもので、自動的に挿入されるパラメータは文またはカーソルの DESCRIBE からは参照できません。

パラメータ化の動作は、このオプションの次の値によって制御されます。

**Off** 文はパラメータ化されません。この設定をすると、ソフトウェアのバージョン 16 およびそれ以前のバージョンと同じ動作をします。

**Simple** データベースサーバは、文をパラメータ化するタイミングと、パラメータ化する各文内の定数を決定します。

**Forced** データベースサーバはできるだけ早急にすべての文をパラメータ化し、各文内のすべての対象定数をプレースホルダと置き換えます。ただし、構文的に無効であるか、一般的には使用されないという理由から、パラメータ化がサポートされない文のタイプとクエリコンストラクトもあります。したがって、そのようなコンテキスト内の定数リテラルは置き換えられません。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこのオプションに対するテンポラリ設定や PUBLIC 設定よりも優先させることができます。

## 関連情報

[max\\_client\\_statements\\_cached オプション \[724 ページ\]](#)

[接続プロパティの一覧 \[839 ページ\]](#)

## 1.5.6.7.96 pinned\_cursor\_percent\_of\_cache オプション

カーソルを固定するために使用できるキャッシュの割合を指定します。

### 指定可能な値

整数 (0 ~ 100)

### デフォルト

10

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

## 備考

データベースサーバは、カーソルの実装に必要なデータ構造を、仮想メモリのページに格納します。これらのページは、フェッチ要求から次のフェッチ要求までの間メモリ内にロックされているため、次のフェッチ要求を受信したときにすぐに使用できません。

メモリが少ない環境で、これらのページによって必要以上にキャッシュが占有されないように、カーソルの固定に使用するキャッシュの割合は制限されています。この制限を調整するには、`pinned_cursor_percent_of_cache` オプションを使用します。

このオプションの値は、0 から 100 までの割合で指定します。デフォルトは 10 です。このオプションを 0 に設定すると、カーソルのページがフェッチの間、固定されません。

## 1.5.6.7.97 post\_login\_procedure オプション

ユーザが接続したときにアプリケーションが表示する必要のあるメッセージが結果セットに含まれるプロシージャを指定します。

### 指定可能な値

文字列

### デフォルト

`dbo.sa_post_login_procedure`

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい (現在の接続のみ)、SET ANY SECURITY OPTION あり	いいえ

## 備考

post\_login\_procedure を空の文字列以外に設定すると、アプリケーションは、このオプションに指定されたプロシージャを接続プロセス中に呼び出し、ユーザに対して表示するメッセージを特定できます。このオプションには、**owner.function-name** という形式の値を指定する必要があります。これによって、ユーザが関数を無効にすることを防止できます。

このオプションを設定すると、SQL Central 用の SQL Anywhere17 プラグインである Interactive SQL がプロシージャを呼び出し、プロシージャが返すメッセージをウィンドウに表示します。SQL Anywhere に付属していないアプリケーションについては、必要に応じて、このオプションに指定されたプロシージャを呼び出してメッセージを表示するように変更を加えてください。

接続時にアプリケーションがメッセージを表示できる状況では、パスワード有効期限システムを実装している場合に、パスワードの期限切れが近づいていることをユーザに通知することがあります。このような機能を利用すると、ユーザが接続するたびに、パスワードが期限切れとなることを数日前に知らせることができます。

このオプションに指定するプロシージャは、1 つ以上のローと 2 つのカラムで構成される結果セットを返す必要があります。1 つ目のカラムは VARCHAR(255) 型で、メッセージのテキスト (メッセージがない場合は NULL) を返します。2 つ目のカラムは INT 型で、アクションのタイプを返します。アクションとして指定可能な値は次のとおりです。

0

メッセージを表示 (メッセージがある場合)

1

メッセージを表示し、ユーザにパスワードの変更を要求

2-99

予約済み

100 以上

ユーザ定義

SQL Anywhere17 プラグインと Interactive SQL (dbisql) は、アクションの値にかかわらず、NULL 以外であれば、すべてのメッセージを表示します。アクションを 1 に設定すると、これらのツールはユーザにパスワードを変更するよう要求し、新しいパスワードをユーザ指定の値に設定します。

post\_login\_procedure の使用と、パスワード有効期限の運用などの詳細なパスワード規則の実装を示す例については、「パスワード検証関数の使用」を参照してください。

## 例

次の例では、p\_post\_login\_check というプロシージャを使用して、ユーザにパスワードの期限切れが近づいていることを知らせ、パスワードを変更するよう要求します。

```
CREATE PROCEDURE DBA.p_post_login_check( )
RESULT( message_text VARCHAR(255), message_action INT )
BEGIN
  DECLARE message_text          CHAR(255);
  DECLARE message_action        INT;

  -- assume the password_about_to_expire variable was
  -- set by the login procedure
  IF password_about_to_expire = 1 THEN
    SET message_text = 'Your password is about to expire';
    SET message_action = 1;
  ELSE
    SET message_text = NULL;
    SET message_action = 0;
  END IF;
  -- return message (if any) through this result set
  SELECT message_text, message_action;
END;
GRANT EXECUTE ON DBA.p_post_login_check TO PUBLIC;
SET OPTION PUBLIC.post_login_procedure = 'DBA.p_post_login_check';
```

## 関連情報

[セキュリティ: パスワード \[1602 ページ\]](#)

[login\\_procedure オプション \[720 ページ\]](#)

[verify\\_password\\_function オプション \[826 ページ\]](#)

[NewPassword \(NEWPWD\) 接続パラメータ \[94 ページ\]](#)

## 1.5.6.7.98 precision オプション

10 進法計算での結果の最大桁数を指定します。

### 指定可能な値

整数 (1 ~ 127)

### デフォルト

30

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	いいえ	いいえ	いいえ

## 備考

精度は数値の合計桁数であり、小数点は含まれません。scale オプションは、計算結果が最大精度にトランケートされた場合の、小数点以下の最小桁数を指定します。たとえば、数値 3.1415926 の精度は 8、スケールは 7 です。

掛け算、割り算、足し算、引き算、集合関数はすべて結果が最大精度を超える可能性があります。

たとえば、DECIMAL(8,2) と DECIMAL(9,2) を掛け合わせる場合、結果には最大 DECIMAL(17,4) が必要です。precision が 15 の場合、15 桁のみが結果に保持されます。scale が 4 の場合、結果は DECIMAL(15,4) になります。scale が 2 の場合、結果は DECIMAL(15,2) になります。どちらの場合も、オーバーフローが発生する可能性があります。

## 関連情報

[scale オプション \[780 ページ\]](#)

## 1.5.6.7.99 prefetch オプション

クライアントアプリケーションで使用できるようになる前に、ローがクライアント側にフェッチされるかどうかを制御します。

### 指定可能な値

Off、Conditional、Always

### デフォルト

Conditional

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、クライアントアプリケーションで使用できるようになる前に、ローがクライアントサイドにフェッチされるかどうかを制御します。一度に複数のローをフェッチすると、クライアントアプリケーションが一度に1つずつローを要求した場合（カーソルのローをループする場合など）でも、応答時間が短縮され、データベースへの要求数の減少によって全体的なスループットも向上します。

- Off はプリフェッチが行われないことを意味します。
- Conditional (デフォルト) の場合は、カーソルタイプが SENSITIVE か、クエリにプロキシテーブルが含まれないかぎり、プリフェッチが発生します。
- Always は、SENSITIVE タイプのカーソルの場合も、プロキシテーブルに関連するカーソルの場合も、プリフェッチが行われることを意味します。

Always は、一部のカーソルのセマンティックに影響するため、注意して使用してください。たとえば、SENSITIVE タイプのカーソルが ASENSITIVE タイプになる場合があります。また、プリフェッチと、アプリケーションのフェッチ要求の間に値が更新された場合は、古い値がフェッチされることがあります。さらに、プロキシテーブルに関連するカーソルでプリフェッチを使用した場合、クライアントがプリフェッチローを再フェッチしようとする、エラー -668 "カーソルは FETCH NEXT 操作に制限されています" が発生する可能性があります。最初のフェッチの後で、初めてフェッチカラムが再バインドまたはバインドされた場合、クライアントは、ロールバックの後、または 0 の相対フェッチ時、あるいは GET DATA が使用されるときに、プリフェッチローを再フェッチしようとする場合があります。

value-sensitive カーソルタイプには、ESQL SENSITIVE と SCROLL カーソルタイプ、ODBC と OLE DB DYNAMIC および KEYSSET カーソルタイプが含まれます。

prefetch オプションの設定は、SAP Open Client 接続と jConnect 接続では無視されます。

DisableMultiRowFetch 接続パラメータが YES に設定されている場合、prefetch データベース オプションは無視され、プリフェッチは行われません。

このオプションでは、以前は On も有効な値でした。この値は現在は Conditional のエイリアスです。

## 関連情報

[DisableMultiRowFetch \(DMRF\) 接続パラメータ \[72 ページ\]](#)

## 1.5.6.7.100 preserve\_source\_format オプション

プロシージャ、トリガ、ビュー、イベントハンドラの元のソース定義をシステムファイルに保存するかどうかを制御します。保存した場合は、SYSTAB、SYSPROCEDURE、SYSTRIGGER、SYSEVENT 内のカラム source に保存されます。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

### 備考

preserve\_source\_format を On に設定すると、データベースサーバは、プロシージャ、ビュー、トリガ、イベントの CREATE 文と ALTER 文によってフォーマットされたソースを保存し、それを適切なシステムビューのソースカラムに配置します。

フォーマットされていないソーステキストは、同じシステムテーブルの proc\_defn、trigger\_defn、view\_defn の各カラムに格納されます。ただし、これらの定義は、SQL Central では簡単には読めません。フォーマットされたソースカラムでは、スペース、コメント、大文字または小文字を任意に選んで定義を参照できます。

このオプションを OFF にすると、データベースにオブジェクト定義を保存するために使用される領域を減らすことができます。

## 1.5.6.7.101 prevent\_article\_pkey\_update オプション [Mobile Link]

Mobile Link パブリケーションに関連するテーブルのプライマリキーカラムの更新を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションを On に設定すると、パブリケーションに含まれるテーブルのプライマリキーカラムの更新が禁止されます。このオプションを使用することで、特にレプリケーションと同期の環境におけるデータの整合性が確保されます。

#### 警告

同期環境やレプリケーション環境では、このオプションを Off に設定しないでください。

## 1.5.6.7.102 priority オプション

接続からの要求を実行する優先度レベルを設定します。

### 指定可能な値

Critical、High、Above Normal、Normal、Below Normal、Low、Background

### デフォルト

normal

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションの設定はすぐに有効になります。

このオプションには、max\_priority オプションの値より高い値は設定できません。

### 関連情報

[max\\_priority オプション \[730 ページ\]](#)

## 1.5.6.7.103 progress\_messages オプション

データベースサーバからクライアントに進行メッセージを送信するかどうかを制御します。

### 指定可能な値

#### Off

クライアントに進行メッセージを送信しません。

#### Raw

Raw を選択すると、進行メッセージに次のフォーマットが使用されます。

```
43;9728;22230;pages;5025;6138
```

Raw 進行メッセージには、セミコロンで区切られた 6 つのフィールドがあります。各フィールドの定義は、次のとおりです。

#### フィールド 1

実行済みの文の割合。

#### フィールド 2

完了したページ数、ロー数、またはバイト数。

#### フィールド 3

処理予定のページ数、ロー数、またはバイト数。

#### フィールド 4

処理中の内容: ページ、ロー、またはバイト。

#### フィールド 5

現在の経過時間 (ミリ秒で表示)。

#### フィールド 6

文の実行が完了するまでの残りの推定所要時間 (ミリ秒)。

#### Formatted

Formatted を選択すると、進行メッセージに次のフォーマットが使用されます。

```
43% (22230 ページの 9728) が 00:00:05 後に完了します。残り約 00:00:06 です。
```

Formatted 進行メッセージはローカライズされており、時間形式は HH:MM:SS です。100 KB 未満の単位はバイト単位、100 MB 未満の単位はキロバイト (KB) 単位、100 MB 以上の単位はメガバイト (MB) 単位でそれぞれ表示されません。

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

進行メッセージは、文の合計推定継続時間の 5% の間隔で送信されます。一般に、推定が完了すると、最初の進行メッセージが 10 秒以内に送信されます。以降の進行メッセージは 30 秒から 5 分の間隔で送信されます。完了した割合が前のメッセージで送信された値と同じ場合は、メッセージが最後に送信されてから 5 分以上経過するまで、更新された進行メッセージは送信されません。文の実行所要時間が 30 秒未満の場合には、進行メッセージは送信されません。

推定時間は継続的に再計算され、操作の進行に伴って推定残り時間の精度が高まります。バックアップなどのイベント時には、文の実行中に合計ページ数が調整されるため、完了した割合と推定残り時間が変わります。BACKUP...WITH CHECKPOINT COPY または UNLOAD SELECT などの文では、影響を受けるページまたはローの合計数が不明なため、完了した割合の値が 100% を超える可能性があります。その結果、推定残り時間は計算できず、進行メッセージには含まれません。

次の文とプロシージャでは、進行メッセージがサポートされています。

- BACKUP DATABASE (イメージとアーカイブの両方)
- LOAD TABLE (USING FILE と USING CLIENT FILE のみ)
- MESSAGE
- REORGANIZE TABLE
- RESTORE DATABASE
- UNLOAD (すべてのタイプ)
- sa\_table\_page\_usage システムプロシージャ

ユーティリティデータベースに接続されている場合は、SET OPTION 文を使用して progress\_messages オプションを設定できます。

また、**ツール** > **オプション** > **SQL Anywhere** > **コマンド** > **進行メッセージを表示** をクリックして、Interactive SQL で progress\_messages オプションを設定することもできます。**進行メッセージを表示** を有効にすると、progress\_messages オプションが Formatted に設定されます。

## 関連情報

[ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

## 1.5.6.7.104 qualify\_owners オプション [SQL Remote]

SQL Remote によってレプリケートされる SQL 文で、修飾されたオブジェクト名を使用するかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

### 備考

SQL Anywhere のインストール環境で修飾が必要でない場合は、オプションを Off にすると、メッセージが少し小さくなります。

## 1.5.6.7.105 query\_mem\_timeout オプション

要求にメモリが付与されるまでの最大待機時間をミリ秒で設定します。

### 指定可能な値

-1、0、正の整数

## デフォルト

-1

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションを -1 (デフォルト) または 0 より小さい値に設定すると、要求にメモリが付与されるまでの待機時間は、要求の推定実行時間より最大で 50 倍の長さになります。このオプションを 0 に設定すると、要求はメモリが付与されるまで永久に待機します。それ以外の場合、設定したミリ秒の値が要求にメモリが付与されるまでの最大待機時間になります。

## 関連情報

[キャッシュとメモリガバナ \[1371 ページ\]](#)

## 1.5.6.7.106 quote\_all\_identifiers オプション [SQL Remote]

SQL Remote によってレプリケートされる SQL 文で、引用符で囲んだ識別子を使用するかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

このオプションを Off に設定すると、dbremote が SQL Anywhere で引用符が必要な識別子を引用符で囲みます。

このオプションが On の場合、すべての識別子が引用符で囲まれます。

## 1.5.6.7.107 quoted\_identifier オプション

二重引用符内の文字列の解釈を制御します。

## 指定可能な値

On、Off

## デフォルト

On

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('Off') によって設定されます。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり

	PUBLIC ロール	現在のユーザ	他のユーザ
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、二重引用符内の文字列を、識別子 (On) とリテラル文字列 (Off) のどちらとして解釈するかを制御します。quoted\_identifier オプションは、Transact-SQL との互換性を保つために実装されています。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.108 read\_past\_deleted オプション

独立性レベル 1 または 2 でコミットされていない削除におけるデータベースサーバ動作を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり

	PUBLIC ロール	現在のユーザ	他のユーザ
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

read\_past\_deleted が On (デフォルト) の場合、独立性レベル 1 または 2 の逐次スキャンではコミットされていない削除ローを省略します。Off の場合、(削除トランザクションがコミットまたはロールバックするまで) 逐次スキャンは独立性レベル 1 または 2 のコミットされていない削除ローをブロックします。このオプションは、独立性レベル 1 と 2 のサーバ動作を変更します。

ほとんどの場合、このオプションは On のままにしてください。Off に設定すると、(使用可能なインデックスがある場合) ブロック動作はオプティマイザが選択したプランに左右されます。

## 1.5.6.7.109 recovery\_time オプション

データベースサーバがシステム障害からリカバリするのに要する最長時間を分単位で設定します。

### 指定可能な値

整数 (分)

### デフォルト

2

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

## 備考

このオプション設定の変更を有効にするには、データベースサーバを再起動する必要があります。

このオプションは、いつチェックポイントを実行すべきかを決定するために、`checkpoint_time` オプションと一緒に使用します。

データベースサーバはヒューリスティックを使用して、最後のチェックポイント以後に行った操作に基づいてリカバリ時間を予測します。この予測には、データベースのリカバリ予想時間とチェックポイント予想時間も含まれます。そのため、リカバリ時間は正確ではありません。

## 関連情報

[自動リカバリ処理 \[917 ページ\]](#)

[データベースサーバがチェックポイントのタイミングを決定する方法 \[293 ページ\]](#)

[checkpoint\\_time オプション \[667 ページ\]](#)

[-gr データベースサーバオプション \[428 ページ\]](#)

## 1.5.6.7.110 remote\_idle\_timeout オプション

Web サービスのクライアントプロシージャとファンクションで許容される休止時間 (秒数) を制御します。

### 指定可能な値

整数 (秒)

### デフォルト

15

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり

	PUBLIC ロール	現在のユーザ	他のユーザ
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、Web サービスのクライアントプロシージャとファンクションに影響します。アクティビティのない状態が指定の秒数を越えると、プロシージャまたはファンクションはタイムアウトになります。

## 1.5.6.7.111 replication\_error オプション [SQL Remote]

SQL エラーが発生したときに Message Agent で呼び出すストアードプロシージャを指定できます。

### 指定可能な値

ストアードプロシージャ名

### デフォルト

空の文字列

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

SQL Remote で replication\_error オプションを使用すると、SQL エラーが生じたときに Message Agent で呼び出すストアードプロシージャを指定できるようにします。デフォルトではプロシージャを呼び出しません。

プロシージャには、データ型が CHAR、VARCHAR、または LONG VARCHAR の引数を 1 つ指定してください。プロシージャは、SQL エラーメッセージで 1 回、エラーの原因となった SQL 文でもう 1 回呼び出されます。状況によっては (外部キー違反の場合など)、エラーの原因となった SQL 文が使用できないために、ストアードプロシージャを 1 回しか呼び出すことができない場合があります。

このオプションを使用すると、レプリケーションで SQL エラーの追跡とモニタができますが、その場合でもやはり、エラーが起こらないように設計する必要があります。このオプションは、そのようなエラーを解決するためのものではありません。

## 関連情報

[replication\\_error\\_piece オプション \[SQL Remote\] \[771 ページ\]](#)

### 1.5.6.7.112 replication\_error\_piece オプション [SQL Remote]

replication\_error オプションとともに機能し、SQL Remote のレプリケーション中に SQL エラーが発生した場合に Message Agent から呼び出される LONG VARCHAR ストアドプロシージャを指定できるようにします。

## 指定可能な値

ストアードプロシージャ名

デフォルト

空の文字列

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

エラーが発生し、`replication_error` が定義されている場合は、完全なエラー文字列とともに `replication_error` プロシージャが呼び出されます。

`replication_error` と `replication_error_piece` の両方が定義されている場合、エラーは VARCHAR 部に分割されます。`replication_error` が最初の VARCHAR 部とともに呼び出され、`replication_error_piece` が残りの VARCHAR 部とともに繰り返し呼び出されます。

## 関連情報

[replication\\_error オプション \[SQL Remote\] \[770 ページ\]](#)

### 1.5.6.7.113 request\_timeout オプション

1つの要求を実行できる最大時間を設定します。このオプションを使用すると、接続が大量のサーバリソースを長時間にわたって消費することを防止できます。

## 指定可能な値

整数 (秒)、範囲は 0 ~ 86400 (1 日)

## デフォルト

0

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションを 0 に設定すると、要求はタイムアウトになりません。

要求の実行時間が request\_timeout に設定された時間 (CPU 時間ではなく実際の時間) を超えた場合、その要求は中断され、ユーザにエラーが返されます。この場合、返されるエラーは `SQL_REQUEST_TIMEOUT` (タイムアウトになったため、要求が中断されました) です。要求がブロックされ、blocking\_timeout オプションが 0 に設定されている場合は、最長 request\_timeout 秒の間引き続き要求がブロックされる可能性があり、その後ブロックエラー (たとえば、`SQL_LOCKED` (ユーザ '%1' の '%2' 内のローがロックされています) など) が返されます。

1 ~ 14 の範囲の値は、ユーザオプションまたは PUBLIC オプションとしては指定できませんが、テンポラリーオプションとしては指定できます。これによって、接続に時間がかかる場合でも (ログインプロシージャが複雑であるなどの理由で)、ユーザがデータベースサーバに接続できなくなることはありません。

このオプションは、データベースクライアント要求と HTTP/HTTPS 要求のいずれに対しても使用できます。ただし、このオプションをストアプロシージャや HTTP/HTTPS 要求に設定しても、現在の要求には影響しません。要求の先頭にあるオプション値が使用されるからです。

request\_timeout パブリックオプションを設定する場合は注意が必要です。このオプションを設定すると、実行時間の長い要求を発行するアプリケーション (dbvalid、dbbackup、dbunload など) が失敗する可能性があります。また、大量のサーバリソースは使用しなくても別のユーザでブロックされる可能性のあるアプリケーションも、request\_timeout が設定されていると失敗することがあります。このような問題に対処するには、ログインプロシージャでは接続の APPINFO 値に基づいて特定のアプリケーションだけに request\_timeout オプションを設定します。

多数のローを含む結果セットをフェッチするなど、個々の要求を高速で評価するアプリケーションでは、このオプションを設定しても大量のサーバリソースの消費を回避できないことがあります。

## 関連情報

[blocking\\_timeout オプション \[664 ページ\]](#)

[AppInfo \(APP\) 接続パラメータ \[48 ページ\]](#)

## 1.5.6.7.114 reserved\_connections オプション

データベースが標準接続に予約する必要がある同時接続の最小数を制御します。このオプションは、データベースサーバが HTTP/HTTPS 接続を受け入れる場合に役立ちます。

### 指定可能な値

#### size

この整数は、データベースが標準接続に予約する必要がある同時接続の数を指定します。

ネットワークデータベースサーバで実行されているデータベースの場合、次の要件を満たす数を指定します。

- ライセンスにより許可されているデータベースサーバ接続の最大数より小さい。
- -gm データベースサーバオプションにより指定されているデータベースサーバ接続の最大数より小さい。
- max\_connections データベースオプションにより指定されているデータベースサーバ接続の最大数より小さい。

パーソナルデータベースサーバの場合、3 ~ 9 の数値を指定します。パーソナルデータベースサーバ (dbeng17) は、最大 10 の同時接続を受け入れ、そのうち 3 つは常に標準接続用に予約されています。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	いいえ	いいえ

## 備考

このオプションを設定すると、HTTP/HTTPS 接続がキューに入っている場合でも標準接続をデータベースが受け入れるようになります。

データベースサーバは、ライセンスの制限に到達するまで HTTP/HTTPS 接続を受け入れ、その後の HTTP/HTTPS 接続をキューに入れて、接続が利用可能になったときに処理します。接続試行がキューに入っている間は、HTTP/HTTPS 接続を標準接続に置き換えることはできません。標準接続の作成を待機しているユーザは、HTTP/HTTPS 接続キューが完了するまで無限に待機することになります。標準接続のみを受け入れ可能なデータベース接続の最小するを指定するには、reserved\_connections オプションを使用します。

パーソナルデータベースサーバは、標準データベース接続が HTTP 接続によりブロックされないように、少なくとも 3 つの接続を標準接続用に常に予約しています。

デフォルトでは、ネットワークデータベースサーバでは接続の予約は行われません。

reserved\_connections 接続プロパティの値を問い合わせることで、データベースの予約済み接続の現在の数を表示できます。

```
SELECT CONNECTION_PROPERTY ( 'reserved_connections' );
```

## 例

次の例では、標準接続用に 10 の接続を予約します。

```
SET OPTION PUBLIC.reserved_connections=10;
```

## 関連情報

[データベースおよびデータベースサーバに接続できる限界数 \[237 ページ\]](#)

[標準接続の予約 \[240 ページ\]](#)

[max\\_connections オプション \[725 ページ\]](#)

## 1.5.6.7.115 reserved\_keywords オプション

1つ以上のキーワードを予約キーワードに変更します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	いいえ	いいえ

### 備考

このオプションでは、指定されたキーワードから予約キーワードが作成されます。現在のところ、キーワード LIMIT のみを予約キーワードとして有効にできます。

デフォルトでは、LIMIT は予約キーワードではないため、SELECT 文で LIMIT 句を使用すると構文エラーが発生します。

```
SELECT * FROM Products
ORDER BY ID
LIMIT 5;
```

これは、たとえば CREATE TABLE 文で識別子として LIMIT を使用できることを示します。LIMIT 句のサポートを有効にするには、reserved\_keywords オプションを使用してキーワード LIMIT を予約キーワードとして有効にする必要があります。次の文がこの処理を行います。

```
SET OPTION PUBLIC.reserved_keywords = 'LIMIT';
```

予約キーワードとして有効になると、二重引用符、角括弧、逆引用符 (`...`) で囲まないと LIMIT を識別子として使用することはできません。

```
SELECT * FROM [LIMIT];
```

ある単語が単語のカンマ区切りリストで受け入れられるかどうかは、次のルールにより決まります。

- その単語が予約語候補のリストに表示されている必要があります。
- その単語が予約語 SET、OPTION、または OPTIONS のいずれかではあってはなりません。

予約語候補のリストは、sa\_reserved\_words システムプロシージャにより生成されます。"予約語" および "予約キーワード" という用語は、このマニュアルでは相互に置き換え可能な方法で使用されています。

個々の予約キーワードは、non\_keywords オプションを使用して無効にできます。

すでに予約キーワードとなっているキーワードを reserved\_keywords リストに追加しても、影響はありません。

すでに non\_keywords に追加されているキーワードを reserved\_keywords リストに追加しても、影響はありません。

このオプションで新規設定を行うと、前に設定しているものに置き換わります。

次の文は以前の設定内容をすべてクリアします。

```
SET OPTION PUBLIC.reserved_keywords =;
```

## 関連情報

[non\\_keywords オプション \[740 ページ\]](#)

### 1.5.6.7.116 return\_date\_time\_as\_string オプション

クエリが実行されたときに、DATE、TIME、または TIMESTAMP の値がクライアントアプリケーションに渡される方法を制御します。

## 指定可能な値

On、Off

## デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	いいえ	いいえ	不可。
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

## 備考

このオプションは、DATE、TIME、または TIMESTAMP の値をアプリケーションに返すときに、DATE、TIME、または TIMESTAMP のデータ型とするか文字列とするかを指定します。

このオプションを On に設定すると、date\_format、time\_format、または timestamp\_format オプションの設定を保持するために、データベースサーバは DATE、TIME、または TIMESTAMP の値を文字列に変換してからクライアントに送信します。

SQL Central と Interactive SQL では、return\_date\_time\_as\_string オプションは自動的に Off に設定されます。このオプションを Off に設定すると、Interactive SQL はクライアントコンピュータのロケール設定を使用して DATE、TIME、TIMESTAMP の値を表示します。

## 関連情報

[date\\_format オプション \[684 ページ\]](#)

[time\\_format オプション \[808 ページ\]](#)

[timestamp\\_format オプション \[811 ページ\]](#)

[timestamp\\_with\\_time\\_zone\\_format オプション \[813 ページ\]](#)

## 1.5.6.7.117 rollback\_on\_deadlock オプション

デッドロック発生時のトランザクションの処理方法を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションを On に設定すると、デッドロックが発生した場合、トランザクションは自動的にロールバックされます。ロールバックは、現在の要求が完了した後で発生します。このオプションを Off に設定した場合、データベースサーバはデッドロックが発生した文を自動的にロールバックし、そのトランザクションに対して、発生したデッドロックの種類を示すエラーメッセージを返します。文のロールバックでは、その文によって取得されたロックが解放される可能性はあまりありません。

### 関連情報

[ICU - International Components for Unicode](#) 

## 1.5.6.7.118 row\_counts オプション

データベースが開いているときに、クエリ内のローの数がデータベースによって常にカウントされるかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションを Off に設定すると、通常、ローの数は推定値のみとなります。On に設定すると、ローの正確な数がカウントされます。

#### 警告

row\_counts を On に設定すると、クエリの実行時間がかかなり長くなる場合があります。実際、通常はデータベースサーバでクエリが 2 回実行され、実行時間が 2 倍になります。

## 1.5.6.7.119 save\_remote\_passwords オプション [SQL Remote]

メッセージリンクに入力されたパスワードを保存します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

### 備考

メッセージリンクパラメータをデータベースではなく外部に保存する場合、パスワードが保存されないように設定できます。このオプションを Off に設定すると、パスワードは保存されません。

## 1.5.6.7.120 scale オプション

計算結果が最大精度でトランケートされる場合の、小数点以下の最小桁数を指定します。

### 指定可能な値

precision データベースオプションに指定した値より小さい 0 ~ 127 の整数

## デフォルト

6

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	いいえ	いいえ	いいえ

## 備考

精度は数値の合計桁数であり、小数点は含まれません。scale オプションは、計算結果が最大精度にトランケートされた場合の、小数点以下の最小桁数を指定します。たとえば、数値 3.1415926 の精度は 8、スケールは 7 です。

掛け算、割り算、足し算、引き算、集合関数はすべて結果が最大精度を超える可能性があります。

## 関連情報

[precision オプション \[756 ページ\]](#)

## 1.5.6.7.121 secure\_feature\_key オプション (廃止予定)

-sf データベースオプションを使用して保護された接続に対して機能を有効にできます。

## 指定可能な値

文字列

## デフォルト

空の文字列

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	×	×	×
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

データベースサーバの起動時に `-sf` データベースサーバオプションを含めると、サーバ上で実行中のデータベースで使用できない機能を指定できます。`-sk` サーバオプションを使用すると、接続に対して保護された (無効にされた) すべての機能を再び有効にするのに使用できるキーを指定でき、データベースサーバ上で実行中のすべてのデータベースに対して保護された機能を変更するためのパーミッションがその接続に付与されます。データベースサーバの起動時に、`secure_feature_key` テンポラリオプションの値を `-sk` オプションで指定されている値に設定すると、そのデータベース接続に対するすべての機能が再び有効化され、その接続では `sa_server_option` システムプロシージャを使用してデータベース機能へのアクセスを制御できます。

`secure_feature_key` オプションを、`-sk` データベースサーバオプションで指定された値以外の任意の値に設定した場合、エラーは生成されず、`-sf` データベースサーバオプションで指定された機能は接続に対して引き続き無効のままになります。

### 例

次のコマンドは、要求ログへのアクセスとすべてのリモートデータアクセス機能を無効にしてデータベースサーバ `secure_server` を起動します。これらの機能は、`-sk` で指定したキーを使用して、特定のデータベース接続において有効にできます。

```
dbsrv17 -n secure_server -sf request_log,remote -sk j978kls12 testdb.db
```

`secure_server` データベースサーバ上のデータベースに接続するときに `secure_feature_key` オプションを `-sk` で指定された値に設定すると、その接続において、要求ログへのアクセスとリモートデータアクセス機能が有効になります。

```
SET TEMPORARY OPTION secure_feature_key = 'j978kls12';
```

## 関連情報

[セキュリティ機能キーの作成 \[1607 ページ\]](#)

[-sk データベースサーバオプション \[479 ページ\]](#)

[-sf データベースサーバオプション \[471 ページ\]](#)

## 1.5.6.7.122 sort\_collation オプション

ORDER BY 式に対して SORTKEY 関数の暗黙の使用を許可します。

### 指定可能な値

Internal、collation-name、collation-id

### デフォルト

Internal

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションの値が Internal である場合、ORDER BY 句はそのまま扱われます。

このオプションの値を、有効な照合名または照合 ID に設定すると、ORDER BY 句の CHAR または NCHAR 文字列式は、SORTKEY 関数が呼び出されたものとして扱われます。BINARY、UUID、XML、VARBIT などのその他の文字列データ型を使用する文字列式は、変更されません。

#### 例

ソート照合をバイナリに設定します。

```
SET TEMPORARY OPTION sort_collation='binary';
```

ソート照合をバイナリに設定すると、以下のクエリに対して次のような変換が行われます。

```
SELECT Name, ID  
FROM Products  
ORDER BY Name, ID;  
SELECT name, ID
```

```
FROM Products
ORDER BY 1, 2;
```

クエリは以下のように変換されます。

```
SELECT Name, ID
FROM Products
ORDER BY SORTKEY(Name, 'binary'), ID;
```

## 1.5.6.7.123 sql\_flagger\_error\_level オプション

指定した規格にはない SQL への応答を制御します。

### 指定可能な値

- Off
- SQL:1992/Entry
- SQL:1992/Intermediate
- SQL:1992/Full
- SQL:1999/Core
- SQL:1999/Package
- SQL:2003/Core
- SQL:2003/Package
- SQL:2008/Core
- SQL:2008/Package
- UltraLite

### デフォルト

W

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり

	PUBLIC ロール	現在のユーザ	他のユーザ
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、指定した規格にはない SQL を警告として通知します。たとえば、SQL:2003/Package を指定すると、データベースサーバは、上級レベルの SQL/2003 構文ではない構文として通知します。

デフォルトの動作 (Off) では、エラー通知はオフに設定されます。

以前の SQL Anywhere バージョンとの互換性を持たせるため、次の値も受け入れられます。

### E

このオプションは、SQL:1992/Entry に対応します。

### I

このオプションは、SQL:1992/Intermediate に対応します。

### F

このオプションは、SQL:1992/Full に対応します。

### W

このオプションは、Off に対応します。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[sql\\_flagger\\_warning\\_level オプション \[785 ページ\]](#)

## 1.5.6.7.124 sql\_flagger\_warning\_level オプション

指定した規格にはない SQL への応答を制御します。

### 指定可能な値

- Off
- SQL:1992/Entry
- SQL:1992/Intermediate

- SQL:1992/Full
- SQL:1999/Core
- SQL:1999/Package
- SQL:2003/Core
- SQL:2003/Package
- SQL:2008/Core
- SQL:2008/Package
- UltraLite

## デフォルト

W

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションは、指定した規格にはない SQL を警告として通知します。たとえば、SQL:2003/Package を指定すると、データベースサーバは、上級レベルの SQL/2003 構文ではない構文として通知します。

デフォルトの動作 (Off) では、警告通知はオフに設定されます。

以前のバージョンとの互換性を持たせるため、次の値も受け入れられます。

**E**

このオプションは、SQL:1992/Entry に対応します。

**I**

このオプションは、SQL:1992/Intermediate に対応します。

**F**

このオプションは、SQL:1992/Full に対応します。

**W**

このオプションは、Off に対応します。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[sql\\_flagger\\_error\\_level オプション \[784 ページ\]](#)

### 1.5.6.7.125 sr\_date\_format オプション [SQL Remote]

データベースから取り出した DATE 値の形式を設定します。

#### 指定可能な値

文字列

#### デフォルト

yyyy/mm/dd

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

#### 備考

日付を保存するカラムをレプリケートするときに Message Agent が使用します。形式は次の記号を組み合わせた文字列です。

シンボル	説明
YY	2 桁の年
YYYY	4 桁の年

シンボル	説明
MM	2桁の月
MMM[m...]	月を略した文字 ("m" の数だけ文字を使用)
DD	2桁の日

各記号は、その形式を適用した日付のデータで置き換えられます。

文字データを表す記号 (MMM など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、すべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、すべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字がデータベースサーバにより選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'MMM' は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

このオプションは、次の記号で構成される文字列です。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。

## 関連情報

[sr\\_time\\_format オプション \[SQL Remote\] \[788 ページ\]](#)

[sr\\_timestamp\\_format オプション \[SQL Remote\] \[790 ページ\]](#)

[sr\\_timestamp\\_with\\_time\\_zone\\_format オプション \[SQL Remote\] \[792 ページ\]](#)

## 1.5.6.7.126 sr\_time\_format オプション [SQL Remote]

データベースから取り出した TIME 値の形式を設定します。

### 指定可能な値

文字列

## デフォルト

hh:nn:ss.Ssssss

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

時間を保存するカラムをレプリケートするときに Message Agent が使用します。形式は次の記号を組み合わせた文字列です。

シンボル	説明
HH	2 桁の時間 (24 時間表示)。
NN	2 桁の分。
MM	コロンの後の場合は、2 桁の分 (HH:MM など)。
SS[.ssssss]	秒と小数点第 6 位までの秒の小数。小数点以下 6 桁のタイムスタンプをサポートしないプラットフォームもあります。

各記号は、その形式を適用した時刻のデータで置き換えられます。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (NN や nn など) で入力すると、0 埋め込みが行われます。たとえば hh:nn:ss.sss と入力すると、02:34:56.123 と表記されます。
- 大文字と小文字を混ぜて入力すると (Nn など)、0 の埋め込みは行われません。たとえば Hh:nn:ss.sss と入力すると、2:34:56.123 と表記されます。
- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSssss)、後ろの 0 が削除されます。たとえば、hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

## 備考

表示形式の文字列で大文字と小文字を混在させて使用すると、数字の前にゼロが付きません。

## 関連情報

[sr\\_date\\_format オプション \[SQL Remote\] \[787 ページ\]](#)

[sr\\_timestamp\\_format オプション \[SQL Remote\] \[790 ページ\]](#)

[sr\\_timestamp\\_with\\_time\\_zone\\_format オプション \[SQL Remote\] \[792 ページ\]](#)

## 1.5.6.7.127 sr\_timestamp\_format オプション [SQL Remote]

データベースから取り出した TIMESTAMP 値の形式を設定します。

### 指定可能な値

文字列

### デフォルト

yyyy/mm/dd hh:nn:ss.Ssssss

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

### 備考

このオプションは、Message Agent で DATETIME と TIMESTAMP の情報をレプリケートするときに使用します。

形式は次の記号を組み合わせた文字列です。

シンボル	説明
YY	2 桁の年。

シンボル	説明
YYYY	4 桁の年。
MM	2 桁の月、または 2 桁の分 (HH:MM のように、コロンの後ろに続く場合)。
MMM[m...]	月を略した文字 ("m" の数だけ文字を使用)。
DD	2 桁の日。
DDD[d...]	曜日を示す略式文字。
HH	2 桁の時間。
NN	2 桁の分。
SS[.sssss]	秒と小数点第 6 位までの秒の小数。小数点以下 6 桁のタイムスタンプをサポートしないプラットフォームもあります。
AA	A.M. または P.M. (12 時間表記)。24 時間表記の場合、AA と PP は省略します。
PP	必要であれば PM (12 時間表記)。24 時間表記の場合、AA と PP は省略します。

各記号は、その形式を適用した日付のデータで置き換えられます。

文字データを表す記号 (MMM など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、すべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、すべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字がデータベースサーバにより選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'MMM' は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。
- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSsss)、後ろの 0 が削除されます。たとえば、hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

## 関連情報

[sr\\_date\\_format オプション \[SQL Remote\] \[787 ページ\]](#)

[sr\\_time\\_format オプション \[SQL Remote\] \[788 ページ\]](#)

[sr\\_timestamp\\_with\\_time\\_zone\\_format オプション \[SQL Remote\] \[792 ページ\]](#)

## 1.5.6.7.128 sr\_timestamp\_with\_time\_zone\_format オプション [SQL Remote]

データベースから取り出した TIMESTAMP WITH TIME ZONE 値の形式を設定します。

### 指定可能な値

文字列

### デフォルト

yyyy/mm/dd hh:nn:ss.Ssssss +hh:nn

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

### 備考

このオプションは、Message Agent で TIMESTAMP WITH TIME ZONE 情報をレプリケートするときに使用します。

形式は次の記号を組み合わせた文字列です。

シンボル	説明
YY	2 桁の年。
YYYY	4 桁の年。
MM	2 桁の月、または 2 桁の分 (HH:MM のように、コロンの後ろに続く場合)。
MMM[m...]	月を略した文字 ("m" の数だけ文字を使用)。
DD	2 桁の日。
DDD[d...]	曜日を示す略式文字。

シンボル	説明
HH	2桁の時間。
NN	2桁の分。
SS[.ssssss]	秒と小数点第6位までの秒の小数。小数点以下6桁のタイムスタンプをサポートしないプラットフォームもあります。
AA	A.M. または P.M. (12 時間表記)。24 時間表記の場合、AA と PP は省略します。
PP	必要であれば PM (12 時間表記)。24 時間表記の場合、AA と PP は省略します。

各記号は、その形式を適用した日付のデータで置き換えられます。

文字データを表す記号 (MMM など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、すべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、すべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字がデータベースサーバにより選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'MMM' は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。
- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSssss)、後ろの 0 が削除されます。たとえば、hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

## 関連情報

[sr\\_date\\_format オプション \[SQL Remote\] \[787 ページ\]](#)

[sr\\_time\\_format オプション \[SQL Remote\] \[788 ページ\]](#)

[sr\\_timestamp\\_format オプション \[SQL Remote\] \[790 ページ\]](#)

## 1.5.6.7.129 st\_geometry\_asbinary\_format オプション

ジオメトリからバイナリへの空間データ値の変換方法を制御します。

### 指定可能な値

このオプションでサポートされている値のリストは、ST\_AsBinary メソッドで説明されているパラメータ (サブパラメータも含む) のリストと同じです。

### デフォルト

WKB

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションは、ジオメトリからバイナリへの変換方法を決定するために使用します。次に、このオプションが使用される主なコンテキストを 2 つ示します。

- クライアントアプリケーションから値をバイナリとしてフェッチ。
- CAST 文の実行。例: `CAST ( geometry-expression AS LONG BINARY )`

クライアントアプリケーションからのフェッチと CAST 文以外のコンテキストでは、ジオメトリタイプから BINARY データ型への割り当てによりエラーになります。たとえば、BINARY パラメータを受け入れるファンクションまたはプロシージャを呼び出してジオメトリを渡すことはできず、ジオメトリを LONG BINARY 変数に割り当てすることもできません。

### 関連情報

[st\\_geometry\\_astext\\_format オプション \[795 ページ\]](#)

[st\\_geometry\\_asxml\\_format オプション \[796 ページ\]](#)

[st\\_geometry\\_describe\\_type オプション \[797 ページ\]](#)

[st\\_geometry\\_on\\_invalid オプション \[799 ページ\]](#)

## 1.5.6.7.130 st\_geometry\_astext\_format オプション

ジオメトリからテキストへの空間データ値の変換方法を制御します。

### 指定可能な値

このオプションでサポートされている値のリストは、ST\_AsText メソッドで説明されているパラメータ (サブパラメータも含む) のリストと同じです。

### デフォルト

WKT

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションは、ジオメトリからテキストへの変換方法を決定するために使用します。次に、このオプションが使用される主なコンテキストを 2 つ示します。

- クライアントアプリケーションから値をテキストとしてフェッチ。
- CAST 文の実行。例: `CAST ( geometry-expression AS LONG VARCHAR )`

クライアントアプリケーションからのフェッチと CAST 文以外のコンテキストでは、ジオメトリタイプから CHAR データ型への割り当てによりエラーになります。たとえば、CHAR パラメータを受け入れるファンクションまたはプロシージャを呼び出してジオメトリを渡すことはできず、ジオメトリを LONG VARCHAR 変数に割り当てすることもできません。

## 関連情報

[st\\_geometry\\_asbinary\\_format オプション \[794 ページ\]](#)

[st\\_geometry\\_asxml\\_format オプション \[796 ページ\]](#)

[st\\_geometry\\_describe\\_type オプション \[797 ページ\]](#)

[st\\_geometry\\_on\\_invalid オプション \[799 ページ\]](#)

## 1.5.6.7.131 st\_geometry\_asxml\_format オプション

ジオメトリから XML への空間データ値の変換方法を制御します。

### 指定可能な値

このオプションでサポートされている値のリストは、ST\_AsXML メソッドで説明されているパラメータ (サブパラメータも含む) のリストと同じです。

### デフォルト

GML

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

このオプションは、ジオメトリから XML への変換方法を決定するために使用します。次に、このオプションが使用される主なコンテキストを 2 つ示します。

- クライアントアプリケーションから値を XML としてフェッチ。
- CAST 文の実行。例: `CAST ( geometry-expression AS XML )`

クライアントアプリケーションからのフェッチと CAST 文以外のコンテキストでは、ジオメトリタイプから XML への割り当てによりエラーになります。たとえば、XML パラメータを受け入れるファンクションまたはプロシージャを呼び出してジオメトリを渡すことはできず、ジオメトリを XML 変数に割り当てることもできません。

## 関連情報

[st\\_geometry\\_asbinary\\_format オプション \[794 ページ\]](#)

[st\\_geometry\\_astext\\_format オプション \[795 ページ\]](#)

[st\\_geometry\\_describe\\_type オプション \[797 ページ\]](#)

[st\\_geometry\\_on\\_invalid オプション \[799 ページ\]](#)

## 1.5.6.7.132 st\_geometry\_describe\_type オプション

空間データの記述方法を制御します。

### 指定可能な値

- CHAR
- NCHAR
- BINARY

### デフォルト

CHAR

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

クライアントライブラリでは空間データ型が直接サポートされないため、空間値は文字またはバイナリとしてフェッチされます。`st_geometry_describe_type` は、ジオメトリカラムまたはジオメトリ式をクライアントに対して記述する方法を指定します。多くのクライアントアプリケーションでは、カラムまたは式の記述タイプを使用して、カラムまたは式のフェッチに使用するクライアントデータ型が決まります。一部のクライアントアプリケーションでは記述タイプが無視されるため、`st_geometry_describe_type` を設定しても期待する結果にならないことがあります。

ジオメトリを文字列型としてフェッチする場合は、`st_geometry_astext_format` オプションを使用してジオメトリからテキストへの変換方法が決まります。バイナリ型としてフェッチする場合は、`st_geometry_asbinary_format` オプションが使用されます。CAST 文の場合にも同じルールが使用されます。

ODBC アプリケーションでは、どのジオメトリ型のカラムについても、`SQLColAttribute` の `SQL_DESC_TYPE_NAME` と `SQL_DESC_LOCAL_TYPE_NAME` フィールド識別子から文字列 `st_geometry` が返されます。JDBC では、`ResultSetMetaData.getColumnTypeName` メソッドによってこれが表現されます。

## 関連情報

[st\\_geometry\\_asbinary\\_format オプション \[794 ページ\]](#)

[st\\_geometry\\_astext\\_format オプション \[795 ページ\]](#)

[st\\_geometry\\_asxml\\_format オプション \[796 ページ\]](#)

[st\\_geometry\\_on\\_invalid オプション \[799 ページ\]](#)

## 1.5.6.7.133 st\_geometry\_interpolation オプション

空間計算が円弧を含むときに内挿法を制御します。

### 指定可能な値

このデータベースオプションは、セミコロンで区切られた `key=value` ペアのセット (`key1=value1;key2=value2;...`) になります。許可されているキーは、`relative-tolerance-percent` と `absolute-tolerance` です。

#### **relative-tolerance-percent=DOUBLE**

`relative-tolerance-percent` は、最大の変動をパーセントで表記して指定するために使用します。これは、`ST_CircularString` セグメントの半径に対して、`ST_CircularString` セグメントの中心点から内挿された `ST_LineString` セグメントまでの距離で許可されています。`relative-tolerance-percent` に許可される最小値は 0 であり、`absolute-tolerance` 設定が常に使用されることを意味します。

#### **absolute-tolerance=DOUBLE**

`absolute-tolerance` は、最大の絶対変動を線形測定単位で表記して指定するために使用します。これは、`ST_CircularString` セグメントの中心点から内挿された `ST_LineString` セグメントまでの距離で許可されています。`absolute-tolerance` に許可される最小値は、SRID で指定される許容値でもあります。

## デフォルト値

どちらのキーも指定されない場合、relative-tolerance-percent のデフォルト値は 0.3 になり、これは、完全な円に 40 の点を提供します。また、absolute-tolerance のデフォルト値は SRID で指定される許容値になります。

1つのキーだけが指定される場合、もう1方のキーはその最小値に設定されます。

両方のキーが指定される場合、点の最小数はすべての制限を満たすかたちで生成されます。

## デフォルト

relative-tolerance-percent=.3

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このデータベースオプションの設定に関係なく、曲線セグメントごとに少なくとも3つの点があります。さらに、内挿された ST\_LineString の連続する2つの点が互いの許容値内に存在することはありません。

ST\_LineString の ST\_Envelope がオリジナルの ST\_CircularString の ST\_Envelope と同じになるように、さらに点を追加することもできます。

## 1.5.6.7.134 st\_geometry\_on\_invalid オプション

ジオメトリが基本的な検証に失敗した場合 (1ポイントの LineString、リングが閉じていない多角形、空間参照系の境界を越えたジオメトリなど) の動作を制御します。

## 指定可能な値

Ignore

ジオメトリが表面検証に失敗しても何も処理を行わず、操作を続行します。

#### Error

ジオメトリが表面検証に失敗するとエラーを返し、操作が失敗します。

## デフォルト

Error

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

無効な空間データをロードする必要がある場合は、オプション値を ignore に設定してから無効なデータをロードします。データをロードしたら、ST\_IsSimple や ST\_IsValue などのメソッドを使用して無効なローを判断したり、ST\_AsText を使用してデータの WKT 表現を取得したりします。他のほとんどの空間メソッドでは、無効なジオメトリを渡すとエラーになります。テキストエディタを使用して、無効なジオメトリを WKT で修正し、データベースでジオメトリを更新します。

## 関連情報

[st\\_geometry\\_asbinary\\_format オプション \[794 ページ\]](#)

[st\\_geometry\\_astext\\_format オプション \[795 ページ\]](#)

[st\\_geometry\\_asxml\\_format オプション \[796 ページ\]](#)

[st\\_geometry\\_describe\\_type オプション \[797 ページ\]](#)

## 1.5.6.7.135 string\_rtruncation オプション

文字列がトランケートされた場合にエラーを発生させるかどうかを決定します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	いいえ

### 備考

トランケートされた文字またはバイナリデータ (x20) がスペースのみで構成されている場合、例外は発生しません。On に設定すると、ANSI/ISO SQL 標準の動作になります。Off に設定すると、例外は発生せず、文字列は何も通知されずにトランケートされます。

文字列のトランケートは、さまざまな場所で発生する可能性があります。たとえば、INSERT、UPDATE、CAST 文を使用したり、変数に割り当てたりすると、宣言したデータ型が短すぎた場合、文字列がトランケートされる場合があります。トランケートは文字とバイナリデータの両方で発生する可能性があります。

### 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.136 subscribe\_by\_remote オプション [SQL Remote]

SUBSCRIBE BY 値が NULL または空の文字列である場合の解釈を制御します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

### 備考

このオプションを On に設定すると、NULL または空の文字列である SUBSCRIBE BY 値のあるローに対してリモートデータベースから操作が行われた場合、リモートユーザがローにサブスクリプションを作成したと見なします。Off に設定すると、リモートユーザはローにサブスクリプションを作成していないと見なされます。

このオプションのただ 1 つの制限として、リモートユーザが実際に、NULL または空のサブスクリプション式 (統合データベースでのみ保持される情報を表す) を使用してローの INSERT (または UPDATE) を必要としている場合は、エラーとなります。この問題はそれほど深刻ではなく、インストール環境で、リモートユーザに属さないサブスクリプション値を割り当てることで対処できます。

## 1.5.6.7.137 subsume\_row\_locks オプション

データベースサーバがテーブル用に個別のローロックをいつ取得するかを制御します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

subsume\_row\_locks オプションが On (デフォルト) の場合、LOCK TABLE *t* IN EXCLUSIVE MODE で排他的にテーブル *t* がロックされるたびに、データベースサーバは *t* 用に個別のローロックを取得しなくなります。

これにより、単一のトランザクションで広範な更新が *t* に対して実行された場合 (特に *t* がキャッシュサイズに比べて大きい場合)、パフォーマンスが大幅に改善されます。また、ロックテーブルで現在処理可能な量より多くのアトミック更新オペレーションを実行できるようになります (約 200 万 ~ 400 万ロー)。

このオプションを On に設定すると、データベース内のいずれかのローが変更された場合に、この方法でロックされたテーブルの上にあるキーセットカーソルから、カーソル内のすべてのローに対してロー変更の警告が返されます。データベースサーバは ORDER BY のある更新可能なカーソルを、結果としてキーセットカーソルにすることができます。

## 1.5.6.7.138 suppress\_tds\_debugging オプション

TDS デバッグ情報をデータベースサーバメッセージウィンドウに表示するかどうかを決定します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

データベースサーバを -z オプションで起動すると、TDS プロトコルに関するデバッグ情報を含むデバッグ情報がデータベースサーバメッセージウィンドウに表示されます。

suppress\_tds\_debugging オプションは、TDS に関するデバッグ情報がデータベースサーバメッセージウィンドウに表示されないようにします。このオプションを Off (デフォルト) に設定すると、TDS デバッグ情報がデータベースサーバメッセージウィンドウに表示されます。

### 関連情報

[-z データベースサーバオプション \[509 ページ\]](#)

## 1.5.6.7.139 synchronize\_mirror\_on\_commit オプション

非同期モードまたは非同期フルページモードにおいて、データベースの変更がミラーサーバへ送信されたことがどの時点で保証されるかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

synchronize\_mirror\_on\_commit オプションでは、非同期モードまたは非同期フルページモードにおいて、データベースの変更がミラーサーバへ送信されたことがどの時点で保証されるかについて厳密に制御できます。このオプションのデフォルトは Off です。On に設定すると、COMMIT が発生するたびにトランザクションログに記録されたすべての変更がミラーサーバへ送信され、それらの変更がミラーサーバで受信されると、ミラーサーバからプライマリサーバへ受信確認が送信されます。

このオプションは、SET TEMPORARY OPTION を使用して特定のトランザクションに対して設定できます。

また、ログインプロシージャで APPINFO 文字列を調べ、このオプションを特定のアプリケーションだけに設定すると、効果的な場合もあります。これによって、ミラーリング動作をさまざまなアプリケーションのニーズに適合させることができます。

### 関連情報

[データベースミラーリング \[1660 ページ\]](#)

## 1.5.6.7.140 tds\_empty\_string\_is\_null オプション

TDS 接続で空の文字を NULL として返すか、ブランク文字 1 文字を含む文字列として返すかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

デフォルトでは、このオプションには Off が設定され、空の文字列は、TDS 接続用の 1 文字のブランク文字を含む文字列で返されます。このオプションを On に設定すると、空の文字列は、TDS 接続では NULL 文字列として返されます。TDS 以外の接続では、空の文字列と NULL 文字列は区別されます。

## 1.5.6.7.141 temp\_space\_limit\_check オプション

接続によって使用されるテンポラリファイル領域のサイズを確認し、要求された領域サイズが接続に許容されるサイズよりも大きい場合は要求は失敗になります。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

### 備考

temp\_space\_limit\_check オプションを On に設定すると、接続がそのクォータを超えるテンポラリファイル領域を要求した場合、その要求は失敗し、エラー SQLSTATE\_TEMP\_SPACE\_LIMIT が返されます。このオプションを Off (デフォルト) に設定すると、データベースサーバは接続が使用するテンポラリファイル領域のサイズをチェックしません。接続がそのクォータを超えるテンポラリ領域を要求した場合、このオプションが Off に設定されていると、致命的なエラーが発生することがあります。

次の 2 つのしきい値のうち小さい方が、接続のテンポラリファイルのクォータになります。

1. max\_temp\_space オプションで指定された各接続に許可されるテンポラリファイル領域の最大サイズ
2. テンポラリファイルのサイズを接続数で除算した見込みの最大サイズ

しきい値は、テンポラリファイルのサイズが最大サイズの 80% に達した場合にのみ使用されます。これは、オペレーティングシステムによって報告される、デバイス上に残っている空き領域のサイズに基づいて判断されます。接続要求によって、クォータが許可するサイズよりも大きいテンポラリファイルが要求された場合、接続の現在の要求は SQLSTATE '54W05' (SQLSTATE\_TEMP\_SPACE\_LIMIT) で失敗します。

接続が使用するテンポラリファイル領域の最大サイズは、max\_temp\_space オプションで指定できます。

## 関連情報

[データベースのサイズおよび数に関する物理的な制限 \[891 ページ\]](#)

[max\\_temp\\_space オプション \[734 ページ\]](#)

[-dt データベースサーバオプション \[400 ページ\]](#)

## 1.5.6.7.142 time\_format オプション

データベースから取り出した時刻の表示形式を設定します。

### 指定可能な値

文字列

### デフォルト

HH:NN:SS.SSS

現在のユーザのテンポラリ設定が ODBC ドライバと JDBC ドライバ ('hh:nn:ss')、および OLE DB ドライバ ('hh:nn:ss.sssss') によって設定されます。

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('HH:NN:SS.SSS') によって設定されます。

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

形式は次の記号を組み合わせた文字列です。

シンボル	説明
HH	2桁の時間。
NN	2桁の分。
SS[.sssss]	秒と小数点第6位までの秒の小数。小数点以下6桁のタイムスタンプをサポートしないプラットフォームもあります。
AA	AM または PM (12 時間表記)。24 時間表記の場合、AA と PP は省略します。
PP	必要であれば PM (12 時間表記)。24 時間表記の場合、AA と PP は省略します。

各記号は、その形式を適用した時刻のデータで置き換えられます。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (NN や nn など) で入力すると、0 埋め込みが行われます。たとえば hh:nn:ss.sss と入力すると、02:34:56.123 と表記されます。
- 大文字と小文字を混ぜて入力すると (Nn など)、0 の埋め込みは行われません。たとえば Hh:nn:ss.sss と入力すると、2:34:56.123 と表記されます。
- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSsss)、後ろの 0 が削除されます。たとえば、hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[date\\_format オプション \[684 ページ\]](#)

[timestamp\\_format オプション \[811 ページ\]](#)

[timestamp\\_with\\_time\\_zone\\_format オプション \[813 ページ\]](#)

## 1.5.6.7.143 time\_zone\_adjustment オプション

1つの接続でタイムゾーンの調整を修正できます。

### 指定可能な値

符号なし整数 (例: 300)

引用符で囲まれた正または負の符号付き整数 (例: '+300' または '-300')

先頭が + または - で、引用符で囲まれた時と分を表す文字列 (例: '+5:00'、'-5:00')

## デフォルト

0

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	×	×	×
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

time\_zone\_adjustment オプションの値は `SELECT CONNECTION_PROPERTY( 'TimeZoneAdjustment' )` が返す値と同じです。この値は、接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある分数を表します。

## 1.5.6.7.144 time\_zone オプション

データベースでタイムゾーンの計算に使用するタイムゾーンを指定します。

### 指定可能な値

CREATE TIME ZONE 文を使用して作成された既存のタイムゾーンを表す文字列。

符号なし整数 (例: 300)。

引用符で囲まれた正または負の符号付き整数 (例: '+300' または '-300')。

先頭が + または - で、引用符で囲まれた時刻 (時と分) を表す文字列 (例: '+5:00'、'-5:00')。

## デフォルト

空の文字列

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、MANAGE TIME ZONE あり	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、MANAGE TIME ZONE あり	いいえ	いいえ

## 備考

このオプションを設定すると、指定したタイムゾーンで実行されているかのようにデータベースが動作します。このデータベースオプションは、CURRENT TIME 特別値を使用する計算や、NOW または GETDATE 関数に影響しますが、CURRENT UTC TIME の値や CURRENT SERVER TIME 特別値には影響しません。

データベースに対してこのオプションをオフセット (分、時間:分の数字) に設定し、固有のオフセットを持ち、夏時間のない、名前なしのタイムゾーンを使用することもできます。

### i 注記

タイムゾーンを示す文字列を指定するには、ISYSTIMEZONE テーブルにそのタイムゾーンを移植します。ISYSTIMEZONE を移植するには、CREATE TIME ZONE 文または ALTER TIME ZONE 文のいずれかを実行します。

## 1.5.6.7.145 timestamp\_format オプション

データベースから取り出したタイムスタンプの形式を設定します。

### 指定可能な値

文字列

### デフォルト

YYYY-MM-DD HH:NN:SS.SSS

現在のユーザのテンポラリ設定が ODBC ドライバ、JDBC ドライバ、OLE DB ドライバによって設定されます ('yyyy-mm-dd hh:nn:ss.sssss')。

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続によって設定されます ('YYYY-MM-DD HH:NN:SS.SSS')。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

形式は次の記号を組み合わせた文字列です。

シンボル	説明
YY	2桁の年。
YYYY	4桁の年。
MM	2桁の月、または2桁の分 (HH:MM のように、コロンの後ろに続く場合)。
MMM[m...]	月を略した文字 ("m" の数だけ文字を使用)。
D	曜日を示す1桁の数字 (1 = 日曜、7 = 土曜)。
DD	2桁の日。
DDD[d...]	曜日を示す略式文字。
JJJ	指定年の日 (1 ~ 366)。
HH	2桁の時間。
NN	2桁の分。
SS[.ssssss]	秒と小数点第6位までの秒の小数。小数点以下6桁のタイムスタンプをサポートしないプラットフォームもあります。
AA	AM または PM (12 時間表記)。24 時間表記の場合、AA と PP は省略します。
PP	必要であれば PM (12 時間表記)。24 時間表記の場合、AA と PP は省略します。

各記号は、その形式を適用した日付のデータで置き換えられます。

文字データを表す記号 (MMM など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、すべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。
- 記号をすべて小文字で入力すると、すべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字がデータベースサーバにより選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば 'mmm' は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。
- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSsss)、後ろの 0 が削除されます。たとえば、hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

#### i 注記

日付形式の順序が変更されるように timestamp\_format の設定を変更する場合は、同じ変更が反映されるように date\_order オプションも変更し、同様に date\_order を変更する場合は timestamp\_format も変更してください。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

[date\\_format オプション \[684 ページ\]](#)

[date\\_order オプション \[686 ページ\]](#)

[time\\_format オプション \[808 ページ\]](#)

[timestamp\\_with\\_time\\_zone\\_format オプション \[813 ページ\]](#)

## 1.5.6.7.146 timestamp\_with\_time\_zone\_format オプション

データベースから取り出した TIMESTAMP WITH TIME ZONE 値の形式を設定します。

### 指定可能な値

文字列

### デフォルト

YYYY-MM-DD HH:NN:SS.SSS+HH:NN

現在のユーザのテンポラリ設定が ODBC ドライバ、JDBC ドライバ、OLE DB ドライバによって設定されます ('yyyy-mm-dd hh:nn:ss.ssssss +hh:nn')。

現在のユーザのテンポラリ設定が .NET プロバイダによって設定されます ('YYYY-MM-DD HH:NN:SS.SSSSSS +HH:NN')。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

形式は次の記号を組み合わせた文字列です。

シンボル	説明
YY	2桁の年。
YYYY	4桁の年。
MM	2桁の月、または2桁の分 (HH:MM のように、コロンの後ろに続く場合)。
MMM[m...]	月を略した文字 - "m" の数だけ文字を使用。
D	曜日を示す1桁の数字 (1 = 日曜、7 = 土曜)。
DD	2桁の日。
DDD[d...]	曜日を示す略式文字。
JJJ	指定年の日 (1 ~ 366)。
HH	2桁の時間、または '+' の後にある場合はタイムゾーンオフセットの2桁の時間。
NN	2桁の分、または '+' の後にある場合はタイムゾーンオフセットの2桁の分。
SS[.sssss]	秒と小数点第6位までの秒の小数。小数点以下6桁のタイムスタンプをサポートしないプラットフォームもあります。
AA	AM または PM (12 時間表記)。24 時間表記の場合、AA と PP は省略します。
PP	必要であれば PM (12 時間表記)。24 時間表記の場合、AA と PP は省略します。
HH	2桁の時間 (タイムゾーンオフセット)。
NN	2桁の分 (タイムゾーンオフセット)。

各記号は、その形式を適用した日付のデータで置き換えられます。

文字データを表す記号 (MMM など) では、出力の文字を次のように制御できます。

- 記号をすべて大文字で入力すると、すべて大文字で表記されます。たとえば、MMM と入力すると、JAN と表記されます。

- 記号をすべて小文字で入力すると、すべて小文字で表記されます。たとえば、mmm と入力すると、jan と表記されます。
- 大文字と小文字を混ぜて入力すると、使用される言語に適切な文字がデータベースサーバにより選択されます。たとえば、Mmm と入力すると、英語では May、フランス語では mai と表記されます。

文字データがマルチバイトである場合、各記号の長さはバイト数ではなく文字数を表します。たとえば、mmm は 3 文字の月名を示しています。

数値データを表す記号では、記号に大文字を使用するか小文字を使用するかで 0 埋め込みを制御できます。

- 記号をすべて大文字または小文字 (MM や mm など) で入力すると、0 埋め込みが行われます。たとえば、yyyy/mm/dd と入力すると、2002/01/01 と表記されます。
- 大文字と小文字を混ぜて入力すると (Mm など)、0 埋め込みは行われません。たとえば、yyyy/Mm/Dd と入力すると、2002/1/1 と表記されます。
- 小数点以下の秒の最初の 2 桁の大文字と小文字を混ぜて入力すると (Ss または sSsss)、後ろの 0 が削除されます。たとえば、hh:nn:ss.Sss と入力すると、12:34:56.1 と表記されます。

#### i 注記

日付形式の順序が変更されるように timestamp\_with\_time\_zone\_format オプションの設定を変更する場合は、同じ変更が反映されるように date\_order オプションも変更し、同様に date\_order を変更する場合は timestamp\_with\_time\_zone\_format も変更してください。

## 関連情報

[timestamp\\_format オプション \[811 ページ\]](#)

[date\\_format オプション \[684 ページ\]](#)

[date\\_order オプション \[686 ページ\]](#)

[time\\_format オプション \[808 ページ\]](#)

## 1.5.6.7.147 truncate\_timestamp\_values オプション

TIMESTAMP 値の精度を制限します。

### 指定可能な値

On、Off

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

## 備考

SQL Anywhere の TIMESTAMP 値の精度は、小数点以下 6 桁です。ただし、TIMESTAMP 値を小数点以下 3 桁などにトランケートする他のソフトウェアとの互換性を維持するために、truncate\_timestamp\_values オプションを On に設定すると、データベースサーバが格納する小数点以下の桁数を制限できます。default\_timestamp\_increment オプションは、TIMESTAMP 値を小数点以下何桁でトランケートするかを決定します。

このオプションは、タイムスタンプデータがすでに格納されているデータベースに対しては有効にしないでください。

Mobile Link 同期に対し、このオプションを設定する場合は、最初の同期の実行前に設定する必要があります。

データベースサーバが truncate\_timestamp\_values と default\_timestamp\_increment の組み合わせで指定されたものより細かい TIMESTAMP 値を検出した場合、エラーがレポートされます。

通常、データベースをアンロードしてそれを truncate\_timestamp\_values 値と default\_timestamp\_increment 値が設定されている新しいデータベースに再ロードする方法が、確実に適切な TIMESTAMP 値を使用する一番簡単なソリューションです。ただし、テーブル内の TIMESTAMP カラムのタイプにより、次のように行うこともできます。

- TIMESTAMP カラムが DEFAULT TIMESTAMP または DEFAULT UTC TIMESTAMP で定義されている場合 (つまり、ローが変更されるとデータベースサーバによって値が自動的に更新される)、truncate\_timestamp\_values オプションを変更する前にテーブル内のすべてのローを削除する必要があります。ローは DELETE 文または TRUNCATE TABLE 文を使用して削除します。
- TIMESTAMP カラムが DEFAULT TIMESTAMP または DEFAULT UTC TIMESTAMP 以外で定義されている場合、UPDATE 文を実行して文字列に値を割り当て、TIMESTAMP に戻します。次に例を示します。

```
UPDATE T
  SET ts = CAST( DATEFORMAT( ts, 'yyyy/mm/dd hh:nn:ss.ss' )
  AS TIMESTAMP );
```

この手順によって、必要とされる精度より低下する可能性があります。使用するフォーマット文字列は、保持する精度の桁数によります。

### 例

たとえば、default\_timestamp\_increment オプションを 100000 に設定すると、秒の部分は小数点以下 1 桁の後でトランケートされるので、"2000/12/05 10:50:53.700" のような値を格納できるようになります。

## 関連情報

[default\\_timestamp\\_increment オプション \[692 ページ\]](#)

### 1.5.6.7.148 trusted\_certificates\_file オプション

データベースサーバが LDAP サーバのクライアントとして機能するときに、信頼できる認証局の証明書のリストを含むファイルを指定します。

#### 指定可能な値

文字列

デフォルト

空の文字列

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	いいえ	いいえ
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	いいえ	いいえ

#### 備考

このオプションは、データベースサーバがセキュアな Web プロシージャのクライアントである場合は適用されません。

データベースサーバへの受信 TLS 接続では、-ec サーバオプションで設定された証明書設定が使用されます。データベースサーバが別のサーバに対するクライアントとして機能するためには (たとえばデータベースサーバから LDAP サーバに接続する場合など)、TLS 証明書に署名した認証局 (CA) を認識させる必要があります。

クライアント接続の場合は、trusted\_certificates 接続オプションに、信頼できる CA 証明書のリストを含むファイルへのパスを設定します。同様に、データベースサーバがクライアントとして機能する場合、trusted\_certificates\_file データベースオプションでは、信頼された認証局証明書を指定します。

このオプションを \* に設定するか、空にすると、LDAP URL が `ldaps://` で始まる場合または LDAP サーバが TLS プロトコルを使用するように設定されている場合、ソフトウェアではオペレーティングシステムの証明書ストアにある証明書が使用されます。

このオプションを NULL に設定した場合は、信頼できる認証局が存在しないため、送信 TLS 接続を開始できません。

#### 例

この例では、サーバ証明書に署名する、信頼できる認証局のリストが、`trusted.txt` という名前のローカルファイルにあります。

```
SET OPTION PUBLIC.trusted_certificates_file = 'C:¥¥certificates¥¥shared¥¥trusted.crt';
```

次の 2 つの例では、サーバ証明書に署名する、信頼できる認証局のリストがオペレーティングシステムの証明書ストアにあります。

```
SET OPTION PUBLIC.trusted_certificates_file = '*';
```

```
SET OPTION PUBLIC.trusted_certificates_file = '';
```

## 関連情報

[-ec データベースサーバオプション \[402 ページ\]](#)

[trusted\\_certificates プロトコルオプション \[234 ページ\]](#)

## 1.5.6.7.149 tsql\_outer\_joins オプション

Transact-SQL 外部ジョイン演算子の \*= と =\* を文やビューで使用可能にするかどうかを指定します。

### 指定可能な値

On、Off

### デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

Transact-SQL 外部ジョインはサポートされなくなりました。このオプションを On に設定すると、Transact-SQL 外部ジョインを使用できます。

## 関連情報

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.150 tsql\_variables オプション

◎ 符号を、Embedded SQL ホスト変数名のプレフィクスとして使用できるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

現在のユーザのテンポラリ設定が SAP Open Client および jConnect TDS 接続 ('On') によって設定されます。

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションを On に設定すると、Embedded SQL のホスト変数名のプレフィクスとしてコロンの代わりに @ 符号を使用できます。これは、主に Transact-SQL との互換性を保つために実装されています。

## 関連情報

[Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

[Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

## 1.5.6.7.151 updatable\_statement\_isolation オプション

isolation\_level オプションが Readonly-statement-snapshot に設定されている場合に更新可能な文に適用する独立性レベルを指定します。

### 指定可能な値

0、1、2、3

### デフォルト

0

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

updatable\_statement\_isolation オプションで指定した独立性レベルは、isolation\_level オプションが Readonly-statement-snapshot に設定されている場合に、更新可能な文によって使用されます。次の値を指定できます。

0

ダーティリード、繰り返し不可能読み出し、幻ローを許可します。

1

ダーティリードを防ぎます。繰り返し不可能読み出しと幻ローを許可します。

2

ダーティリードと繰り返し不可能読み出しを防ぎます。幻ローを許可します。

3

直列化可能。ダーティリード、繰り返し不可能読み出し、幻ローを防ぎます。

## 関連情報

[isolation\\_level オプション \[711 ページ\]](#)

## 1.5.6.7.152 update\_statistics オプション

接続から統計ガバナーにクエリのフィードバックを送信できるかどうかを制御します。

## 指定可能な値

On、Off

## デフォルト

On

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

データベースサーバは通常のクエリ実行中に統計情報を収集し、収集した統計を使用してカラム統計を自己チューニングします。このオプションを Off に設定すると、統計ガバナーは統計の正常性に関するフィードバックを受信せず、指定された接続の統計も修正しません。ただし、統計の使用状況のモニタと、使用状況に基づいた統計の作成または削除は引き続き行われます。

SET ANY PUBLIC OPTION システム権限を持つユーザがこのオプションを設定した場合は、設定内容がすべてのデータベース接続に影響します。そうでない場合は、現在のユーザの接続のみが影響を受けます。

通常的环境において、このオプションをオフにする必要はありません。

update\_statistics オプションは、データの更新 (LOAD/INSERT/UPDATE/DELETE) による統計の変更に影響しません。これらの文に基づいて統計を更新するかどうかを設定するには、collect\_statistics\_on\_dml\_updates データベースオプションを使用します。

collect\_statistics\_on\_dml\_updates オプションと update\_statistics オプションが異なる点は、update\_statistics オプションを On に設定した場合、ある述部を満たすローの実際の数とその述部を満たすと予想されるローの数が比較されてから、推定値が更新されることです。collect\_statistics\_on\_dml\_updates オプションを On に設定した場合は、挿入、更新、または削除されたローの値に基づいてカラム統計が修正されます。

## 関連情報

[統計ガバナが統計を保守する方法 \[1379 ページ\]](#)

[collect\\_statistics\\_on\\_dml\\_updates オプション \[671 ページ\]](#)

## 1.5.6.7.153 user\_estimates オプション

クエリの述部に含まれるユーザ選択性推定をクエリオプティマイザが尊重するか無視するかを制御します。

### 指定可能な値

Enabled、Disabled、Override-Magic

### デフォルト

Override-Magic

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

ユーザ選択性推定を指定することで、データベースサーバが述部の選択性を正確に推定できないときにオプティマイザのパフォーマンスを向上させることができます。ただし、ユーザ選択性推定は、適切な状況でのみ使用してください。たとえば、オプティマイザが使用する Override-Magic 選択性推定と実際の選択性が大きく異なっている場合は、1 つ以上の関数に関係する述部に対して選択性推定を指定することは有用です。

ソフトウェアによって選択されたアクセスプランが不適切であり、パフォーマンス問題を回避するために選択性推定を使用したものの、それが不正確であった場合は、このオプションを Disabled に設定してください。不正確な推定が使用された場合、データベースサーバは最適なプランを選択できません。

ユーザ選択性推定に述部が指定されているときは、このオプションの設定に基づいて、その推定は尊重されるか無視されません。次の値を指定できます。

#### Enabled

ユーザが提供する選択性推定をすべて尊重します。このオプションは、On を使用して有効にすることもできます。

#### Override-Magic

ユーザ選択性推定は尊重されますが、オプティマイザが最後の手段であるヒューリスティック値 (マジック値とも呼ばれます) の使用を選択する以外に方法がない場合のみ使用されます。

## Disabled

他の推定データが使用できないときは、ユーザ推定は無視され、マジック値が使用されます。このオプションは、Off を使用して無効にすることもできます。

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文に OPTION 句を含めることによって、各文で指定したオプション設定をこのオプションに対するテンポラリ設定や PUBLIC 設定よりも優先させることができます。

## 1.5.6.7.154 uuid\_has\_hyphens オプション

ユニークな識別子の値を文字列に変換するときのフォーマットを設定します。

### 指定可能な値

On、Off

### デフォルト

On

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

### 備考

uuid\_has\_hyphens オプションを On に設定すると、変換後の文字列には 4 つのハイフンが挿入されます。たとえば、12345678-1234-1234-1234-1234567890ab のようになります。このオプションを Off に設定すると、文字列にハイフンは挿入されません。文字列からユニークな識別子の値への変換では、データベースサーバはハイフンが挿入された UUID 文字列とハイフンのない UUID 文字列の両方をサポートします。

## 1.5.6.7.155 verify\_all\_columns オプション [SQL Remote]

ローカルデータベースで発行した更新を含むメッセージがすべてのカラム値を含んで送信されるかどうかを制御します。

### 指定可能な値

On、Off

### デフォルト

Off

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

### 備考

このオプションは、SQL Remote のみで使用します。On に設定すると、ローカルデータベースの発行した更新を含むメッセージはすべてのカラム値を含んで送信され、カラムの重複はサブスクライバデータベースで RESOLVE UPDATE トリガを起動します。

#### 例

次の文は、SQL Anywhere で、すべてのユーザに対して verify\_all\_columns オプションを Off に設定します。

```
SET OPTION PUBLIC.verify_all_columns = 'Off';
```

## 1.5.6.7.156 verify\_password\_function オプション

パスワード規則を実装します。

### 指定可能な値

文字列

### デフォルト

空の文字列

### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり	はい、SET ANY SECURITY OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY SECURITY OPTION あり	はい (現在の接続のみ)、SET ANY SECURITY OPTION あり	×

### 備考

パスワードが設定されている場合、値が空の文字列以外であるときに関数は呼び出されません。

このオプションで指定された関数は、NULL 以外のパスワードが作成されるか設定されると、自動的に呼び出されます。ログインポリシーで change\_password\_dual\_control オプションが有効になったユーザに対しては、指定された関数は呼び出されません。ユーザによって関数が上書きされないようにするには、オプション値を owner.function-name に設定します。ユーザは、データベースに接続可能なパスワードを持っている必要があります。

パスワードの作成または設定に使用する文の検証後、関数が呼び出され、指定した規則を使用してパスワードが検証されます。パスワードが規則に従っている場合、関数は成功を示す NULL を返し、呼び出し元の文が実行されます。それ以外の場合は NULL 以外の文字列が返され、エラーの理由が示されます。

パスワード検証関数は、user-name VARCHAR(128) および new-pwd VARCHAR(255) という 2 つのパラメータを受け取ります。VARCHAR(255) 型の値を返します。

パスワード検証関数がデバッグによってステップスルーされないように、この関数で ALTER FUNCTION function-name SET HIDDEN 文を実行してください。

verify\_password\_function オプションを設定する場合、GRANT CONNECT 文で複数のユーザ ID とパスワードを指定することはできません。

## 例

次の例では、テーブルと関数を 1 つずつ定義し、いくつかのログインポリシーオプションを設定しています。同時に、パスワードに特定の種類の文字を要求し、パスワードの再利用を禁止して、パスワード有効期限を適用するなど、詳細なパスワード規則を実装しています。ユーザ ID が作成されるか、パスワードが変更されると、データベースサーバによって verify\_password\_function オプションを使用して関数が呼び出されます。アプリケーションは、post\_login\_procedure オプションで指定されたプロシージャを呼び出して、パスワードが期限切れになる前にパスワードの変更が必要であることを通知できます。

このサンプルコードは、次の場所でも利用できます: %SQLANYSAMP17%¥SQLAnywhere¥SQL  
¥verify\_password.sql

```
-- Only a database administrator should have permissions on this table.
CREATE TABLE IF NOT EXISTS DBA.t_pwd_history(
    pk          INT          DEFAULT AUTOINCREMENT PRIMARY KEY,
    user_name   CHAR(128),   -- the user whose password is set
    pwd_hash    CHAR(32) );  -- hash of password value to detect
                                -- duplicate passwords
-- Called whenever a non-NULL password is set to verify that the password
-- conforms to password rules.
CREATE OR REPLACE FUNCTION DBA.f_verify_pwd( uid      VARCHAR(128),
                                              new_pwd VARCHAR(255) )
RETURNS VARCHAR(255)
BEGIN
    -- a table with one row per character in new_pwd
    DECLARE local temporary table pwd_chars(
        pos INT PRIMARY KEY,    -- index of c in new_pwd
        c   CHAR( 1 CHAR ) );  -- character
    -- new_pwd with non-alpha characters removed
    DECLARE pwd_alpha_only CHAR(255);
    DECLARE num_lower_chars INT;
    -- enforce minimum length (can also be done with
    -- min_password_length option)
    IF length( new_pwd ) < 6 THEN
        RETURN 'password must be at least 6 characters long';
    END IF;
    -- break new_pwd into one row per character
    INSERT INTO pwd_chars SELECT row_num, substr( new_pwd, row_num, 1 )
        FROM dbo.RowGenerator
        WHERE row_num <= length( new_pwd );
    -- copy of new_pwd containing alpha-only characters
    SELECT list( c, '' ORDER BY pos ) INTO pwd_alpha_only
        FROM pwd_chars WHERE c BETWEEN 'a' AND 'z' OR c BETWEEN 'A' AND 'Z';
    -- number of lower case characters IN new_pwd
    SELECT count(*) INTO num_lower_chars
        FROM pwd_chars WHERE CAST( c AS BINARY ) BETWEEN 'a' AND 'z';
    -- enforce rules based on characters contained in new_pwd
    IF ( SELECT count(*) FROM pwd_chars WHERE c BETWEEN '0' AND '9' )
        < 1 THEN
        RETURN 'password must contain at least one numeric digit';
    ELSEIF length( pwd_alpha_only ) < 2 THEN
        RETURN 'password must contain at least two letters';
    ELSEIF num_lower_chars = 0
        OR length( pwd_alpha_only ) - num_lower_chars = 0 THEN
        RETURN 'password must contain both upper- and lowercase characters';
    END IF;
    -- not the same as any user name
    -- (this could be modified to check against a disallowed words table)
    IF EXISTS( SELECT * FROM SYS.SYSUSER
        WHERE lower( user_name ) IN ( lower( pwd_alpha_only ),
        lower( new_pwd ) ) ) THEN
```

```

        RETURN 'password or only alphabetic characters in password ' ||
              'must not match any user name';
    END IF;
    -- not the same as any previous password for this user
    IF EXISTS( SELECT * FROM t_pwd_history
              WHERE user_name = uid
                AND pwd_hash = hash( uid || new_pwd, 'md5' ) ) THEN
        RETURN 'previous passwords cannot be reused';
    END IF;
    -- save the new password
    INSERT INTO t_pwd_history( user_name, pwd_hash )
        VALUES( uid, hash( uid || new_pwd, 'md5' ) );
    RETURN( NULL );
END;
ALTER FUNCTION DBA.f_verify_pwd SET HIDDEN;
GRANT EXECUTE ON DBA.f_verify_pwd TO PUBLIC;
SET OPTION PUBLIC.verify_password_function = 'DBA.f_verify_pwd';
-- All passwords expire in 180 days. Expired passwords can be changed
-- by the user using the NewPassword connection parameter.
ALTER LOGIN POLICY root password_life_time = 180;
-- If an application calls the procedure specified by the post_login_procedure
-- option, then the procedure can be used to warn the user that their
-- password is about to expire.
ALTER LOGIN POLICY root password_grace_time = 30;
-- Five consecutive failed login attempts will result in a userid being locked.
ALTER LOGIN POLICY root max_failed_login_attempts = 5;

```

## 関連情報

[パスワードおよびユーザ ID の制限事項と考慮事項 \[1554 ページ\]](#)

[セキュリティ: パスワード \[1602 ページ\]](#)

[セキュリティ: パスワード \[1602 ページ\]](#)

[二重制御パスワード \[1557 ページ\]](#)

[min\\_password\\_length オプション \[736 ページ\]](#)

[NewPassword \(NEWPWD\) 接続パラメータ \[94 ページ\]](#)

## 1.5.6.7.157 verify\_threshold オプション [SQL Remote]

更新がレプリケートされるときにどのカラムが確認されるかを制御します。

### 指定可能な値

整数 (バイト)

## デフォルト

1000

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY USER DEFINED OPTION あり	○	はい、SET ANY USER DEFINED OPTION あり
一時的な設定が許可されているかどうか	×	×	×

## 備考

このオプションは、SQL Remote のみで使用します。カラムのデータ型がしきい値より長い場合は、カラムの古い値は UPDATE がレプリケートされるときに確認されません。このようにして、SQL Remote メッセージのサイズが大きくなるようにしますが、競合する長い値の更新が検出されないという短所もあります。

## 1.5.6.7.158 wait\_for\_commit オプション

データが操作されるときに、いつ外部キー整合性をチェックするかを決定します。

## 指定可能な値

On、Off

## デフォルト

Off

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	○	はい、SET ANY PUBLIC OPTION あり
一時的な設定が許可されているかどうか	はい、SET ANY PUBLIC OPTION あり	はい (現在の接続のみ)	×

## 備考

このオプションを On に設定すると、データベースは次の COMMIT 文まで外部キー整合性をチェックしません。Off の場合は、CHECK ON COMMIT 句で作成されていないすべての外部キーが、挿入、更新、削除時にチェックされます。

## 1.5.6.7.159 webservice\_namespace\_host オプション

DISH サービスの XML ネームスペース仕様内のホスト名を指定します。

## 指定可能な値

NULL または hostname-string (空も可能)

## デフォルト

空の文字列

## スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

## 備考

このオプションは、HTTP Host 要求ヘッダのデフォルトの使用法より優先されます。

DISH サービスは Web Services Description Language (WSDL) をエクスポートします。WSDL は、使用可能な SOAP サービスの説明を含んだ XML ドキュメントです。XML ドキュメント内の targetNamespace と soapAction 操作の URL にはホスト名が含まれます。このオプションがデフォルト値の NULL に設定されている場合、ホスト名はデータベースサーバ実行中のコンピュータのホスト名になります。このオプションが文字列値に設定されていると、ホスト名にはその文字列が使用されます。このオプションは、展開時に開発用以外のホストをターゲットとする Web サービスクライアントアプリケーションを開発する場合に使用することを目的としています。

### 1.5.6.7.160 webservice\_sessionid\_name オプション

セッション管理が使用されているかどうかを判断するために Web サーバが使用する項目を再定義します。

#### 指定可能な値

NULL または sessionid-string

#### デフォルト

SessionID

#### スコープ

	PUBLIC ロール	現在のユーザ	他のユーザ
永続的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×
一時的な設定が許可されているかどうか	はい、SET ANY SYSTEM OPTION あり	×	×

## 備考

webservice\_sessionid\_name オプションを使用すると、セッション管理が使用中かどうかを判断するために Web サーバが探す項目を再定義できます。デフォルトでは、セッションの識別子名は SessionID です。

このデータベースオプションは、Web アプリケーションでセッションの識別子名に SessionID 以外の名前 (JSessionID など) を使用する場合に便利です。

Web アプリケーションで URL ベースのセッション管理を使用する場合、セッションの識別子は GET 要求に出現します。

```
GET /service?JSessionID=session_63520025849921 HTTP/1.1
```

Web アプリケーションで cookie ベースのセッション管理を使用する場合、セッションの識別子は Cookie ヘッダに出現します。

```
Cookie: JSessionID=cookie_session_63520025481454
```

webservice\_sessionid\_name データベースオプションで指定した名前は、Web サーバが Web サービス要求を分析するときだけ使用されます。

#### 例

- 次の例は、Web サービス SessionID の名前を JSessionID に変更します。

```
SET OPTION PUBLIC.webservice_sessionid_name="JSessionID";
```

## 1.5.6.8 さまざまなカテゴリのデータベースオプション

データベースオプションには、互換性、同期、SQL Remote、および Interactive SQL の 4 つのカテゴリがあります。

このセクションの内容:

### [Open Client/jConnect TDS の互換性オプション \[833 ページ\]](#)

SAP Open Client または jConnect を使用してデータベースへの接続が行われた場合、Adaptive Server Enterprise との互換性の保持のために、現在の接続に対していくつかのオプションが明示的に設定されます。

### [Transact-SQL および ANSI/ISO SQL 標準の互換性オプション \[835 ページ\]](#)

SQL Anywhere の動作に Adaptive Server Enterprise との互換性を持たせたり、両方の旧バージョンの動作をサポートしたり、ANSI/ISO SQL 標準の動作を使用可能にしたりできる互換性オプションがあります。

### [同期オプション \[836 ページ\]](#)

次のデータベースオプションは、Mobile Link 同期クライアントとして使用する SQL Anywhere データベースの設定に使用できます。

### [SQL Remote オプション \[837 ページ\]](#)

SQL Remote のレプリケーション動作を制御するオプションがいくつか用意されています。

## 関連情報

[Interactive SQL オプション \[1003 ページ\]](#)

## 1.5.6.8.1 Open Client/jConnect TDS の互換性オプション

SAP Open Client または jConnect を使用してデータベースへの接続が行われた場合、Adaptive Server Enterprise との互換性の保持のために、現在の接続に対していくつかのオプションが明示的に設定されます。

TDS 接続用のデフォルトデータベースオプションは、システムプロシージャ `sp_tsql_environment` を使用して設定されます。

### i 注記

`sp_tsql_environment` プロシージャは、変更しないでください。このプロシージャは、システムが専用に使用します。

データベースサーバでは長いユーザ名およびパスワードが許可されますが、TDS クライアント名とパスワードは 30 バイトを超えることはできません。パスワードまたはユーザ ID が 30 バイト以上の場合、TDS 経由の接続試行 (たとえば、jConnect の使用) では無効なユーザまたはパスワードエラーが返されます。

オプション	設定
<code>allow_nulls_by_default</code> オプション	Off
<code>ansi_blanks</code> オプション	On
<code>ansinull</code> オプション	Off
<code>chained</code> オプション	Off
<code>close_on_endtrans</code> オプション	Off (SQL Anywhere データベースのみ)
<code>date_format</code> オプション	YYYY-MM-DD
<code>date_order</code> オプション	MDY
<code>escape_character</code> オプション (システムで使用するために予約済み)	Off
<code>isolation_level</code> オプション	1
<code>on_tsq_error</code> オプション	Continue
<code>quoted_identifier</code> オプション	Off
<code>time_format</code> オプション	HH:NN:SS.SSS
<code>timestamp_format</code> オプション	YYYY-MM-DD HH:NN:SS.SSS
<code>tsql_variables</code> オプション	On

このセクションの内容:

[TDS 接続用オプションの設定値の変更 \[834 ページ\]](#)

TDS 接続用オプションを変更します。

## 関連情報

[allow\\_nulls\\_by\\_default オプション \[641 ページ\]](#)

[ansi\\_blanks オプション \[646 ページ\]](#)

[ansinull オプション \[652 ページ\]](#)  
[chained オプション \[666 ページ\]](#)  
[close\\_on\\_endtrans オプション \[670 ページ\]](#)  
[date\\_format オプション \[684 ページ\]](#)  
[date\\_order オプション \[686 ページ\]](#)  
[escape\\_character オプション \(システムで使用するために予約済み\) \[698 ページ\]](#)  
[isolation\\_level オプション \[711 ページ\]](#)  
[on\\_tsq\\_error オプション \[747 ページ\]](#)  
[quoted\\_identifier オプション \[766 ページ\]](#)  
[time\\_format オプション \[808 ページ\]](#)  
[timestamp\\_format オプション \[811 ページ\]](#)  
[tsql\\_variables オプション \[819 ページ\]](#)

## 1.5.6.8.1.1 TDS 接続用オプションの設定値の変更

TDS 接続用オプションを変更します。

### 前提条件

ユーザ本人が所有するプロシージャを作成するには、CREATE PROCEDURE システム権限が必要です。

login\_procedure オプションを設定するには SET ANY SECURITY OPTION システム権限が必要です。

### コンテキスト

データベースサーバは TDS を通じてアプリケーションからの要求を処理するとき、関連するデータベースオプションを、Adaptive Server Enterprise のデフォルトの動作と互換性のある値に自動的に設定します。

### 手順

1. 目的のデータベースオプションを設定するプロシージャを作成します。
2. PUBLIC login\_procedure オプションに新しいプロシージャ名を設定します。

## 結果

今後の接続では、このプロシージャを使用します。別のユーザ ID に対して、別のプロシージャを設定することもできます。

### 例

このプロシージャの例では、デフォルト設定の `quoted_identifier` オプションだけを変更します。

```
CREATE PROCEDURE DBA.my_startup_procedure ()
BEGIN
  CALL dbo.sp_login_environment ();
  IF CONNECTION_PROPERTY('CommProtocol')='TDS' THEN
    SET TEMPORARY OPTION quoted_identifier='On';
  END IF
END;
GRANT EXECUTE ON DBA.my_startup_procedure TO PUBLIC;
```

次のコマンドを実行し、PUBLIC login\_procedure オプションを新しいプロシージャ名に設定します。

```
SET OPTION PUBLIC.login_procedure = 'DBA.my_startup_procedure';
```

次のコマンドを実行し、ユーザ Browser のみに対して、login\_procedure オプションを新しいプロシージャ名に設定します。

```
SET OPTION Browser.login_procedure = 'DBA.my_startup_procedure';
```

## 関連情報

[データベースオプション \[622 ページ\]](#)

[login\\_procedure オプション \[720 ページ\]](#)

### 1.5.6.8.2 Transact-SQL および ANSI/ISO SQL 標準の互換性オプション

SQL Anywhere の動作に Adaptive Server Enterprise との互換性を持たせたり、両方の旧バージョンの動作をサポートしたり、ANSI/ISO SQL 標準の動作を使用可能にしたりできる互換性オプションがあります。

SAP Adaptive Server Enterprise に互換性オプションがある場合、そのエントリには ASE のフラグが付けられます。

これらのオプションのデフォルト設定は、Adaptive Server Enterprise のデフォルト設定と一部異なります。SQL Anywhere データベースと Adaptive Server Enterprise データベース間の互換性を確保するためには、個々の互換性オプションを明示的に設定してください。

Adaptive Server Enterprise との互換性をさらに高めるために、SQL Anywhere の SET OPTION 文の代わりに Transact-SQL の SET 文を使用して、いくつかのオプションを現在の接続の間だけ設定できます。

- allow\_nulls\_by\_default オプション

- ansi\_blanks オプション
- ansi\_close\_cursors\_on\_rollback オプション
- ansi\_permissions オプション (ASE)
- ansi\_substring オプション
- ansi\_update\_constraints オプション
- ansinull オプション (ASE)
- chained オプション (ASE)
- close\_on\_endtrans オプション (ASE)
- continue\_after\_raisererror オプション
- conversion\_error オプション
- date\_format オプション (ASE)
- date\_order オプション
- divide\_by\_zero\_error オプション
- escape\_character オプション (システムで使用するために予約済み)
- fire\_triggers オプション
- first\_day\_of\_week オプション (ASE)
- isolation\_level オプション (ASE)
- nearest\_century オプション
- non\_keywords オプション
- on\_tsq\_error オプション
- quoted\_identifier オプション (ASE)
- reserved\_keywords オプション
- string\_truncation オプション (ASE)
- time\_format オプション
- timestamp\_format オプション
- tsq\_outer\_joins オプション
- tsq\_variables オプション
- SET 文 [T-SQL]

### 1.5.6.8.3 同期オプション

次のデータベースオプションは、Mobile Link 同期クライアントとして使用する SQL Anywhere データベースの設定に使用できません。

オプション	値	デフォルト
default_timestamp_increment オプション	1 ~ 1000000 マイクロ秒を表す整数	1
delete_old_logs オプション [SQL Remote]	On、Off、Delay、n days (日数)	Off
ml_remote_id オプション [Mobile Link]	Mobile Link による同期時にデータベースをユニークに識別する任意の値	NULL
prevent_article_pkey_update オプション	On、Off	On

オプション	値	デフォルト
truncate_timestamp_values オプション	On、Off	Off

## 関連情報

[default\\_timestamp\\_increment オプション \[692 ページ\]](#)  
[delete\\_old\\_logs オプション \[Mobile Link\]\[SQL Remote\] \[695 ページ\]](#)  
[ml\\_remote\\_id オプション \[Mobile Link\] \[738 ページ\]](#)  
[prevent\\_article\\_pkey\\_update オプション \[Mobile Link\] \[760 ページ\]](#)  
[truncate\\_timestamp\\_values オプション \[815 ページ\]](#)

### 1.5.6.8.4 SQL Remote オプション

SQL Remote のレプリケーション動作を制御するオプションがいくつか用意されています。

オプション	値	デフォルト
blob_threshold オプション [SQL Remote]	整数 (バイト)	256
compression オプション [SQL Remote]	-1 ~ 9 の整数	6
delete_old_logs オプション [SQL Remote]	On、Off、Delay、 <i>n</i> days (日数)	Off
external_remote_options オプション [SQL Remote]	On、Off	Off
qualify_owners オプション [SQL Remote]	On、Off	On
quote_all_identifiers オプション [SQL Remote]	On、Off	Off
replication_error オプション [SQL Remote]	スタアドプロシージャ名	(プロシージャなし)
replication_error_piece オプション [SQL Remote]	スタアドプロシージャ名	(プロシージャなし)
save_remote_passwords オプション [SQL Remote]	On、Off	On
sr_date_format オプション [SQL Remote]	<i>date-string</i>	yyyy/mm/dd
sr_time_format オプション [SQL Remote]	<i>time-string</i>	hh:nn:ss.Ssssss
sr_timestamp_format オプション [SQL Remote]	<i>timestamp-string</i>	yyyy/mm/dd hh:nn:ss.Ssssss
sr_timestamp_with_time_zone_format オプション [SQL Remote]	<i>timestamp-with-time-zone-string</i>	yyyy/mm/dd hh:nn:ss.Ssssss +hh:nn

オプション	値	デフォルト
subscribe_by_remote オプション [SQL Remote]	On, Off	On
verify_all_columns オプション [SQL Remote]	On, Off	Off
verify_threshold オプション [SQL Remote]	整数 (バイト)	1000

## 関連情報

- [blob\\_threshold オプション \[SQL Remote\] \[661 ページ\]](#)
- [compression オプション \[SQL Remote\] \[672 ページ\]](#)
- [delete\\_old\\_logs オプション \[Mobile Link\]\[SQL Remote\] \[695 ページ\]](#)
- [external\\_remote\\_options オプション \[SQL Remote\] \[701 ページ\]](#)
- [qualify\\_owners オプション \[SQL Remote\] \[764 ページ\]](#)
- [quote\\_all\\_identifiers オプション \[SQL Remote\] \[765 ページ\]](#)
- [replication\\_error オプション \[SQL Remote\] \[770 ページ\]](#)
- [replication\\_error\\_piece オプション \[SQL Remote\] \[771 ページ\]](#)
- [save\\_remote\\_passwords オプション \[SQL Remote\] \[780 ページ\]](#)
- [sr\\_date\\_format オプション \[SQL Remote\] \[787 ページ\]](#)
- [sr\\_time\\_format オプション \[SQL Remote\] \[788 ページ\]](#)
- [sr\\_timestamp\\_format オプション \[SQL Remote\] \[790 ページ\]](#)
- [sr\\_timestamp\\_with\\_time\\_zone\\_format オプション \[SQL Remote\] \[792 ページ\]](#)
- [subscribe\\_by\\_remote オプション \[SQL Remote\] \[802 ページ\]](#)
- [verify\\_all\\_columns オプション \[SQL Remote\] \[825 ページ\]](#)
- [verify\\_threshold オプション \[SQL Remote\] \[828 ページ\]](#)

## 1.5.7 接続、データベース、データベースサーバのプロパティ

接続、データベース、およびデータベースサーバのプロパティは、データがデータベースでどのようにアクセスおよび処理されるのかを制御します。

このセクションの内容:

### [接続プロパティの一覧 \[839 ページ\]](#)

接続プロパティは、データベースへの各接続で使用できます。接続プロパティ名では大文字と小文字が区別されません。CONNECTION\_PROPERTY システム関数を使用するか、sa\_conn\_properties システムプロシージャを使用して、接続プロパティを取得します。

### [データベースサーバプロパティの一覧 \[859 ページ\]](#)

データベースサーバプロパティは、データベースへの各接続で使用できます。PROPERTY システム関数を使用して個々のプロパティの値を取得し、sa\_eng\_properties システムプロシージャを使用してすべてのデータベースサーバプロパティの値を取得します。

#### データベースプロパティの一覧 [874 ページ]

データベースプロパティは、データベースへの各接続で使用できます。DB\_PROPERTY システム関数を使用して個々のデータベースプロパティの値を取得するか、sa\_db\_properties システムプロシージャを使用してすべてのデータベースプロパティの値を取得します。

#### ユーザ定義プロパティ [888 ページ]

ユーザ定義プロパティは、データベースへの各接続で使用できます。

#### データベースサーバプロパティの追跡 [888 ページ]

長期間にわたってデータベースサーバプロパティの数値の変更を追跡できるように、データベースサーバにデータベースサーバプロパティの数値を格納できます。

## 1.5.7.1 接続プロパティの一覧

接続プロパティは、データベースへの各接続で使用できます。接続プロパティ名では大文字と小文字が区別されません。CONNECTION\_PROPERTY システム関数を使用するか、sa\_conn\_properties システムプロシージャを使用して、接続プロパティを取得します。

### 例

次の文は、現在の接続によってファイルから読み込まれたページ数を返します。

```
SELECT CONNECTION_PROPERTY ( 'DiskRead' );
```

sa\_conn\_properties システムプロシージャを使用して、すべての接続プロパティの値を取り出します。

```
CALL sa_conn_properties( );
```

## 接続プロパティ

プロパティ名	説明
allow_nulls_by_default	NULL か NOT NULL かを指定しないで作成された新規カラムに NULL 値を許可するかどうか。このプロパティは接続の allow_nulls_by_default オプションに相当します。
allow_read_client_file	データベースサーバがクライアントコンピュータ上のファイルの読み込みを許可するかどうか。このプロパティは接続の allow_read_client_file オプションに相当します。
allow_snapshot_isolation	スナップショットアイソレーションを有効にするか無効にするか。このプロパティは接続の allow_snapshot_isolation オプションに相当します。

プロパティ名	説明
allow_write_client_file	データベースサーバがクライアントコンピュータへのファイルの書き込みを許可するかどうか。このプロパティは接続の allow_write_client_file オプションに相当します。
ansi_blanks	文字データがクライアント側でランケットされるタイミングを示す。このプロパティは接続の ansi_blanks オプションに相当します。
ansi_close_cursors_on_rollback	WITH HOLD 句で開いたカーソルを、ROLLBACK を実行するときに閉じるかどうか。このプロパティは接続の ansi_close_cursors_on_rollback オプションに相当します。
ansi_permissions	DELETE 文と UPDATE 文の権限をチェックするかどうか。このプロパティは接続の ansi_permissions オプションに相当します。
ansi_substring	start パラメータまたは length パラメータに負の値が設定された場合の SUBSTRING (SUBSTR) 関数の動作。このプロパティは接続の ansi_substring オプションに相当します。
ansi_update_constraints	更新可能な範囲。このプロパティは接続の ansi_update_constraints オプションに相当します。
ansinull	NULL 値をどのように解釈するか。このプロパティは ansinull オプションに相当します。
ApplInfo	接続を確立したクライアントに関する情報。HTTP 接続では、ブラウザの情報が含まれています。SAP Open Client または jConnect の古いバージョンを使用した接続については、情報が不完全な場合があります。
ApproximateCPUTime	特定の接続で使用された累積 CPU 時間の概算値 (秒単位)。この値と実際の CPU 時間との誤差は、通常は 5 ~ 10 % ですが、最大で 50 % に達することもあります。マルチプロセッサコンピュータでは、CPU (ハイパースレッドまたはコア) ごとに使用時間が積算されるため、すべての接続の累積時間を合計した値が実際の経過時間より大きくなる場合もあります。
audit_log	監査ログが保存されるロケーション。
auditing	データベースで監査を有効にする (On) か、無効にするか (Off)。このプロパティは auditing オプションに相当します。
auditing_options	このプロパティはシステムで使用するために予約されています。
Authenticated	アプリケーションが有効な接続認証文字列を送信する (Yes) か、送信しない (No) か。
AuthType	接続時に使用される認証のタイプ。戻り値は、Standard、Integrated、Kerberos、LDAPUA、CloudAdmin、または空の文字列です。接続が内部接続である場合、または AUTHORIZATION OFF を使用する HTTP サービスの接続である場合、値は空の文字列になります。
auto_commit	データベースサーバが各文の後に自動でコミットするかどうか。デフォルトでは、データベースサーバは手動コミットモードで動作します。オートコミットをオンにするには、auto_commit データベースオプション (サーバ側オプション) を設定してください。同じ名前の Interactive SQL オプションとこのオプションを混同しないように注意してください。
auto_commit_on_create_local_temp_index	ローカルテンポラリテーブルにインデックスを作成する前にデータベースサーバが COMMIT を実行するかどうか。このプロパティは auto_commit_on_create_local_temp_index オプションの値に相当します。

プロパティ名	説明
background_priority	このプロパティは使用されなくなりました。現在の接続がどの程度他の接続のパフォーマンスに影響するかを示す、接続の background_priority オプションの値。
BlockedOn	現在の接続がブロックされるかブロックされないか (ゼロ)。ロックの競合により接続がブロックされると、値は接続がブロックされた接続の番号になります。
blocking	ロック競合に対するデータベースサーバの動作。このプロパティは接続の blocking オプションに相当します。
blocking_others_timeout	現在の接続がロールバックされる前に、別の接続が現在の接続のローおよびテーブルロックでブロックされる可能性がある時間の長さ。このプロパティは blocking_others_timeout オプションの値に相当します。
blocking_timeout	トランザクションがロックを獲得するまでの待ち時間 (ミリ秒単位)。このプロパティは blocking_timeout オプションに相当します。
BytesReceived	クライアント/サーバ通信中に受信したバイト数。この値は、HTTP 接続と HTTPS 接続の場合に更新されません。
BytesReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したバイト数。この値は、圧縮が無効の場合は BytesReceived の値と同じです。
BytesSent	クライアント/サーバ通信中に送信されたバイト数。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。
BytesSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたバイト数。この値は、圧縮が無効の場合は BytesSent の値と同じです。
CacheHits	成功したキャッシュ読み込み回数。
CacheRead	キャッシュ内で検索されたデータベースページの数。
CacheReadIndInt	キャッシュから読み込まれたインデックス内部ノードページの数。
CacheReadIndLeaf	キャッシュから読み込まれたインデックスリーフページの数。
CacheReadTable	キャッシュから読み込まれたテーブルページの数。
CacheReadWorkTable	キャッシュワークテーブルの読み込み数。
CarverHeapPages	クエリ最適化などの短期的な目的に使用されたヒープページの数。
chained	BEGIN TRANSACTION 文が指定されていない場合に使用されるトランザクションモードを示す、chained オプションの値。
CharSet	接続で使用される CHAR 文字セット。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
checkpoint_time	データベースサーバがチェックポイントを行わずに実行される最大分数を示す checkpoint_time オプションの値。

プロパティ名	説明
cis_option	リモートデータアクセス用のデバッグ情報をデータベースサーバメッセージウィンドウに表示する場合、7を指定。リモートデータアクセス用のデバッグ情報をデータベースサーバメッセージウィンドウに表示しない場合、0を指定。このプロパティは cis_option オプションに相当します。
cis_rowset_size	各フェッチに対してリモートサーバから返されるローの数。このプロパティは cis_rowset_size オプションの値に相当します。
ClientLibrary	接続ライブラリタイプ。値は、jConnect 接続の場合は jConnect、SAP Open Client 接続の場合は CT_Library、HTTP 接続の場合は None、ODBC 接続、Embedded SQL 接続、OLE DB 接続、ADO.NET 接続、SQL Anywhere JDBCドライバ接続の場合は CmdSeq になります。
ClientNodeAddress	クライアント/サーバ接続のクライアント側に対応するノード。クライアントとサーバの両方が同じコンピュータにある場合は、空の文字列を返します。このプロパティは NodeAddress プロパティの同意語です。 現在実行中の要求がイベントハンドラの一部である場合、値は NA になります。
ClientPort	クライアントの TCP/IP ポート番号、または接続の種類が TCP/IP でない場合は 0。
ClientStmtCacheHits	クライアントに文がキャッシュされていることが理由で、この接続には不要な準備文の数。この値は、クライアントでの文のキャッシュが無効になった場合に必要となる、追加の準備文の数です。
ClientStmtCacheMisses	この接続において、クライアントのキャッシュに存在した文の中で、再び準備された文の数。この値は、キャッシュされた文が再使用できるか検討されたものの、スキーマの変更、データベースのオプション設定の変更、DROP VARIABLE 文によって再使用できなかった回数です。
close_on_endtrans	カーソルがトランザクションの終了時に閉じられるかどうか。このプロパティは close_on_endtrans オプションに相当します。
collect_statistics_on_dml_updates	INSERT、DELETE、UPDATE などのデータを変更する DML 文の実行中に統計情報を収集するかどうか。このプロパティは collect_statistics_on_dml_updates オプションに相当します。
Commit	処理されたコミット要求の数。
CommLink	接続用の通信リンク。この値は、サポートされているネットワークプロトコルの 1 つであり、同一コンピュータ接続の場合は local になります。現在実行中の要求がイベントハンドラの一部である場合、値は NA になります。
CommNetworkLink	接続用の通信リンク。返される値は、サポートされているネットワークプロトコルの 1 つです。値には SharedMemory と TCPIP があります。値には、同一コンピュータかどうかにかかわらず、常にリンク名が含まれます。現在実行中の要求がイベントハンドラの一部である場合、値は NA になります。
CommProtocol	通信プロトコル。値は、SAP Open Client および jConnect 接続の場合は TDS、HTTP 接続の場合は HTTP、HTTPS 接続の場合は HTTPS です。値は、ODBC、Embedded SQL、OLE DB、ADO.NET、および SQL Anywhere JDBCドライバ接続の場合は CmdSeq です。
Compression	接続で通信圧縮を有効するかどうか。現在実行中の要求がイベントハンドラの一部である場合、値は NA になります。

プロパティ名	説明
conn_auditing	auditing オプションが <i>On</i> に設定されている場合に、接続で監査を有効にするか無効にするか。このプロパティは conn_auditing オプションに相当します。
ConnectedTime	接続がこれまで接続してきた合計の時間数 (秒単位)。
connection_authentication	クライアントの認証に使用される文字列。データベースが変更可能になる前に、認証が必要です。このプロパティは connection_authentication オプションに相当します。
connection_type	connection_type データベースオプションの値 (次のうちの 1 つ)。Event、Internal、Standard、Monitor
continue_after_raiserror	RAISERROR 文が見つかるたびにプロシージャまたはトリガの実行を停止するかどうか。このプロパティは continue_after_raiserror オプションに相当します。
conversion_error	データベースの情報をフェッチするときにデータ型変換エラーをレポートするかどうか。このプロパティは conversion_error オプションの値に相当します。
cooperative_commit_timeout	このプロパティは使用されなくなりました。cooperative_commit_timeout オプションの値 (cooperative_commit_timeout オプションの値は、データベースサーバがディスクへの書き込み前に他の接続の書き込みによって 1 つのログページが満杯になるまで待っているミリ秒単位の時間)。
cooperative_commits	このプロパティは使用されなくなりました。cooperative_commits オプションの値。コミットがいつディスクに書き込まれるかを示す On または Off。
CurrentLineNumber	接続で実行されているプロシージャまたは複合文の現在の行番号。実行されているプロシージャの名前は CurrentProcedure プロパティで取得できます。現在の行がクライアント側からの複合文の一部である場合は、空の文字列を返します。
CurrentProcedure	接続で現在実行されているプロシージャの名前。接続でネストされたプロシージャコールが実行されている場合は、現在のプロシージャの名前を返します。実行されているプロシージャがない場合は、空の文字列を返します。
Cursor	現在データベースサーバが保持している、宣言されたカーソルの数。
CursorOpen	現在データベースサーバが保持している、開いたカーソルの数。
database_authentication	データベースの認証に使用される文字列を示す。データベースサーバの認証が完了してからでなければ、データベースに変更を加えることはできません。このプロパティは database_authentication オプションに相当します。
date_format	date_format オプションの値 (データベースから取り出した日付の形式を示す文字列)。
date_order	date_order オプションの値 (日付形式を設定する方法を示す文字列)。
DBNumber	データベースの ID 番号。
db_publisher	データベースパブリッシャのユーザ ID。このプロパティは db_publisher オプションに相当します。

プロパティ名	説明
debug_messages	DEBUG ONLY 句を含む MESSAGE 文を実行するかどうか。このプロパティは debug_messages オプションに相当します。
dedicated_task	要求処理タスクがこの接続からの要求の処理専用であるかどうか。このプロパティは dedicated_task オプションに相当します。
default_dbSPACE	デフォルト DB 領域の名前。デフォルト DB 領域が指定されていない場合、空の文字列。このプロパティは default_dbSPACE オプションに相当します。
default_timestamp_increment	カラム中の値をユニークにするために TIMESTAMP データ型のカラムに追加する時間 (マイクロ秒単位)。このプロパティは default_timestamp_increment に相当します。
delayed_commit_timeout	COMMIT の後でデータベースサーバがアプリケーションに制御を戻すまでの待ち時間 (ミリ秒単位)。このプロパティは delayed_commit_timeout オプションに相当します。
delayed_commits	COMMIT の後でデータベースサーバからアプリケーションに制御が戻されるかどうか。このプロパティは delayed_commits オプションに相当します。
DiskRead	ディスクから読み込まれたページの数。
DiskReadHint	ディスクの読み込みヒント数。
DiskReadHintPages	ディスクの読み込みヒントページの数。
DiskReadIndInt	ディスクから読み込まれたインデックス内部ノードページの数。
DiskReadIndLeaf	ディスクから読み込まれたインデックスリーフページの数。
DiskReadTable	ディスクから読み込まれたテーブルページの数。
DiskReadWorkTable	ディスクワークテーブルの読み込み数。
disk_sandbox	データベースの読み込み/書き込みファイル操作が、メインのデータベースファイルがあるディレクトリに制限されているかどうか。このプロパティは disk_sandbox オプションに相当します。
DiskSyncRead	同期的に発行されたディスクの読み込み数。
DiskSyncWrite	同期的に発行された書き込み数。
DiskWaitRead	データベースサーバによる非同期読み込みの待機回数。
DiskWaitWrite	データベースサーバによる非同期書き込みの待機回数。
DiskWrite	ディスクに書き込まれた修正ページの数。
DiskWriteHint	ディスクの書き込みヒント数。
DiskWriteHintPages	ディスクの書き込みヒントページの数。

プロパティ名	説明
divide_by_zero_error	ゼロ除算によってエラーを返す (On) か返さない (Off) か。このプロパティは divide_by_zero_error オプションに相当します。
Encryption	データベースが暗号化されているかどうか。
escape_character	このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。
EventName	接続でイベントハンドラが実行されている場合の関連するイベント名。それ以外の場合は、空の文字列を返します。
exclude_operators	このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。
ExprCacheAbandons	ヒット率が低すぎるために式キャッシュが中止された回数。
ExprCacheDropsToRead Only	ヒット率が低いために式キャッシュが読み込み専用ステータスに変更された回数。
ExprCacheEvicts	式キャッシュからの出力数。
ExprCacheHits	式キャッシュ内のヒット数。
ExprCacheInserts	式キャッシュに挿入された値の数。
ExprCacheLookups	式キャッシュ内で実行されたルックアップの回数。
ExprCacheResumesOfReadWrite	ヒット率の上昇によって式キャッシュが読み込み/書き込みステータスに戻った回数。
ExprCacheStarts	式キャッシュが開始された回数。
extern_login_credentials	リモート接続の試行で、ログインしたユーザの外部ログインクレデンシャルを使用するか、有効なユーザの外部ログインクレデンシャルを使用するか。このプロパティは extern_login_credentials オプションに相当します。
extended_join_syntax	複数テーブルのジョインに対して、重複する相関名構文を持つクエリを許可するか、エラーとしてレポートするか。このプロパティは extended_join_syntax オプションに相当します。
fire_triggers	データベースでトリガ起動をするかどうか。このプロパティは fire_triggers オプションに相当します。
first_day_of_week	週の最初の曜日を表す番号。7 は日曜日、1 は月曜日。このプロパティは first_day_of_week オプションに相当します。
for_xml_null_treatment	NULL 値を含む要素と属性を結果から省略する場合、値は Omit。クエリ内で FOR XML 句が使用されている場合に NULL 値に対応する空の要素または属性が生成される場合、値は Empty。このプロパティは for_xml_null_treatment オプションに相当します。

プロパティ名	説明
force_view_creation	このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。
FullCompare	インデックスのハッシュ値を超えて実行された比較の回数。
GetData	GETDATA 要求の数。
global_database_id	DEFAULT GLOBAL AUTOINCREMENT が指定されたカラムの開始値。このプロパティは global_database_id オプションに相当します。
HashForcedPartitions	メモリ競合が原因でハッシュ演算子が分割された回数。
HashRowsFiltered	ビットベクトルフィルタによって拒否されたプローブ数。
HashRowsPartitioned	ハッシュワークテーブルに書き込まれたロー数。
HasSecuredFeature	1つ以上の機能セットが保護されているか (Yes)、保護されていないか (No)。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
HashWorkTables	ハッシュベースの演算用に作成されたワークテーブルの数。
HeapsCarver	クエリ最適化などの短期的な目的に使用されたヒープの数。
HeapsLocked	キャッシュ内で現在ロックされている再配置可能なヒープの数。
HeapsQuery	クエリ処理 (ハッシュ操作およびソート操作) に使用されるヒープの数。
HeapsRelocatable	再配置可能なヒープの数。
http_connection_pool_basesize	データベース接続の公称スレッショルドサイズ。このプロパティは http_connection_pool_basesize オプションに相当します。
http_connection_pool_timeout	未使用の接続が接続プールに格納される最長時間。このプロパティは http_connection_pool_basesize オプションに相当します。
http_session_timeout	現在の HTTP セッションタイムアウト (分単位)。このプロパティは http_session_timeout オプションに相当します。
HttpServiceName	現在の HTTP 要求のサービス名エントリポイント。このプロパティは、エラーレポートやフロー制御の場合に便利です。このプロパティを HTTP 要求以外により実行されたスタアドプロシージャから選択した場合、または接続が現在アクティブではないか、HTTP セッションの再開を待っている状態である場合は、空の文字列を返します。
IdleTimeout	接続のアイドルタイムアウト値。現在実行中の要求がイベントハンドラの一部である場合、値は NA になります。
IndAdd	インデックスに追加されたエントリの数。
IndLookup	インデックス内で検索されたエントリの数。

プロパティ名	説明
integrated_server_name	統合化ログインを目的とする Windows ユーザグループメンバーシップの検索に使用されるドメインコントローラサーバの名前。このプロパティは integrated_server_name オプションに相当します。
IsDebugger	接続がデバッガの実行に使用されている (Yes) かいない (No) か。現在の接続番号がデバッガ接続の接続番号と対応する場合は値が Yes、それ以外の場合は No になります。
isolation_level	接続の独立性レベル。このプロパティは isolation_level オプションに相当します。
java_class_path	クラスを検索する追加のディレクトリまたは JAR ファイルのリスト。このプロパティは java_class_path オプションに相当します。
java_location	データベースに対して Java VM が指定されている場合の Java VM のパス。このプロパティは java_location オプションに相当します。
java_vm_options	データベースサーバが Java VM を起動するときに使用するコマンドラインオプション。このプロパティは java_vm_options オプションに相当します。
java_main_userid	このプロパティは使用されなくなりました。
Language	ロケール言語。
LastCommitRedoPos	接続によって最後の COMMIT 操作がトランザクションログに書き込まれた後の REDO ログの位置。
LastIdle	要求間のチェックの数。
LastPlanText	接続で最後に実行されたクエリの長いテキストプラン。最後のプランを記憶するかどうかは、sa_server_option システムプロシージャの RememberLastPlan オプションを設定するか、-zp サーバオプションを使用することで制御します。
LastReqTime	指定された接続において最後の要求が開始された時刻 (データベースのタイムゾーン)。このプロパティは、イベントなどの内部接続の場合は空の文字列を返すことがあります。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
LastStatement	現在の接続で最後に準備された SQL 文。LastStatement の値は、文が準備されると同時に設定され、文が削除されると同時にクリアされます。各接続につき 1 つの文の文字列のみが記憶されます。クライアントでの文のキャッシュが有効であり、キャッシュされた文が再使用されている場合、値は空の文字列になります。ある接続について sa_conn_activity が空でない値を返した場合、その接続で現在実行されている文である可能性が高くなります。その文が完了している場合は、文がすでに削除され、このプロパティの値がクリアされている可能性があります。アプリケーションが複数の文を準備し、それらの文のステートメントハンドルを保持している場合、LastStatement が返す値は接続で現在実行されている処理を表しません。
LivenessTimeout	現在の接続の活性タイムアウト時間。現在実行中の要求がイベントハンドラの一部である場合、値は NA になります。
lock_rejected_rows	このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。
LockCount	接続で保持されているロックの数。

プロパティ名	説明
LockObjectOID	テーブル、ミューテックス、またはセマフォで接続がブロックされていない場合、または接続が CONNECTION_PROPERTY を呼び出した接続とは別のデータベース上にある場合、値は 0 になります。それ以外の場合、LockObjectOID は接続がブロックされているテーブル、永続ミューテックス、または永続セマフォのオブジェクト ID になります。負の値は一時ミューテックスまたはセマフォの ID を示します。LockObjectOID によって、sp_list_mutexes_semaphores システムプロシージャを使用して一時ミューテックスおよびセマフォに関する情報を検索できます。オブジェクトがテーブルの場合、LockObjectOID によって、SYSTAB システムビューを使用してテーブル情報を検索できます。
LockObjectType	接続がブロックされているオブジェクトタイプの ID。この ID を使用して、SYSOBJECT ビューにあるオブジェクトタイプを検索します。'TABLE' または 'MUTEX SEMAPHORE' のどちらかになります。
LockIndexID	ロックされたインデックスの識別子。
LockName	接続が解放を待っているロックを示す 64 ビット符号なし整数。
LockRowID	ロックされたローの識別子。
LockTableOID	接続がブロックされていない場合、または接続が CONNECTION_PROPERTY を呼び出した接続とは別のデータベース上にある場合は 0。それ以外の場合は、接続が解放を待っているロックが設定されているテーブルのオブジェクト ID。このオブジェクト ID によって、SYSTAB システムビューを使用してテーブル情報を検索できます。
log_deadlocks	デッドロック情報を記録する (On) か、しない (Off) か。このプロパティは log_deadlocks オプションに相当します。
LogFreeCommit	Redo Free Commit の数。Redo Free Commit が発生するのは、トランザクションログのコミットが要求されているが、ログはすでに書き込まれている (コミットが実行されている) 場合です。
login_mode	サポートされているログインのタイプ。このプロパティは login_mode オプションに相当します。
login_procedure	起動時に互換性オプションを設定するスタアドプロシージャの名前。このプロパティは login_procedure オプションに相当します。
LoginTime	接続が確立された日付と時刻。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
LogWrite	トランザクションログに書き込まれたページの数。
materialized_view_optimization	クエリの最適化時にマテリアライズドビューを使用するかどうかを示す、接続の materialized_view_optimization オプションの値。
max_connections	データベースに対して許可する同時接続数を示す max_connections オプションの値。
max_client_statements_cached	クライアントでキャッシュされる文の数を示す max_client_statements_cached オプションの値。
max_cursor_count	1つの接続が同時に使用できるカーソルの最大数。このプロパティは max_cursor_count オプションに相当します。
max_hash_size	このプロパティは使用されなくなりました。

プロパティ名	説明
max_plans_cached	キャッシュに格納される実行プランの最大数。このプロパティは max_plans_cached オプションに相当します。
max_priority	接続に設定できる最高優先度レベル。このプロパティは接続の max_priority オプションに相当します。
max_query_tasks	データベースサーバがクエリの処理に使用できる要求の最大数。このプロパティは max_query_tasks オプションに相当します。
max_recursive_iterations	再帰共通テーブル式が反復できる最大回数。このプロパティは max_recursive_iterations オプションに相当します。
max_statement_count	接続で一度に使用できる準備文の最大数。このプロパティは max_statement_count オプションに相当します。
max_temp_space	1つの接続で使用できるテンポラリファイル領域の最大サイズ。このプロパティは接続の max_temp_space オプションに相当します。
MessageReceived	MESSAGE 文によって生成され、WAITFOR 文が中断する原因となった文字列。それ以外の場合は、空の文字列を返します。
min_password_length	データベースにおける新しいパスワードの最小長。このプロパティは min_password_length オプションに相当します。
min_role_admins	ロールに必要な管理者の最小数。このプロパティは min_role_admins オプションに相当します。
Name	現在の接続の名前。ConnectionName (CON) 接続パラメータを使用して接続名を指定できます。
NcharCharSet	接続で使用される NCHAR 文字セット。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
nearest_century	文字列から日付への変換において 2 桁の年がどのように解釈されるかを示す nearest_century オプションの値。
NodeAddress	クライアント/サーバ接続のクライアント側に対応するノード。クライアントとサーバの両方が同じコンピュータにある場合は、空の文字列を返します。
non_keywords	オフになっているため識別子として使用できるキーワードのリスト (存在する場合) である、non_keywords オプションの値。
NumLocalTempTables	接続によって使用されているローカルテンポラリテーブルの数。
Number	現在の接続の接続 ID (数値)。
odbc_describe_binary_as_varbinary	SQL Anywhere ODBC ドライバが BINARY と VARBINARY の両方のカラムを SQL_BINARY として記述する場合、値は Off になります。ODBC ドライバが BINARY と VARBINARY のカラムを SQL_VARBINARY として記述する場合、値は On になります。このプロパティは odbc_describe_binary_as_varbinary オプションに相当します。

プロパティ名	説明
odbc_distinguish_char_and_varchar	CHAR カラムを SQL_CHAR (On) として記述するか、SQL_VARCHAR (OFF) として記述するか。このプロパティは odbc_distinguish_char_and_varchar オプションに相当します。
oem_string	データベースファイルのヘッダページ内の文字列。このプロパティは oem_string オプションに相当します。
on_charset_conversion_failure	文字セットの変換中にエラーが発生した場合の動作。このプロパティは on_charset_conversion_failure オプションに相当します。
on_tsq_error	ストアプロシージャまたは T-SQL バッチの実行中にエラーが発生した場合の動作。このプロパティは on_tsq_error オプションに相当します。
optimization_goal	どのようにクエリ処理を最適化するか。このプロパティは optimization_goal オプションに相当します。
optimization_level	optimization_level オプションの値 (0 ~ 15)。この値を使用して、SQL Anywhere クエリオプティマイザが SQL 文のアクセスプランの検索に費やす作業レベルを指定します。
optimization_workload	SQL Anywhere クエリオプティマイザが SQL 文のアクセスプランの検索に費やす作業レベル。このプロパティは接続の optimization_workload オプションに相当します。
OSUser	クライアントプロセスに関連付けられたオペレーティングシステムユーザ名。クライアントプロセスで別のユーザを同一化している場合 (UNIX の場合は、セット ID ビットが設定されている場合)、同一化されたユーザ名が返されます。バージョン 10.0.1 以前のクライアントと HTTP や TDS クライアントでは、空の文字列を返します。
PacketSize	接続で使用されるパケットサイズ (バイト単位)。現在実行中の要求がイベントハンドラの一部である場合、値は NA になります。このプロパティは CommBufferSize (CBSIZE) 接続パラメータに相当します。
PacketsReceived	受信したクライアント/サーバ通信パケットの数。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したパケット数。この値は、圧縮が無効の場合は PacketsReceived の値と同じです。
PacketsSent	送信されたクライアント/サーバ通信パケットの数。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたパケット数。この値は、圧縮が無効の場合は PacketsSent の値と同じです。
parameterization_level	文のパラメータ化の動作を示す parameterization_level オプションの値。
ParameterizationPrepare Count	自動的にパラメータ化された文に対する準備の数。
ParentConnection	データベースオペレーション (データベースのバックアップや作成など) を実行するためにテンポラリ接続を作成した接続の接続 ID。その他のタイプの接続の場合、この値は NULL になります。

プロパティ名	説明
pinned_cursor_percent_of_cache	カーソルを固定するために使用するキャッシュの割合を示す pinned_cursor_percent_of_cache オプションの値。
post_login_procedure	ユーザが接続したときにアプリケーションが表示する必要があるメッセージがプロシージャの結果セットに含まれている場合のプロシージャの名前。このプロパティは post_login_procedure オプションに相当します。
precision	10 進数精度と数値精度の設定を示す precision オプションの値。
prefetch	prefetch オプションの値。プリフェッチが行われない場合、この値は Off になります。カーソルタイプが SENSITIVE か、クエリにプロキシテーブルが含まれないかぎり、プリフェッチが発生する場合、この値は Conditional になります。SENSITIVE タイプのカーソルでも、プロキシテーブルが関連するカーソルでも、プリフェッチが行われる場合、この値は Always になります。
Prepares	接続に対して実行された文の準備作業の数。
PrepStmt	現在データベースサーバが保持している準備文の数。
preserve_source_format	プロシージャ、トリガ、ビュー、イベントハンドラの元のソース定義がシステムファイルに保存される (On) か、保存されない (Off) か。このプロパティは preserve_source_format オプションに相当します。
prevent_article_pkey_update	パブリケーションで使用するテーブルのプライマリキーカラムの更新が許可されている (On) か、許可されていない (Off) か。このプロパティは prevent_article_pkey_update オプションに相当します。
priority	接続の優先度レベルを示す、接続の priority オプションの値。
Progress	クエリが実行されている時間に関する情報。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
progress_messages	progress_messages オプションの値。
query_mem_timeout	query_mem_timeout オプションの値。
QueryBypassed	オプティマイザバイパスによって最適化された要求の数。
QueryBypassedCosted	オプティマイザバイパスによってコストを使用して処理された要求の数。
QueryBypassedHeuristic	オプティマイザバイパスによってヒューリスティックを使用して処理された要求の数。
QueryBypassedOptimized	オプティマイザバイパスによって最初に処理されてから、オプティマイザによって完全に最適化された要求の数。
QueryCachedPlans	接続において現在キャッシュされているクエリ実行プランの数。
QueryCachePages	実行プランの保存に使用されるキャッシュページの数。
QueryDescribedBypass	オプティマイザバイパスによって処理された DESCRIBE 要求の数。
QueryDescribedOptimizer	オプティマイザによって処理された DESCRIBE 要求の数。

プロパティ名	説明
QueryHeapPages	クエリ処理 (ハッシュ操作およびソート操作) に使用されるキャッシュページの数。
QueryJHToJNLOptUsed	ハッシュジョインがネストループジョインに変換された回数。
QueryLowMemoryStrategy	メモリ不足が原因でサーバが実行プランを実行中に変更した回数。現在使用できるメモリがオプティマイザの推定よりも少ない、または実行プランが必要とするメモリがオプティマイザの推定よりも多い場合に、プランが変更されることがあります。
QueryMemActiveCurr	クエリメモリをアクティブに使用する要求の数。
QueryMemGrantFailed	要求がクエリメモリ待ちになり、取得できなかった合計回数。
QueryMemGrantGranted	要求に現在付与されているページ数。
QueryMemGrantRequested	要求がクエリメモリを取得しようとした合計回数。
QueryMemGrantWaited	要求がクエリメモリ待ちになった合計回数。
QueryMemGrantWaiting	クエリメモリ待ちになっている現在の要求の数。
QueryOpened	実行対象の OPEN 要求の数。
QueryOptimized	完全に最適化された要求の数。
QueryReused	プランキャッシュから再利用された要求の数。
QueryRowsFetched	この接続について、逐次スキャンまたはインデックススキャンによってベーステーブルから読み込まれたローの数。
QueryRowsMaterialized	クエリ処理中にワークテーブルに書き込まれるロー数。
quoted_identifier	二重引用符で囲まれた文字列を識別子 (On) として解釈するか、単なる文字列 (Off) として解釈するか。このプロパティは quoted_identifier オプションに相当します。
read_past_deleted	独立性レベル 1 または 2 の逐次スキャンにおいてコミットされていない削除ローが無視される (On) か、独立性レベル 1 または 2 の逐次スキャンがコミットされていない削除ローでブロックされるか (Off)。このプロパティは read_past_deleted オプションに相当します。
recovery_time	データベースサーバがシステム障害から回復するのに要する最長時間 (分単位)。このプロパティは recovery_time オプションに相当します。
Recursivelterations	再帰ユニオンの反復回数。
RecursivelterationsHash	再帰ハッシュジョインでハッシュ方式が使用された回数。
RecursivelterationsNested	再帰ハッシュジョインでネストループ方式が使用された回数。
RecursiveJNLMisses	再帰ハッシュジョインでのインデックスプローブキャッシュミス数。

プロパティ名	説明
RecursiveJNLProbes	再帰ハッシュジョインにおけるインデックスプローブの試行回数。
remote_idle_timeout	Web サービスのクライアントプロシージャと関数で許容される休止時間 (秒単位)。このプロパティは remote_idle_timeout オプションに相当します。
replicate_all	内部でのみ使用。
ReqCountActive	処理が完了した要求の数。RequestTiming サーバプロパティが Off に設定されている場合は NULL。
ReqCountBlockContention	接続がアトミックアクセスを待った回数。-zt オプションが指定されていない場合は NULL。
ReqCountBlockIO	接続が I/O 処理の完了を待った回数。-zt オプションが指定されていない場合は NULL。
ReqCountBlockLock	接続がロックを待った回数。-zt オプションが指定されていない場合は NULL。
ReqCountUnscheduled	接続がスケジューリングを待った回数。-zt オプションが指定されていない場合は NULL。
ReqStatus	要求のステータス。現在要求を処理していない場合、この値は Idle になります。この接続で他の処理が実行されており、使用可能なデータベースサーバワーカを待っている場合、この値は Unscheduled* になります。接続がブロックされ、I/O 処理の完了を待っている場合、この値は BlockedIO* になります。接続がブロックされ、共有データベースサーバデータ構造体へのアクセスを待っている場合、この値は BlockedContention* になります。接続がブロックされ、オブジェクトのロックの解放を待っている場合、この値は BlockedLock になります。接続で要求を実行している場合、この値は Executing になります。アスタリスク (*) が付いている値が返されるのは、-zt サーバオプションによってデータベースサーバで要求タイミング情報のロギングがオンになっている場合に限られます。要求タイミング情報のロギングが実行されていない場合は (デフォルト)、値は Executing として返されます。
ReqTimeActive	要求の処理にかかった時間 (秒単位)、または、-zt オプションが指定されていなかった場合は NULL。
ReqTimeBlockContention	アトミックアクセスを待っていた時間 (秒単位)、または、RequestTiming サーバプロパティが Off に設定されている場合は NULL。
ReqTimeBlockIO	I/O 処理の完了を待っていた時間 (秒単位)、または、-zt オプションが指定されていなかった場合は NULL。
ReqTimeBlockLock	ロックを待っていた時間 (秒単位)、または、-zt オプションが指定されていなかった場合は NULL。
ReqTimeUnscheduled	未スケジュール時間、または -zt オプションが指定されていなかった場合は NULL。
ReqType	最終要求のタイプ。接続が接続プールによってキャッシュされる場合、その ReqType 値は CONNECT_POOL_CACHE になります。
request_timeout	1 つの要求を実行できる最大時間を示す request_timeout オプションの値。
RequestsReceived	クライアント/サーバ通信要求またはラウンドトリップの数。PacketsReceived とは異なり、マルチパケット要求を 1 つの要求として数え、活性パケットを計数の対象から除外します。
reserved_connections	標準接続用に予約された接続数。このプロパティは reserved_connections オプションに相当します。

プロパティ名	説明
reserved_keywords	データベースで有効なデフォルト以外の予約キーワードのリストを指定する reserved_keywords オプションの値。
return_date_time_as_string	DATE、TIME、TIMESTAMP の値が文字列としてアプリケーションに返される (On) か、DATE、TIME、または TIMESTAMP データ型として返される (Off) か。このプロパティは return_date_time_as_string オプションに相当します。
Rlbc	処理されたロールバック要求の数。
rollback_on_deadlock	デッドロックが発生した場合、トランザクションを自動的にロールバックする (On) か、しない (Off) か。このプロパティは rollback_on_deadlock オプションに相当します。
RollbackLogPages	ロールバックログのページ数。
row_counts	ローの数が常に正確である (On) か、ローの数が通常は推定値である (Off) か。このプロパティは row_counts オプションに相当します。
scale	接続における 10 進数と数値の位取り。このプロパティは scale オプションに相当します。
secure_feature_key	このプロパティは使用されなくなりました。データベースサーバにおいて機能を有効または無効にするキーを格納する secure_feature_key オプションの値。このプロパティの値を選択すると、常に空の文字列が返されます。
ServerNodeAddress	クライアント/サーバ接続のサーバ側に対応するノード。クライアントとサーバの両方が同じコンピュータにある場合は、空の文字列を返します。現在実行中の要求がイベントハンドラの一部である場合、値は NA になります。
ServerPort	データベースサーバの TCP/IP ポート番号または 0。
SessionCreateTime	HTTP セッションが作成された時刻。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
SessionID	接続のセッション ID がある場合はそのセッション ID、それ以外の場合は空の文字列。
SessionLastTime	HTTP セッションにおける最後の要求の時刻。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
SessionTimeout	非アクティブ状態の HTTP セッションが維持される時間 (分単位)。
SnapshotCount	接続に関連付けられているスナップショットの数。
sort_collation	sort_collation オプションの値。この値は、ORDER BY 句が変更されていない場合は Internal、それ以外の場合は照合名または照合 ID になります。
SortMergePasses	ソート中に使用されたマージパスの数。
SortRowsMaterialized	ソートワークテーブルに書き込まれたロー数。
SortRunsWritten	ソート中に書き込まれたソート実行の数。

プロパティ名	説明
SortSortedRuns	実行の生成中に作成されたソート実行の数。
SortWorkTables	ソート用に作成されたワークテーブルの数。
sql_flagger_error_level	指定した規格にはない SQL への応答を制御する sql_flagger_error_level オプションの値。このプロパティは sql_flagger_error_level オプションに相当します。
sql_flagger_warning_level	sql_flagger_warning_level の値。このプロパティは sql_flagger_warning_level オプションに相当します。
st_geometry_asbinary_format	ジオメトリからバイナリへの空間データ値の変換方法。このプロパティは st_geometry_asbinary_format オプションに相当します。
st_geometry_astext_format	ジオメトリからテキストへの空間データ値の変換方法。このプロパティは st_geometry_astext_format オプションに相当します。
st_geometry_asxml_format	ジオメトリから XML への空間データ値の変換方法。このプロパティは st_geometry_asxml_format オプションに相当します。
st_geometry_describe_type	空間データの記述方法。このプロパティは st_geometry_describe_type オプションに相当します。
st_geometry_interpolation	ST_CircularString ジオメトリの内挿法設定。このプロパティは st_geometry_interpolation オプションに相当します。
st_geometry_on_invalid	ジオメトリが表面検証に失敗した場合の動作。このプロパティは st_geometry_on_invalid オプションに相当します。
StatementDescribes	DESCRIBE 要求によって処理された文の合計数。
StatementPostAnnotations	セマンティックなクエリ変換フェーズによって処理された文の数。
StatementPostAnnotationsSimple	セマンティックなクエリ変換フェーズによって処理されたが、一部のセマンティック変換が省略された文の数。
StatementPostAnnotationsSkipped	セマンティックなクエリ変換フェーズが完全に省略された文の数。
string_rtruncation	文字列がトランケートされた場合にエラーを発生させる (On) か、エラーを発生させない (Off) か。このプロパティは string_rtruncation オプションに相当します。
subsume_row_locks	データベースサーバがテーブルに対する個別ローロックを取得する (On) か、取得しない (Off) か。このプロパティは subsume_row_locks オプションに相当します。
suppress_tds_debugging	TDS デバッグ情報をデータベースサーバメッセージウィンドウに表示する (Off) か、表示しない (On) か。このプロパティは suppress_tds_debugging オプションに相当します。

プロパティ名	説明
synchronize_mirror_on_commit	データベースミラーサーバをコミット時に同期する (On) か、しない (Off) か。このプロパティは synchronize_mirror_on_commit オプションに相当します。
tds_empty_string_is_null	TDS 接続において、空の文字列を NULL として返す (On) か、1 つの空白文字を含む文字列を返す (Off) か。このプロパティは tds_empty_string_is_null オプションに相当します。
temp_space_limit_check	データベースサーバが接続で使用できるテンポラリ領域のサイズをチェックする (On) か、データベースサーバが接続で使用できる領域のサイズをチェックしない (Off) か。このプロパティは temp_space_limit_check オプションに相当します。
TempFilePages	接続で使用されるテンポラリファイルページの数。
TempTablePages	テンポラリテーブルで使用されるテンポラリファイルのページ数。
time_format	データベースから取り出される時刻の文字列形式。このプロパティは time_format オプションに相当します。
time_zone	データベースでタイムゾーンの計算に使用されるタイムゾーン。このプロパティは time_zone オプションに相当します。
time_zone_adjustment	接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある分数。このプロパティは time_zone_adjustment オプションに相当します。
timestamp_format	データベースから取り出したタイムスタンプの形式。このプロパティは timestamp_format オプションに相当します。
timestamp_with_time_zone_format	データベースから取り出した TIMESTAMP WITH TIME ZONE 値の形式。このプロパティは timestamp_with_time_zone_format オプションに相当します。
TimeZoneAdjustment	接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある分数。
TransactionStartTime	値は、COMMIT または ROLLBACK の後にデータベースが最初に変更された時刻を含む文字列になります。最後の COMMIT または ROLLBACK 以降にデータベースが変更されていない場合、空の文字列になります。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
truncate_timestamp_values	TIMESTAMP 値で使用される小数点以下の桁数に制限がある (On) か、ない (Off) か。このプロパティは truncate_timestamp_values オプションに相当します。
trusted_certificates_file	データベースサーバが LDAP サーバのクライアントとして機能するときに、信頼できる認証局の証明書のリストを含むファイル。このプロパティは trusted_certificates_file オプションに相当します。
tsql_outer_joins	DML 文で Transact-SQL 外部ジョインが使用できる (On) か、使用できない (Off) か。このプロパティは tsql_outer_joins オプションの値に相当します。
tsql_variables	Embedded SQL のホスト変数名のプレフィクスとしてコロン代わりに @ 符号を使用できる (On) か、使用できない (Off) か。このプロパティは tsql_variables オプションの値に相当します。

プロパティ名	説明
UncommitOp	コミットされていないオペレーションの数。
updatable_statement_isolation	isolation_level オプションが Readonly-statement-snapshot に設定されている場合に更新可能な文に使用される独立性レベル (0、1、2、または 3)。このプロパティは updatable_statement_isolation オプションに相当します。
update_statistics	接続が統計ガバナーにクエリフィードバックを送信できる (On) か、統計ガバナーが現在の接続からクエリフィードバックを受け取らない (Off) か。このプロパティは update_statistics オプションに相当します。
upgrade_database_capability	このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。
user_estimates	クエリオプティマイザがクエリの述部に含まれる選択性推定を尊重するか無視するかを制御する値。このプロパティは user_estimates オプションに相当します。
UserAppInfo	AppInfo 接続パラメータによって接続文字列に指定された文字列。
UserDefinedCounterRate01	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRate02	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRate03	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRate04	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRate05	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRaw01	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRaw02	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRaw03	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRaw04	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserDefinedCounterRaw05	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。
UserID	接続のユーザ ID。

プロパティ名	説明
UtilCmdsPermitted	接続で CREATE DATABASE、DROP DATABASE、RESTORE DATABASE などの SQL 文が許可されているかどうか。指定した接続 ID が現在の接続のものでない場合、値は空の文字列になります。
uuid_has_hyphens	ユニークな識別子の値を文字列に変換するときのフォーマット。このオプションを On に設定すると、変換後の文字列には 4 つのハイフンが挿入されます。このプロパティは uuid_has_hyphens オプションに相当します。
verify_password_function	パスワードの検証に使用される関数が指定されている場合の関数名。このプロパティは verify_password_function に相当します。
wait_for_commit	データベースが次の COMMIT 文まで外部キー整合性をチェックしない (On) か、CHECK ON COMMIT 句で作成されていないすべての外部キーが挿入、更新、削除時にチェックされる (Off) か。このプロパティは wait_for_commit オプションに相当します。
WaitStartTime	接続が待機を始めた時刻 (接続が待機していない場合は空の文字列)。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
WaitType	待機の理由 (ある場合)。接続がロックの解放を待っている場合、値は lock になります。接続が waitfor 文を実行している場合、値は waitfor になります。接続が待機していないか、待機の理由が不明な場合、値は空の文字列になります。
webservice_namespace_host	生成された WSDL ドキュメント内で XML ネームスペースとして使用されるホスト名 (そのようなホスト名が指定されている場合)。このプロパティは webservice_namespace_host オプションに相当します。
webservice_sessionid_name	セッション管理が使用されているかどうかを判別するときに Web サーバが使用するセッションの識別子名。このプロパティは webservice_sessionid_name オプションに相当します。

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[-ze データベースサーバオプション \[510 ページ\]](#)

[データベースプロパティの一覧 \[874 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

## 1.5.7.2 データベースサーバプロパティの一覧

データベースサーバプロパティは、データベースへの各接続で使用できます。PROPERTY システム関数を使用して個々のプロパティの値を取得し、sa\_eng\_properties システムプロシージャを使用してすべてのデータベースサーバプロパティの値を取得します。

### 例

次の文は、グローバルサーバデータ構造に使用されたキャッシュページ数を返します。

```
SELECT PROPERTY ( 'MainHeapPages' );
```

sa\_eng\_properties システムプロシージャを使用して、すべてのデータベースサーバプロパティの値を取り出します。

```
CALL sa_eng_properties;
```

## データベースサーバプロパティ

プロパティ名	説明
ActiveReq	現在クライアント側の要求を処理中のサーバワーカの数。
ApproximateCPUTime	データベースサーバで使用された累積 CPU 時間の概算値 (秒単位)。この値と実際の CPU 時間との誤差は、通常は 5 ~ 10 % ですが、最大で 50 % に達することもあります。マルチプロセッサコンピュータでは、CPU (ハイパースレッドまたはコア) ごとに使用時間が積算されるため、すべての接続の累積時間を合計した値が実際の経過時間より大きくなる場合もあります。
AutoMultiProgrammingLevel	データベースサーバがマルチプログラミングレベルを自動的に調整しているかどうか。
AutoMultiProgrammingLevelStatistics	データベースサーバのマルチプログラミングレベルの自動調整に関するメッセージが、データベースサーバのメッセージログに表示されるかどうか。
AvailIO	現在使用可能な I/O 制御ブロックの数。
BuildChange	システムで使用するために予約済み。
BuildClient	システムで使用するために予約済み。このプロパティの設定は変更しないでください。
BuildProduction	データベースサーバが、実際の運用で使用するようにコンパイルされている (Yes) か、デバッグ用のビルドである (No) か。
BuildReproducible	システムで使用するために予約済み。
BytesReceived	クライアント/サーバ通信中に受信したバイト数。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。

プロパティ名	説明
BytesReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したバイト数。この値は、圧縮が無効の場合は BytesReceived の値と同じです。
BytesSent	クライアント/サーバ通信中に送信されたバイト数。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。
BytesSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたバイト数。この値は、圧縮が無効の場合は BytesSent の値と同じです。
CacheAllocated	サーバデータ構造に割り付けられたキャッシュページの数。
CacheFile	データベースファイルから取得したデータを格納するキャッシュページの数。
CacheFileDirty	ダーティな (書き込みが必要な) キャッシュページの数。
CacheFree	未使用のキャッシュページ数。
CacheHits	データベースページのルックアップ回数。
CacheMisses	ページがキャッシュになかった回数。
CachePanics	キャッシュマネージャが割り付けるページの検索に失敗した回数。
CachePinned	固定されたキャッシュページの数。
CacheRead	キャッシュの読み込み数。
CacheReplacements	キャッシュ内の置換されたページの数。
CacheScavenges	キャッシュマネージャが割り付けるページをスカベンジした回数。
CacheScavengeVisited	割り付けるページのスカベンジ中にアクセスしたページの数。
CacheSizingStatistics	キャッシュのサイズを変更するときにデータベースサーバがキャッシュサイズ設定の統計を表示するかどうか。
CarverHeapPages	クエリ最適化などの短期的な目的に使用されたヒープページの数。
CharSet	データベースサーバが使用している CHAR 文字セット。
ClientStmtCacheHits	クライアントに文がキャッシュされていることが理由で、不要な準備文の数。これは、クライアントでの文のキャッシュが無効になった場合に必要となる、追加の準備文の数です。
ClientStmtCacheMisses	クライアントのキャッシュに存在した文の中で、再び準備された文の数。これは、キャッシュされた文が再使用できるか検討されたものの、スキーマの変更、データベースのオプション設定の変更、DROP VARIABLE 文によって再使用できなかった回数です。
CockpitDB	データベースサーバで現在使用されている一連のオプションとテンポラリパラメータを一覧表示します。Temp が "はい" に設定されている場合、コックピットデータベースはテンポラリデータベースです。

プロパティ名	説明
CockpitURL	コックピットの有効な URL をセミコロンで区切ったリスト。
CollectStatistics	データベースサーバがパフォーマンス統計を収集するかどうか。
CommandLine	データベースサーバの起動に使用されたコマンドライン引数。 -ek オプションを使用してデータベース用の暗号化キーを指定した場合、キーはこのプロパティの値にあるアスタリスクの定数文字列で置き換えられます。
Commit	処理されたコミット要求の数。
CompactPlatformVer	PlatformVer プロパティの縮小バージョン。
CompanyName	このソフトウェアを所有している会社の名前。
CompletedReq	完了した要求の数。
ConnCount	データベースサーバとの接続の数。このプロパティ値には、内部処理に使用されている接続は含まれませんが、イベントと外部環境サポートに使用されている接続は含まれます。
ConnectedTime	すべての接続がこれまでデータベースサーバに接続してきた合計の時間数 (秒単位)。 接続に対する要求が完了したとき、または接続が切断されたときにだけ値が更新されます。その結果、値は、データベースサーバ内で長いことアイドル状態であるか、実行中でビジーである接続に対しては遅れる可能性があります。値には、データベースイベントやバックグラウンドサーバ接続 (データベースクリーナなど) といった任意の接続によって増えた時間も含まれます。
ConnsDisabled	サーバオプションが新しい接続を無効にするように設定されているかどうか。
ConsoleLogFile	-o オプションが指定されている場合にデータベースサーバメッセージがロギングされるファイルの名前。-o オプションが指定されていない場合、値は空の文字列になります。
ConsoleLogMaxSize	データベースサーバメッセージのロギングに使用するファイルの最大サイズ (バイト単位)。
CurrentCacheSize	現在のキャッシュサイズ (キロバイト単位)。
CurrentMirrorBackgroundWorkers	データベースミラーリングのバックグラウンドタスクで現在使用されているワーカの数。これらのワーカは、マルチプログラミングレベルで制御されるワーカとは別のものです。
CurrentMultiProgrammingLevel	データベースサーバが同時に処理できる現在のタスク数。
CurrRead	データベースサーバが発行し、まだ完了していない現在のファイル読み込み数。
CurrWrite	データベースサーバが発行し、まだ完了していない現在のファイル書き込み数。
Cursor	現在データベースサーバが保持している、宣言されたカーソルの数。
CursorOpen	現在データベースサーバが保持している、開いたカーソルの数。
DebuggingInformation	サーバがトラブルシューティング用の診断メッセージを表示するかどうか。

プロパティ名	説明
DefaultCollation	新規データベースに使用される照合 (明示的に指定されている照合がない場合)。
DefaultNcharCollation	サーバコンピュータのデフォルト NCHAR 照合の名前 (ICU がインストールされている場合は UCA、インストールされていない場合は UTF8BIN)。
DiskRead	ディスクの読み込み数。
DiskReadHintScatterLimit	分散読み込みヒントのサイズ制限 (バイト単位)。
DiskRetryRead	ディスクの読み込みリトライ数。
DiskRetryReadScatter	分散読み込みのためのディスク読み込みリトライ数。
DiskRetryWrite	ディスクの書き込みリトライ数。
DiskSandbox	データベースの読み込み/書き込みファイル操作が、メインのデータベースファイルがあるディレクトリに制限されている (On) か、制限されていない (Off) か。
DiskWrite	ディスクに書き込まれた修正ページの数。
EventTypeDesc	特定のイベントタイプ ID に関連付けられたイベントタイプの説明。
EventTypeName	特定のイベントタイプ ID に関連付けられたシステムイベントタイプの名前。
ExchangeTasks	クエリの並列実行に現在使用されているタスクの数。
ExchangeTasksCompleted	データベースサーバが起動された後、クエリ内並列処理に使用された内部タスクの総数。
FipsMode	データベースサーバの起動時に -fips オプションが使用されたかどうか。
FirstOption	データベースオプションに対応する最初の接続プロパティを表す番号。
FreeBuffers	使用できるネットワークバッファの数。
FunctionMaxParms	関数で指定可能なパラメータの最大数。関数は、正の整数である <code>function-number</code> によって指定される値で識別されます。次に例を示します。 <pre>SELECT PROPERTY ( 'FunctionMaxParms', function-number );</pre> <code>function-number</code> は、リリース間で変更されることがあります。
FunctionMinParms	関数で指定しなければならないパラメータの最小数。関数は、正の整数である <code>function-number</code> によって指定される値で識別されます。次に例を示します。 <pre>SELECT PROPERTY ( 'FunctionMinParms', function-number );</pre> <code>function-number</code> は、リリース間で変更されることがあります。

プロパティ名	説明
FunctionName	<p>正の整数である <code>function-number</code> によって指定される値で識別される関数名。</p> <pre>SELECT PROPERTY ( 'FunctionName', function-number );</pre> <p><code>function-number</code> は、リリース間で変更されることがあります。</p>
HasSecuredFeature	<p>すべての機能セットのうち 1 つ以上の機能がグローバルサーバレベルで保護されているか (Yes)、保護されていないか (No)。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。</p>
HasSecureFeatureKey	<p>データベースに 1 つ以上のセキュリティ機能キーがあるかどうか。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。</p>
HeapsCarver	クエリ最適化などの短期的な目的に使用されたヒープの数。
HeapsLocked	キャッシュ内で現在ロックされている再配置可能なヒープの数。
HeapsQuery	クエリ処理 (ハッシュ操作およびソート操作) に使用されるヒープの数。
HeapsRelocatable	再配置可能なヒープの数。
HttpAddresses	<p>データベースサーバがクライアントからの HTTP 接続で受信するセミコロンで区切られた IP アドレスのリスト。次に例を示します。</p> <pre>(::1):80;127.0.0.1:80</pre>
HttpConnectionsQueued	現在キューに入っている接続の数。
HttpListeners	データベースサーバが HTTP 接続で受信するために使用している IP address:port の組み合わせのセミコロンで区切られたリスト。
HttpNumActiveReq	HTTP 要求をアクティブに処理している HTTP 接続数。応答を送信した HTTP 接続は含まれません。
HttpNumConnections	データベースサーバで現在開いている HTTP 接続数。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。
HttpNumSessions	データベースサーバ内のアクティブおよび休止状態の HTTP セッション数。
HttpPorts	Web サーバの HTTPS ポート番号のカンマで区切られたリスト。
HttpQueueCount	データベースサーバが起動されてからキューに入れられた接続の合計数。
HttpQueueMaxCount	一度にキューに入れられた接続の最大数。
HttpQueueTimedOut	キューに入れられた後タイムアウトした接続の合計数。

プロパティ名	説明
HttpsAddresses	<p>サーバがクライアントからの HTTPS 接続で受信する IP アドレスのセミコロンで区切られたリスト。次に例を示します。</p> <pre>(::1):443;127.0.0.1:443</pre>
HttpsListeners	データベースサーバが HTTPS 接続で受信するために使用している IP address:port の組み合わせのセミコロンで区切られたリスト。
HttpsNumActiveReq	HTTPS 要求をアクティブに処理しているセキュア HTTPS 接続数。応答を送信した HTTPS 接続は含まれません。
HttpsNumConnections	データベースサーバで現在開いている HTTPS 接続数。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。
HttpsPorts	Web サーバの HTTP ポート番号のカンマで区切られたリスト。
IdleTimeout	デフォルトのアイドルタイムアウト。
IPAddressMonitorPeriod	データベースサーバが新しい IP アドレスをチェックする時間 (秒単位)。
IsAesniAvailable	データベースサーバコンピュータの CPU が Intel AES-NI 命令セットをサポートしており、このコンピュータが、サポートされているオペレーティングシステムを実行しているかどうか。
IsFipsAvailable	FIPS 認定の DLL がインストールされているかどうか。
IsIQ	システムで使用するために予約済み。
IsNetworkServer	接続先がネットワークデータベースサーバ (Yes) か、パーソナルデータベースサーバ (No) か。
IsPortableDevice	<p>データベースサーバがラップトップ、ノートブック、またはその他のポータブルデバイスで実行されているかどうか。</p> <p>VMWare は考慮されないため、ラップトップで実行されている VM で実行中のデータベースサーバの場合、値は No になります。</p> <p>Windows では、デバイスがポータブルかどうかを判別できない場合、値は NULL になります。</p> <p>UNIX の場合、このプロパティは常に NULL になります。</p>
IsRsaAvailable	RSA DLL がインストールされているかどうか。
IsRuntimeServer	このプロパティは、データベースサーバのすべてのバージョンについて No になります。
IsService	データベースサーバがサービスとして実行されているかどうか。
JavaVM	データベースサーバで、データベースごとに 1 つの Java VM を使用する場合、空の文字列。データベースサーバで、すべてのデータベースに対して 1 つの Java VM を使用する場合、このプロパティは JAVA 実行可能ファイルへのパスを示します。
Language	サーバのロケール言語。

プロパティ名	説明
LastConnectionProperty	最後の接続プロパティを表す番号。
LastDatabaseProperty	最後のデータベースプロパティを表す番号。
LastOption	データベースオプションに対応する最後の接続プロパティを表す番号。
LastServerProperty	最後のサーバプロパティを表す番号。
LegalCopyright	ソフトウェアの著作権を示す文字列。
LegalTrademarks	ソフトウェアの商標を示す文字列。
LicenseCount	ライセンスされたシートまたはプロセッサの数。
LicensedCompany	ライセンスされた会社の名前。
LicensedUser	ライセンスされたユーザの名前。
LicenseKey	システムで使用するために予約済み。
LicenseType	ライセンスタイプ。ネットワークシート (per-seat) または CPU ベースのいずれかです。
LivenessTimeout	クライアント活性タイムアウトのデフォルト。
LockedCursorPages	メモリ内でカーソルヒープを固定するために使用されているページ数。
LockedHeapPages	キャッシュ内でロックされているヒープページの数。
MachineName	データベースサーバを実行しているコンピュータの名前。通常は、コンピュータのホスト名です。
MainHeapBytes	グローバルサーバデータ構造に使用されているバイト数。
MainHeapPages	グローバルサーバデータ構造に使用されているページの数。
MapPhysicalMemoryEng	Address Windowing Extensions を使用して、キャッシュ内の物理メモリにマッピングされているデータベースページのアドレス領域ウィンドウの数。
MaxCacheSize	許容最大キャッシュサイズ (キロバイト単位)。
MaxConnections	データベースサーバが許容する同時接続の最大数。ネットワークデータベースサーバの場合、デフォルトはライセンスによって異なります。デフォルト値を小さくするには、-gm サーバオプションを使用します。パーソナルデータベースサーバの場合、デフォルト設定は 10 です。
MaxEventType	有効な最大イベントタイプ ID。
MaxMessage	このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウから取得できる行番号の現在の最大値。この値は、データベースサーバメッセージウィンドウに最後に表示されたメッセージを表します。
MaxMirrorBackgroundWorkers	サーバが起動してからデータベースミラーリングのバックグラウンドタスクで使用されたワーカの最大数。これらのワーカは、マルチプログラミングレベルで制御されるワーカとは別のものです。

プロパティ名	説明
MaxMultiProgrammingLevel	データベースサーバが同時に処理できるタスクの最大数。AutoMultiProgrammingLevel が有効になると、サーバはマルチプログラミングレベルをこの値まで上げることができます。
MaxRemoteCapability	有効な最大機能 ID。
Message. <b>linenumber</b>	データベースサーバメッセージウィンドウから取得したメッセージ行 (行の先頭にはメッセージが表示された日付と時刻が追加される)。2 番目のパラメータは行番号を指定します。このプロパティは使用されなくなりました。  PROPERTY ( "message" ) の値は、データベースサーバメッセージウィンドウに書き込まれた出力の最初の行です。PROPERTY ( "message", n ) を呼び出した場合、サーバ出力の n 行目 (0 を最初の行とする) です。バッファは有限であるため、メッセージの生成とともに最初の行が削除されていき、メモリからなくなる場合があります。この場合、値は NULL になります。
MessageCategoryLimit	sa_server_messages システムプロシージャを使用して取り出すことのできる重大度レベルとカテゴリそれぞれのメッセージの最小数。デフォルト値は 400 です。
MessageText, <b>linenumber</b>	このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウ内の指定された行番号に対応するテキスト (日付と時刻を含まない)。2 番目のパラメータは行番号を指定します。
MessageTime, <b>linenumber</b>	このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウ内の指定された行番号に対応する日付と時刻。2 番目のパラメータは行番号を指定します。
MessageWindowSize	このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウから取得できる行の最大数。
MinCacheSize	許容最小キャッシュサイズ (キロバイト単位)。
MinMultiProgrammingLevel	サーバが同時に処理できるタスクの最小数。AutoMultiProgrammingLevel が有効になると、サーバはマルチプログラミングレベルをこの値まで下げることができます。
MultiPacketsReceived	クライアント/サーバ通信中に受信したマルチパケット要求の数。
MultiPacketsSent	クライアント/サーバ通信中に送信されたマルチパケット応答の数。
MultiPageAllocs	マルチページキャッシュ割り付けの数。
MultiProgrammingLevel	サーバが同時に処理できる同時タスクの現在の最大数。この値を超える数の同時タスクが存在する場合、要求はキューイングされます。この動作を変更するには、-gn サーバオプションを使用します。
Name	データベースへの接続に使用するサーバの別名 (別名が指定されている場合)。別名が指定されていない場合、実際のサーバ名。  クライアントがコピーノードに接続され、接続文字列で NodeType=COPY が指定されている場合、このプロパティの値は、ServerName 接続パラメータでクライアントの接続文字列に指定されたデータベースサーバ名と異なることがあります。

プロパティ名	説明
NativeProcessorArchitecture	<p>ソフトウェアが実行されているネイティブプロセッサタイプを表す文字列。プロセッサをエミュレートできるプラットフォーム (x64 や x86 など) の場合は、OS アーキテクチャタイプではなく、実際のプロセッサタイプが返されます。</p> <p>この値は、オペレーティングシステムが 32 ビットと 64 ビットのいずれであるかを示すものではありません。</p> <p>値には、次のものがあります。</p> <ul style="list-style-type: none"> <li>• Windows - X86 または X86_64</li> <li>• Solaris - SPARC または X86_64</li> <li>• AIX - PPC</li> <li>• HP - IA64</li> <li>• Linux - X86 または X86_64</li> <li>• MAC OS - X86_64</li> </ul> <p>X86 は 32 ビットハードウェアアーキテクチャを表します。X86_64 は 64 ビットハードウェアアーキテクチャを表します。</p>
NumLogicalProcessors	サーバコンピュータ上の論理プロセッサ数 (コアとハイパースレッドを含む)。
NumLogicalProcessorsUsed	データベースサーバが使用する論理プロセッサの数。Windows の場合、使用する論理プロセッサ数を変更するには、-gtc オプションを使用します。
NumPhysicalProcessors	サーバコンピュータ上で有効な物理プロセッサ数。この値は、NumLogicalProcessors の値を物理プロセッサあたりのコア数またはハイパースレッド数で割った数に相当します。Windows 以外のプラットフォームでは、コアまたはハイパースレッドが物理プロセッサとして計数される場合があります。
NumPhysicalProcessorsUsed	データベースサーバが使用する物理プロセッサの数。Windows でネットワークデータベースサーバが使用する物理プロセッサの数を変更するには、-gt オプションを使用します。
ObjectType	データベースオブジェクトのタイプ。この値は、SYSOBJECT システムビューによって使用されます。
ODataAddresses	OData サーバが OData 接続で受信するために使用している TCP/IP とポートのセミコロンで区切られたリスト。
ODataSecureAddresses	OData サーバが安全な OData 接続で受信するために使用している TCP/IP とポートのセミコロンで区切られたリスト。
OmnIdentifier	システムで使用するために予約済み。
PacketsReceived	受信したクライアント/サーバ通信パケットの数。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したパケット数。この値は、圧縮が無効の場合は PacketsReceived の値と同じです。
PacketsSent	送信されたクライアント/サーバ通信パケットの数。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。

プロパティ名	説明
PacketsSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたパケット数。この値は、圧縮が無効の場合は PacketsSent の値と同じです。
PageSize	データベースサーバのキャッシュページのサイズ。-gp オプションを使用して設定可能です。設定しない場合はコマンドラインで指定したデータベースの最大ページサイズです。
ParameterizationPrepareCount	自動的にパラメータ化された文に対する準備の数。
PeakCacheSize	現在のセッションでキャッシュが到達した最大値 (キロバイト単位)。
PlanStatisticsStored	内部でのみ使用。
PlanStatisticsStoredMax	内部でのみ使用。
Platform	ソフトウェアを実行しているオペレーティングシステム。
PlatformVer	ソフトウェアを実行しているオペレーティングシステムの名前 (ビルド番号やサービスパックなどの情報も含む)。
PrepStmt	現在データベースサーバが保持している準備文の数。
ProcessCPU	データベースサーバプロセスにおける CPU の使用量。値は秒数単位です。このプロパティは Windows と UNIX コンピュータでサポートされています。  この値はデータベースサーバが起動してからの累積値です。この値は Windows タスクマネージャや Windows パフォーマンスモニタなどのアプリケーションに表示される瞬時値とは一致しません。
ProcessCPUSystem	データベースサーバプロセス CPU における CPU の使用量。これは、データベースサーバがオペレーティングシステムカーネルの内部で使用した CPU 時間を表します。値は秒数単位です。このプロパティは Windows と UNIX コンピュータでサポートされています。  この値はデータベースサーバが起動してからの累積値です。この値は Windows タスクマネージャや Windows パフォーマンスモニタなどのアプリケーションに表示される瞬時値とは一致しません。
ProcessCPUUser	データベースサーバプロセスにおけるユーザ CPU の使用量。値は秒数単位です。これには、データベースサーバがオペレーティングシステムカーネルの内部で使用した CPU 時間は含まれません。このプロパティは Windows と UNIX コンピュータでサポートされています。  この値はデータベースサーバが起動してからの累積値です。この値は Windows タスクマネージャや Windows パフォーマンスモニタなどのアプリケーションに表示される瞬時値とは一致しません。
ProcessID	データベースサーバプロセスのプロセス ID。
ProcessorAffinity	-gta オプション、または sa_server_option システムプロシージャと ProcessorAffinity オプションによる指定に従ってデータベースサーバが使用する論理プロセッサ。

プロパティ名	説明
ProcessorArchitecture	現在のソフトウェアが構築された対象のプロセッサタイプを表す文字列。次のような値があります。 <ul style="list-style-type: none"> <li>Windows - X86 または X86_64</li> <li>Solaris - SPARC または X86_64</li> <li>AIX - PPC</li> <li>HP - IA64</li> <li>Linux - X86 または X86_64</li> <li>MAC OS - X86_64</li> </ul> X86 は 32 ビットデータベースサーバを表します。X86_64 は 64 ビットデータベースサーバを表します。
ProductName	ソフトウェアの名前。
ProductVersion	実行中のソフトウェアのバージョン。
ProfileFilterConn	特定の接続をプロファイルするプロシージャがオンの場合にモニタされる接続の ID。プロファイルがオフの場合、値は空の文字列になります。
ProfileFilterUser	特定のユーザのプロシージャプロファイルがオンの場合にモニタされるユーザ ID。特定のユーザのプロシージャプロファイルがオフの場合、値は空の文字列になります。
PropertyHistoryList	追跡されている現在の最小プロパティセット。
PropertyHistoryListActual	追跡されている現在のプロパティセット。
PropertyHistorySize	追跡されたプロパティ値を保存する最小時間、または追跡されたプロパティ値の保存に使用するメモリの最大量を示します。
PropertyHistorySizeBytes	プロパティの履歴追跡に現在使用されている現在のメモリ量 (バイト単位)。
QueryHeapPages	クエリ処理 (ハッシュ操作およびソート操作) に使用されるキャッシュページの数。
QueryMemActiveCurr	クエリメモリをアクティブに使用する要求の数。
QueryMemActiveEst	安定した状態のデータベースサーバで、クエリメモリをアクティブに使用する要求の推定平均数。
QueryMemActiveMax	クエリメモリをアクティブに使用できる要求の最大数。
QueryMemExtraAvail	基本的なメモリ集約型の付与を超えて付与可能なメモリ量。
QueryMemGrantBase	すべての要求に付与される最小メモリ量。
QueryMemGrantBaseMI	メモリ集約型の要求に付与される最小メモリ量。
QueryMemGrantExtra	アクティブなメモリ集約型の要求間で QueryMemGrantBaseMI の値を超えて分配されるクエリメモリページの数。
QueryMemGrantFailed	要求がクエリメモリ待ちになり、取得できなかった合計回数。
QueryMemGrantGranted	要求に現在付与されているページ数。

プロパティ名	説明
QueryMemGrantRequested	要求がクエリメモリを取得しようとした合計回数。
QueryMemGrantWaited	要求がクエリメモリ待ちになった合計回数。
QueryMemGrantWaiting	クエリメモリ待ちになっている現在の要求の数。
QueryMemPages	ページ数で表されるクエリ実行アルゴリズムで使用可能なメモリの容量。
QueryMemPercentOfCache	最大キャッシュサイズの割合 (パーセント) で表されるクエリ実行アルゴリズムで使用可能なメモリの容量。
QuittingTime	サーバのシャットダウン時間。シャットダウン時間が指定されていない場合、値は none になります。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
RememberLastPlan	オプティマイザが返す最後のクエリ最適化プランをデータベースサーバが記録しているかどうか。
RememberLastStatement	サーバが各接続で準備された最後の文を記録しているかどうか。
RemoteCapability	特定の機能 ID に関連付けられたリモート機能名。
RemoteputWait	通信パケットの送信中にサーバがブロックしなければならなかった回数。通常、ブロックはクライアントまたはネットワークが受信可能な速度を超えてデータベースサーバがデータを送信した場合にのみ起こりません。エラー条件は示しません。
Req	新しい要求を処理するか、または既存の要求の処理を続行するよう、サーバが要求された回数。
ReqCountActive	アクティブな要求の数。
ReqCountBlockContention	内部サーバのリソースの競合により接続がブロックされた回数。
ReqCountBlockIO	IO 要求の完了を待つ間に接続がブロックされた回数。
ReqCountBlockLock	別の接続が保持しているローロックを待つ間に接続がブロックされた回数。
ReqCountUnscheduled	サーバスレッドが接続を処理するのを待つ間に接続がブロックされた回数。
ReqTimeActive	サーバが要求の処理で直接費やした合計時間。
ReqTimeBlockContention	内部サーバのリソースの競合により接続がブロックされた合計時間。
ReqTimeBlockIO	IO 要求の完了を待つ間に接続がブロックされた合計時間。
ReqTimeBlockLock	別の接続が保持しているローロックを待つ間に接続がブロックされた合計時間。
ReqTimeUnscheduled	サーバスレッドが接続を処理するのを待つ間に接続がブロックされた合計時間。
RequestFilterConn	ロギング情報をフィルタしている接続の ID。フィルタを実行していない場合、値は -1 になります。
RequestFilterDB	ロギング情報をフィルタしているデータベースの ID。フィルタを実行していない場合、値は -1 になります。

プロパティ名	説明
RequestLogFile	要求ロギングファイルの名前、または要求ロギングが存在しない場合は空の文字列。
RequestLogging	現在の要求ロギングの設定。値は、SQL、PLAN、HOSTVARS、PROCEDURES、TRIGGERS、OTHER、BLOCKS、REPLACE、ALL、NONE のいずれかです。
RequestLogMaxSize	要求ログファイルの最大サイズ。
RequestLogNumFiles	保管される要求ログファイルの数。
RequestsReceived	クライアント/サーバ通信要求またはラウンドトリップの数。PacketsReceived とは異なり、マルチパケット要求を1つの要求として数え、活性パケットを計数の対象から除外します。
RequestTiming	要求タイミング情報のロギングをオンにするかどうか。要求タイミング情報のロギングをオンにするには、-zr データベースサーバオプションを使用します。
Rlbc	処理されたロールバック要求の数。
SendFail	通信プロトコルがパケット送信に失敗した回数。
ServerEdition	<p>データベースサーバのタイプを記述する単語のスペースで区切られたリスト。次のような値があります。</p> <ul style="list-style-type: none"> <li>• Evaluation</li> <li>• Developer</li> <li>• Web</li> <li>• Educational</li> <li>• Standard</li> <li>• Advanced</li> <li>• Workgroup</li> <li>• OEM</li> <li>• Authenticated</li> </ul> <p>次のいずれかの機能のためのライセンスを別途保有している場合は、次の該当する文字列がライセンス文字列値に追加されます。</p> <ul style="list-style-type: none"> <li>• HighAvailability</li> <li>• InMemory</li> <li>• FIPS</li> </ul>
ServerName	実際のサーバ名 (サーバの代替名でない)。この値によって、稼働しているサーバのうちどれがデータベースミラーリング構成のプライマリサーバであるかを特定できます。
SharedMemoryListener	データベースサーバが共有メモリ接続を許容しているかどうか。許容していない場合は No。
SingleCLR	データベースサーバが、すべてのデータベースに対して1つの CLR 外部環境を使用する場合は CLR のバージョン番号、CLR スタアドプロシージャの実行時にデータベースごとに1つの CLR 外部環境を使用する場合は NONE。
SingleJVM	データベースサーバが、データベースサーバで実行されるすべてのデータベースに対して1つの Java VM を使用する (Yes)か、Java スタアドプロシージャの実行時にデータベースごとに1つの Java VM を使用する (No) か。

プロパティ名	説明
StartDBPermission	-gd サーバオプションの設定 (DBA、all、none のいずれか)。
StartTime	サーバが起動された日付/時刻。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
StreamsUsed	使用中のデータベースサーバストリーム数。
TcpIpAddresses	サーバがクライアントからの Command Sequence 接続または TDS 接続で受信する IP アドレスのセミコロンで区切られたリスト。次に例を示します。  (::1):2638;127.0.0.1:2638
TcpIpListeners	データベースサーバが TCP/IP 接続で受信するために使用している IP アドレスと、IP アドレス:ポートの組み合わせのセミコロンで区切られたリスト。
TempDir	テンポラリファイルが格納されているサーバ上のディレクトリ。
ThreadDeadlocksAvoided	スレッドデッドロックエラーが検出されたがクライアントアプリケーションにレポートされなかった回数。データベースサーバが起動すると、このプロパティの値は 0 になります。  スレッドデッドロックエラーを防ぐため、データベースサーバはマルチプログラミングレベルを動的に上げます。マルチプログラミングレベルを上げることができない場合、スレッドデッドロックエラーはクライアントアプリケーションに返され、ThreadDeadlocksReported プロパティが増分されます。  パーソナルサーバ上ではマルチプログラミングレベルの動的なチューニングがデフォルトで無効になっているため、ThreadDeadlocksAvoided プロパティは常に 0 になります。
ThreadDeadlocksReported	スレッドデッドロックエラーがクライアントアプリケーションにレポートされた回数。データベースサーバが起動すると、このプロパティの値は 0 になります。
TimeZoneAdjustment	サーバのローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある時間 (分単位)。
TopologyAwareScheduling	トポロジを考慮したスケジューリングを使用しているかどうか。
TotalBuffers	ネットワークバッファの総数。
UniqueClientAddresses	ネットワークサーバに接続しているユニークなクライアントネットワークアドレスの数 (共有メモリとローカル TCP/IP 接続を除く)。パーシートライセンスで現在使用されるシートの数。
UnschReq	使用できるサーバワーカ解放を待ってキューイングされている要求の数。
UserDefinedCounterRate01	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate02	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。

プロパティ名	説明
UserDefinedCounterRate03	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate04	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate05	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRaw01	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw02	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw03	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw04	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw05	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
WebClientLogFile	Web サービスクライアントログファイルの名前。
WebClientLogging	Web サービスクライアント情報がファイルにロギングされているかどうか。

## 関連情報

[データベースサーバプロパティの追跡 \[888 ページ\]](#)

[ユーザ定義プロパティ \[888 ページ\]](#)

[データベースサーバ起動オプション \[375 ページ\]](#)

[接続プロパティの一覧 \[839 ページ\]](#)

[データベースプロパティの一覧 \[874 ページ\]](#)

## 1.5.7.3 データベースプロパティの一覧

データベースプロパティは、データベースへの各接続で使用できます。DB\_PROPERTY システム関数を使用して個々のデータベースプロパティの値を取得するか、sa\_db\_properties システムプロシージャを使用してすべてのデータベースプロパティの値を取得します。

### 例

次の文は、現在のデータベースのページサイズを返します。

```
SELECT DB_PROPERTY ( 'PageSize' );
```

sa\_db\_properties システムプロシージャを使用して、すべてのデータベースプロパティの値を取り出します。

```
CALL sa_db_properties;
```

## データベースプロパティ

プロパティ名	説明
AccentSensitive	データベースでアクセント記号の違いが区別される (On)か、区別されない (Off) か。データベースでフランス語のアクセント記号の区別規則が適用されている場合、値は FRENCH になります。
Alias	データベース名。
AlternateMirrorServerName	データベースに関連付けられているミラーサーバの代替名 (代替名が指定されている場合)。
AlternateServerName	データベースに関連付けられているサーバの代替名 (代替名が指定されている場合)。
ApproximateCPUTime	データベースで使用された累積 CPU 時間の概算値 (秒単位)。この値と実際の CPU 時間との誤差は、通常は 5 ~ 10 % ですが、最大で 50 % に達することもあります。マルチプロセッサコンピュータでは、CPU (ハイパースレッドまたはコア) ごとに使用時間が積算されるため、すべての接続の累積時間を合計した値が実際の経過時間より大きくなる場合もあります。
ArbiterState	監視サーバのステータス。接続しているデータベースがミラーリングされていない場合、このプロパティは <i>NULL</i> になります。それ以外では、監視サーバがプライマリサーバに接続される場合は <i>connected</i> 、監視サーバがプライマリサーバに接続されない場合は <i>disconnected</i> になります。
AuditingTypes	現在有効な監査のタイプ。
Authenticated	データベースが認証されているかどうか。
BackupInProgress	データベースが現在バックアップ中かどうか。
BlankPadding	データベースで空白埋め込み機能が有効になっているかどうか。

プロパティ名	説明
BytesReceived	クライアント/サーバ通信中に受信したバイト数。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。
BytesReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したバイト数。この値は、圧縮が無効の場合は BytesReceived の値と同じです。
BytesSent	クライアント/サーバ通信中に送信されたバイト数。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。
BytesSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたバイト数。この値は、圧縮が無効の場合は BytesSent の値と同じです。
CacheHits	キャッシュ内の検索において一致したデータベースページの数。
CacheRead	キャッシュ内で検索されたデータベースページの数。
CacheReadIndInt	キャッシュから読み込まれたインデックス内部ノードページの数。
CacheReadIndLeaf	キャッシュから読み込まれたインデックスリーフページの数。
CacheReadTable	キャッシュから読み込まれたテーブルページの数。
CacheReadWorkTable	キャッシュワークテーブルの読み込み数。
Capabilities	データベースに対して有効な機能ビット。このプロパティは、主にテクニカルサポートを提供するために使用されます。
CarverHeapPages	クエリ最適化などの短期的な目的に使用されたヒープページの数。
CaseSensitive	データベースで大文字と小文字が区別される (On)か、区別されない (Off) か。
CatalogCollation	カタログに使用される照合の識別子。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。
CharSet	データベースの CHAR 文字セット。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。
CheckpointLogCommitToDisk	チェックポイントログのディスクへのコミット回数。
CheckpointLogPagesInUse	使用中のチェックポイントログページ数。
CheckpointLogPagesRelocated	再配置されたチェックポイントログページ数。
CheckpointLogPagesWritten	書き込みが完了したチェックポイントログページ数。
CheckpointLogSavePreimage	チェックポイントログに追加されているデータベースページのプレイメージ数。
CheckpointLogSize	チェックポイントログのサイズ (ページ数)。
CheckpointLogWrites	チェックポイントログへの書き込み数。

プロパティ名	説明
CheckpointUrgency	データベースのチェックポイント時間の設定に対するパーセンテージで表した最後のチェックポイント以降の経過時間。
Checksum	データベースのグローバルチェックサムが有効になっているかどうか (チェックサムはデータベースページごとに作成される)。重要なページには必ずチェックサムが含まれています。
Chkpt	実行されたチェックポイントの数。
ChkptFlush	チェックポイント実行中に書き出された一連のページの数。
ChkptPage	トランザクションログチェックポイントの数。
CleanablePagesAdded	データベースサーバの起動後、クリアされるようにマークが付けられたページの数。
CleanablePagesCleaned	データベースサーバの起動後、クリアされたデータベースページの数。
CleanableRowsAdded	データベースサーバの起動後、削除されるようにマークが付けられたローの数。
CleanableRowsCleaned	データベースサーバの起動後、削除されたシャドウテーブルのローの数。
ClientStmntCacheHits	クライアントに文がキャッシュされていることが理由で、このデータベースには不要な準備文の数。これは、クライアントでの文のキャッシュが無効になった場合に必要となる、追加の準備文の数です。
ClientStmntCacheMisses	このデータベースにおいて、クライアントのキャッシュに存在した文の中で、再び準備された文の数。これは、キャッシュされた文が再使用できるか検討されたものの、スキーマの変更、データベースのオプション設定の変更、DROP VARIABLE 文によって再使用できなかった回数です。
Collation	データベースが使用する照合。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
Commit	ディスクキャッシュのフラッシュをサーバが強制的に実行した回数。Windows では、バッファなし (ダイレクト) IO を使用している場合、ディスクキャッシュのフラッシュは不要です。
CommitFile	ディスクキャッシュのフラッシュをサーバが強制的に実行した回数。Windows では、バッファなし (ダイレクト) IO を使用している場合、ディスクキャッシュのフラッシュは不要です。
ConnCount	データベースとの接続の数。プロパティ値には、内部処理に使用されている接続は含まれませんが、イベントと外部環境サポートに使用されている接続は含まれます。ライセンスされている接続の使用中の正確な数を取得するには、次の文を実行します。 <code>SELECT COUNT ( * ) FROM sa_conn_info ( ) WHERE number &lt; 100000000;</code>
ConnectedTime	すべての接続がこれまでデータベースに接続してきた合計の時間数 (秒単位)。接続に対する要求が完了したとき、または接続が切断されたときにだけ値が更新されます。その結果、値は、データベースサーバ内で長いことアイドル状態であるか、実行中でビジーである接続に対しては遅れる可能性があります。値には、データベースイベントやバックグラウンドサーバ接続 (データベースクリーナなど) といった任意の接続によって増えた時間も含まれます。
ConnPoolCachedCount	アプリケーションによって切断されたが、接続プールにキャッシュされた接続の数。

プロパティ名	説明
ConnPoolHits	再利用されたプール内の接続の数。
ConnPoolMisses	キャッシュされたが、再利用できなかった接続プール内の接続の数。
ConnsDisabled	現在のデータベースとの接続が無効である (On) か、無効でない (Off) か。
CopyNodeParent	読み込み専用スケールアウト設定に含まれるコピーノードの現在の親サーバ名。
CurrentRedoPos	次のデータベース操作がロギングされるトランザクションログファイルの現在のオフセット。
CurrentTimelineID	内部でのみ使用。
CurrentTimelineSignature	内部でのみ使用。
CurrentTimezoneOffset	データベースが協定世界時 (UTC: Coordinated Universal Time) よりも前の時間 (分単位)。
CurrIO	サーバが発行し、まだ完了していない現在のファイル I/O の数。
CurrRead	サーバが発行し、まだ完了していない現在のファイル読み込み数。
CurrWrite	サーバが発行し、まだ完了していない現在のファイル書き込み数。
Cursor	現在データベースサーバが保持している、宣言されたカーソルの数。
CursorOpen	現在データベースサーバが保持している、開いたカーソルの数。
DatabaseCleaner	データベースクリーナーが有効かどうか。
DBFileFragments	データベースファイルのフラグメント数。このプロパティは Windows でサポートされています。
DiskRead	ディスクから読み込まれたページの数。
DiskReadHint	ディスクの読み込みヒント数。
DiskReadHintPages	ディスクの読み込みヒントページの数。
DiskReadIndInt	ディスクから読み込まれたインデックス内部ノードページの数。
DiskReadIndLeaf	ディスクから読み込まれたインデックスマリーフページの数。
DiskReadTable	ディスクから読み込まれたテーブルページの数。
DiskReadWorkTable	ディスクワークテーブルの読み込み数。
DiskRetryReadScatter	分散読み込みのためのディスク読み込みリトライ数。
DiskSandbox	データベースの読み込み/書き込みファイル操作が、メインのデータベースファイルがあるディレクトリに制限されているかどうか。
DiskSyncRead	同期的に発行されたディスクの読み込み数。

プロパティ名	説明
DiskSyncWrite	同期的に発行された書き込み数。
DiskWaitRead	データベースサーバによる非同期読み込みの待機回数。
DiskWaitWrite	データベースサーバによる非同期書き込みの待機回数。
DiskWrite	ディスクに書き込まれた修正ページの数。
DiskWriteHint	ディスクの書き込みヒント数。
DiskWriteHintPages	ディスクの収集書き込みヒント数。
DriveBus	データベースファイルが接続されているバスのタイプ (Unknown、SCSI、ATAPI、ATA、1394、SSA、Fibre、USB、RAID、iSCSI、SAS、SATA、SD、MMC、Virtual、FileBackedVirtual のいずれか)。バスのタイプを判別できない場合、このプロパティは NULL になります。プライマリ DB 領域に対するこのプロパティの値は SYSHISTORY ビューに記録されます。
DriveModel	データベースファイルが配置されているドライブのモデル番号。プライマリ DB 領域に対するこのプロパティの値は SYSHISTORY ビューに記録されます。ドライブのモデルを判別できない場合、このプロパティは NULL になります。
DriveType	データベースファイルが配置されているドライブのタイプ (CD、FIXED、RAMDISK、REMOTE、REMOVABLE、UNKNOWN のいずれか)。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。UNIX では、UNIX のバージョンとドライブのタイプにより、ドライブタイプを判別できない場合があります。このような場合、値は UNKNOWN になります。
Encryption	データベース暗号化またはテーブル暗号化のタイプ (None、Simple、AES、AES256、AES_FIPS、AES256_FIPS のいずれか)。
EncryptionScope	データベース内に暗号化が可能な部分がある場合、その部分を示す値。TABLE は、テーブルの暗号化が有効になっていることを示します。DATABASE は、データベース全体が暗号化されることを示します。NONE は、テーブルの暗号化が有効になっておらず、データベースが暗号化されないことを示します。
ExprCacheAbandons	ヒット率が低すぎるために式キャッシュが完全に中止された回数。
ExprCacheDropsToReadOnly	ヒット率が低いために式キャッシュが読み込み専用ステータスに変更された回数。
ExprCacheEvicts	式キャッシュからの出力数。
ExprCacheHits	式キャッシュ内のヒット数。
ExprCacheInserts	式キャッシュに挿入された値の数。
ExprCacheLookups	式キャッシュ内で実行されたルックアップの回数。
ExprCacheResumesOfReadWrite	ヒット率の上昇によって式キャッシュが読み込み/書き込みステータスに戻った回数。

プロパティ名	説明
ExprCacheStarts	式キャッシュが開始された回数。
ExtendDB	データベースファイルを拡張するために追加されたページの数。
ExtendTempWrite	テンポラリファイルを拡張するために追加されたページの数。
File	パスを含むデータベースルートファイル名。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。
FileSize	SYSTEM DB 領域のファイルサイズ (ページ単位)。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。
FreePages	SYSTEM DB 領域の空きページ数。FreePages プロパティは、バージョン 8.0.0 以降で作成されたデータベースでのみサポートされます。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。
FullCompare	インデックスのハッシュ値を超えて実行された比較の回数。
GetData	GETDATA 要求の数。
GlobalDBID	レプリケーション環境でのユニークなプライマリーキー値の生成に使用される global_database_id オプションの値。
HasCollationTailoring	データベースの作成時に照合の適合化が指定されていたかどうか。使用できる値は On と Off です。
HasEndianSwapFix	データベースでプラットフォームのエンディアンに関係なく、すべてのプラットフォームでビッグエンディアンとリトルエンディアンの両方の UTF-16 エンコードがサポートされるかどうか。使用できる値は On と Off です。
HashForcedPartitions	メモリ競合が原因でハッシュ演算子が分割された回数。
HashRowsFiltered	ビットベクトルフィルタによって拒否されたプローブロー数。
HashRowsPartitioned	ハッシュワークテーブルに書き込まれたロー数。
HashWorkTables	ハッシュベースの演算用に作成されたワークテーブルの数。
HasNCHARLegacyCollationFix	データベースが、バージョン 11 以降のソフトウェアで作成されたか、または従来の照合フィックスがインストールされ、従来の NCHAR 照合が使用されているバージョン 10 のデータベースサーバで作成された (On) か。従来の照合フィックスがインストールされていないバージョン 10 のデータベースサーバで作成されたデータベース、または従来の NCHAR 照合が使用されていないバージョン 10 のデータベースサーバで作成されたデータベースの場合、値は Off になります。
HasQDADTTFeature	キューの深さをチェックする機能が有効 (On) か、有効でない (Off) か。
HasTornWriteFix	部分書き込みからのリカバリを可能にする固定ファイルフォーマットがデータベースにあるかどうか。

プロパティ名	説明
HeapsCarver	クエリ最適化などの短期的な目的に使用されたヒープの数。
HeapsLocked	キャッシュ内で現在ロックされている再配置可能なヒープの数。
HeapsQuery	クエリ処理 (ハッシュ操作およびソート操作) に使用されるヒープの数。
HeapsRelocatable	再配置可能なヒープの数。
HttpClientMultipleHeaders	HTTP 結果セット内の同じ名前を持つ複数のヘッダがデータベースによりサポートされるかどうか。
HttpConnPoolCachedCount	すべてのプール内のキャッシュされたデータベース接続の絶対数。
HttpConnPoolHits	同じサービスで再利用された接続の割合。
HttpConnPoolMisses	プールから接続にアクセスできなかったときに割り当てられた新しい接続の割合。このプロパティでは、接続プールを使用するサービスへの HTTP 要求のみがカウントされます。アクセス時、プールのサイズが小さく、またそのプールの最も古い接続が横取りの基準に当てはまらなかったため、接続を使用できなかった可能性があります。
HttpConnPoolSteals	別のサービスに接続が属するサービスによって利用された接続の割合。再利用可能な直接接続がないサービスの HTTP 要求および最も使用されていない接続のプールサイズと経過時間の条件を満たす場合、サービスは別のサービスから接続を横取りします。プールの条件を満たさない場合は、サービスに接続が割り当てられ、HttpConnPoolMisses が増分されます。
IdentitySignature	このプロパティは内部でのみ使用されます。
IdentitySignatureUUID	このプロパティは内部でのみ使用されます。
IdleCheck	サーバのアイドルスレッドがアクティブになり、アイドル書き込み、アイドルチェックポイントなどを実行した回数。
IdleChkpt	サーバのアイドルスレッドが完了したチェックポイントの数。アイドルスレッドが最後のダーティページをキャッシュに書き出すたびに、アイドルチェックポイントが発生します。
IdleChkTime	アイドル I/O 中にチェックポイントに費やした 100 分の 1 秒単位の時間。
IdleWrite	サーバのアイドルスレッドによるディスク書き込みの数。
IndAdd	インデックスに追加されたエントリの数。
IndLookup	インデックス内で検索されたエントリの数。
IOParallelism	DB 領域がサポートする同時 I/O 操作の推定回数。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。
IOTorecover	データベースのリカバリに必要な I/O 操作の推定回数。

プロパティ名	説明
IQStore	このプロパティは内部でのみ使用されます。
JavaVM	データベースサーバがデータベース内で Java を実行するために使用する Java VM。
KeyDerivationIterations	データベースの作成時にデータベース暗号化キーがハッシュされた回数。
Language	データベース照合によってサポートされている言語のカンマで区切ったリスト。言語は 2 文字の ISO フォーマットです。言語が不明である場合、このプロパティは NULL を返します。
LastCheckpointTime	最新のチェックポイントの日時 (ミリ秒単位)。データベースで time_zone オプションが設定されている場合、データベースのタイムゾーンを使用して値が返されます。
LastCommitRedoPos	データベースによって最後の COMMIT 操作がトランザクションログに書き込まれた後の REDO ログの位置。
LastSyncedRedoPos	ディスクに対して書き込みが発行され、データが物理メディアに同期した最後の REDO の位置。この位置よりも前にあるデータは、停電が発生した場合でもディスク上に存在すると考えられます。
LastWrittenRedoPos	ディスクに対して書き込みが発行された最後の REDO の位置。この書き込みはオペレーティングシステム、ディスクコントローラ、またはディスクドライブによってキャッシュできるため、物理メディアに必ずしも同期する必要はありません。
LockCount	データベースで保持されているロックの数。
LockTablePages	ロック情報の保持に使用されているページの数。
LogFileFragments	ログファイルのフラグメント数。このプロパティは Windows でサポートされています。
LogFreeCommit	Redo Free Commit の数。Redo Free Commit が発生するのは、トランザクションログのコミットが要求されているが、ログはすでに書き込まれている (コミットが実行されている) 場合です。
LogMirrorName	パスを含むトランザクションログミラーのファイル名。
LogName	トランザクションログのパスを含むファイル名。
LogWrite	トランザクションログに書き込まれたページの数。
LTMGeneration	このプロパティは内部でのみ使用されます。
LTMTrunc	このプロパティは内部でのみ使用されます。
MaxConnections	データベースに対して許可する最大同時接続数。このプロパティは max_connections データベースオプションに相当します。
MaxIO	CurrIO が到達した最大値。
MaxRead	CurrRead が到達した最大値。
MaxWrite	CurrWrite が到達した最大値。

プロパティ名	説明
MirrorMode	データベースミラーリングが使用中でない (NULL) か、-xp オプションで指定されたミラーリングモードが同期または非同期化か。
MirrorRole	ミラーリング構成におけるデータベースサーバの役割。データベースがプライマリデータベースサーバまたはミラーサーバの場合、値は Primary になります。データベースサーバがミラーサーバの場合、値は Mirror になります。データベースサーバが現在のプライマリサーバ、現在のミラーサーバ、または別の読み込み専用ノードからログページを取得する読み込み専用ミラーサーバの場合、値は Copy になります。
MirrorServerState	ミラーサーバのステータス。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。
MirrorServerWaits	コピーサーバへログページを送信するときにデータベースサーバが 500 ミリ秒を超えて待機した回数。
MirrorState	ミラーリング構成におけるデータベースのステータスに関する情報。このプロパティには、DB_EXTENDED_PROPERTY 関数を使用してプロパティ値を問い合わせるときに指定できる拡張機能があります。
MultiByteCharSet	データベースで UTF-8 などのマルチバイト文字セットが使用されている (On) か、使用されていない (Off) か。
Name	データベース名 (Alias と同じ機能)。
NcharCharSet	データベースの NCHAR 文字セット。
NcharCollation	NCHAR データに使用される照合の名前。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
NextScheduleTime	指定したイベントの次の予定実行時間。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
OptionWatchAction	OptionWatchList プロパティに含まれているデータベースオプションを設定しようとしたときに実行されるアクション。
OptionWatchList	データベースサーバによってモニタされているデータベースオプションのリスト。
PacketsReceived	受信したクライアント/サーバ通信パケットの数。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したパケット数。この値は、圧縮が無効の場合は PacketsReceived の値と同じです。
PacketsSent	送信されたクライアント/サーバ通信パケットの数。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたパケット数。この値は、圧縮が無効の場合は PacketsSent の値と同じです。

プロパティ名	説明
PageRelocations	テンポラリファイルから読み込まれた再配置可能なヒープページの数。
PageSize	データベースのページサイズ (バイト)。
ParameterizationPrepareCount	自動的にパラメータ化された文に対する準備の数。
PartnerState	ミラーリング構成におけるデータベースサーバのステータスに関する情報。次のような値があります。このサーバからパートナーサーバへの接続と、パートナーサーバからこのサーバへの接続がある場合は <i>Connected</i> 、このサーバとパートナーサーバ間の接続がない場合は <i>Disconnected</i> 、パートナーサーバからこのサーバへの接続のみがある場合は <i>Incoming</i> 、このサーバからパートナーサーバへの接続のみがある場合は <i>Outgoing</i> になります。データベースサーバがミラーリング構成に関与しない場合、この値は <i>NULL</i> になります。
PlanStatisticsStored	内部でのみ使用。
Prepares	データベースに対して実行された文の準備作業の数。
PrepStmt	現在データベースサーバが保持している準備文の数。
PreviousTimelineID	内部でのみ使用。
ProcedurePages	プロシージャで使用された再配置可能なヒープページの数。
ProcedureProfiling	データベースのプロシージャプロファイリングが有効になっているかどうか。
PropertyHistoryList	このデータベースに対して追跡されている現在のプロパティセット。
QueryBypassed	プランキャッシュから再利用された要求の数。
QueryBypassedCosted	オプティマイザバイパスによってコストを使用して処理された要求の数。
QueryBypassedHeuristic	オプティマイザバイパスによってヒューリスティックを使用して処理された要求の数。
QueryBypassedOptimized	オプティマイザバイパスによって最初に処理されてから、オプティマイザによって完全に最適化された要求の数。
QueryCachedPlans	すべての接続においてキャッシュされている実行プランの数。
QueryCachePages	実行プランの保存に使用されるページの数。
QueryDescribedBypass	オプティマイザバイパスによって処理された DESCRIBE 要求の数。
QueryDescribedOptimizer	オプティマイザによって処理された DESCRIBE 要求の数。
QueryHeapPages	クエリ処理 (ハッシュ操作およびソート操作) に使用されるキャッシュページの数。
QueryJHToJNLOptUsed	ハッシュジョインがネストループジョインに変換された回数。

プロパティ名	説明
QueryLowMemoryStrategy	メモリ不足が原因でサーバが実行プランを実行中に変更した回数。使用できるメモリがオプティマイザの推定よりも少ない、または実行プランが必要とするメモリがオプティマイザの推定よりも多い場合に、プランが変更されることがあります。
QueryMemActiveCurr	クエリメモリをアクティブに使用する要求の数。
QueryMemGrantFailed	要求がクエリメモリ待ちになり、取得できなかった合計回数。
QueryMemGrantGranted	要求に現在付与されているページ数。
QueryMemGrantRequested	要求がクエリメモリを取得しようとした合計回数。
QueryMemGrantWaited	要求がクエリメモリ待ちになった合計回数。
QueryMemGrantWaiting	クエリメモリ待ちになっている現在の要求の数。
QueryOpened	実行対象の OPEN 要求の数。
QueryOptimized	完全に最適化された要求の数。
QueryReused	再利用されたクエリプランの数。
QueryRowsFetched	このデータベースについて、逐次スキャンまたはインデックススキャンによってベーステーブルから読み込まれたローの数。
QueryRowsMaterialized	クエリ処理中にワークテーブルに書き込まれるロー数。
ReadOnly	データベースが読み込み専用モードで実行されている (On)か、実行されていないか (Off)。
ReceivingTracingFrom	トレーシングデータが保存されていたデータベースの名前。トレーシングが追加されていない場合、値は空白文字列になります。
RecoveryUrgency	データベースのリカバリ時間設定に対する割合で表される、データベースのリカバリに要する推定時間。
Recursivelterations	再帰ユニオンの反復回数。
RecursivelterationsHash	再帰ハッシュジョインでハッシュ方式が使用された回数。
RecursivelterationsNested	再帰ハッシュジョインでネストループ方式が使用された回数。
RecursiveJNLMisses	再帰ハッシュジョインでのインデックスプローブキャッシュミス数。
RecursiveJNLProbes	再帰ハッシュジョインにおけるインデックスプローブの試行回数。
RelocatableHeapPages	再配置可能なヒープ (カーソル、文、プロシージャ、トリガ、ビューなど) で使用されるページの数。
RemoteTrunc	SQL Remote Message Agent 用に最後に確認されたログオフセット。

プロパティ名	説明
ReqCountActive	処理が完了した要求の数。RequestTiming サーバプロパティが Off に設定されている場合は NULL。
ReqCountBlockContention	接続がアトミックアクセスを待ったデータベースへの接続回数。-zt オプションが指定されていない場合は NULL。
ReqCountBlockIO	データベースへの接続が I/O 処理の完了を待った回数。-zt オプションが指定されていない場合は NULL。
ReqCountBlockLock	接続がロックを待ったデータベースへの接続回数。-zt オプションが指定されていない場合は NULL。
ReqCountUnscheduled	接続がスケジューリングを待ったデータベースへの接続回数。-zt オプションが指定されていない場合は NULL。
ReqTimeActive	要求の処理に要した時間。-zt オプションが指定されていない場合は NULL。
ReqTimeBlockContention	アトミックアクセスを取得するまでの待ち時間。RequestTiming サーバプロパティが Off に設定されている場合は NULL。
ReqTimeBlockIO	I/O 処理が完了するまでの待ち時間。-zt オプションが指定されていない場合は NULL。
ReqTimeBlockLock	ロックを待っていた時間。-zt オプションが指定されていない場合は NULL。
ReqTimeUnscheduled	未スケジュール時間、または -zt オプションが指定されていなかった場合は NULL。
RequestsReceived	クライアント/サーバ通信要求またはラウンドトリップの数。このプロパティでは、PacketsReceived とは異なり、マルチパケット要求を 1 つの要求として数え、活性パケットを計数の対象から除外します。
Rlbc	処理されたロールバック要求の数。
RollbackLogPages	ロールバックログのページ数。
SendingTracingTo	トレーシングデータの送信先を示す接続文字列。トレーシングが追加されていない場合、値はブランク文字列になります。
SnapshotCount	データベースに関連付けられているスナップショットの数。
SnapshotIsolationState	データベースでスナップショットアイソレーションが有効になっている (On) か、なっていない (Off) か。または、現在のトランザクションが完了すると、スナップショットアイソレーションが有効になる (in_transition_to_on) か、無効になる (in_transition_to_off) か。
SortMergePasses	ソート中に使用されたマージパスの数。
SortRowsMaterialized	ソートワークテーブルに書き込まれたロー数。
SortRunsWritten	ソート中に書き込まれたソート実行の数。

プロパティ名	説明
SortSortedRuns	実行の生成中に作成されたソート実行の数。
SortWorkTables	ソート用に作成されたワークテーブルの数。
StatementDescribes	DESCRIBE 要求によって処理された文の合計数。
StatementPostAnnotates	セマンティックなクエリ変換フェーズによって処理された文の数。
StatementPostAnnotatesSimple	セマンティックなクエリ変換フェーズによって処理されたが、一部のセマンティック変換が省略された文の数。
StatementPostAnnotatesSkipped	セマンティックなクエリ変換フェーズが完全に省略された文の数。
SynchronizationSchemaChangeActive	アクティブ接続が START SYNCHRONIZATION SCHEMA CHANGE 文を実行したが、STOP SYNCHRONIZATION SCHEMA CHANGE 文を実行していないか (On)、実行した (Off) か。
SyncTrunc	Mobile Link クライアントの dbmlsync 実行プログラム用に最後に確認されたログオフセット。
TempFileName	パスを含むデータベーステンポラリファイル名。
TempTablePages	テンポラリテーブルで使用されるテンポラリファイルのページ数。
TimelineBranchOffset	内部でのみ使用。
TimeWithoutClientConnection	最後の Command Sequence 接続または TDS クライアント接続が確立されてからの時間 (秒単位)、またはデータベースへの接続が確立されていない場合、データベースが開始されてからの時間。現在の接続がない場合、値は 0 になります。
TimeZone	データベースでタイムゾーンの計算に使用されるタイムゾーン。Timezone データベースオプションが設定されていない場合、値は NULL になります。
TriggerPages	トリガで使用される再配置可能なヒープページの数。
UserDefinedCounterRate01	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate02	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate03	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate04	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。

プロパティ名	説明
UserDefinedCounterRate05	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRaw01	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw02	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw03	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw04	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw05	ユーザが定義したパフォーマンスカウンタの現在の値。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UTCimestampCatalog	データベースがシステムテーブルに UTC タイムスタンプを格納している (On) か、していない (Off) か。
VersionStorePages	スナップショットアイソレーションが有効な場合にローバージョンストアで使用するテンポラリファイルのページ数。
ViewPages	ビューで使用される再配置可能なヒープページの数。
WriteChecksum	チェックサムが書き出される前にデータベースサーバによってページに追加される (On) か、されない (Off) か。
XPathCompiles	XPath クエリ (using the OPENXML 演算子を使用する) が、データベースサーバの起動時以降にデータベースサーバによってコンパイルされた回数。

## 関連情報

[データベースオプション \[622 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[接続プロパティの一覧 \[839 ページ\]](#)

## 1.5.7.4 ユーザ定義プロパティ

ユーザ定義プロパティは、データベースへの各接続で使用できます。

一連のユーザ定義プロパティがサポートされます。アプリケーションでは、それぞれのユーザ定義プロパティのセマンティックを定義し、返される値を設定する必要があります。プロパティ値は、sa\_user\_defined\_counter\_set システムプロシージャを使用して 32 ビットの UNSIGNED INTEGER 絶対値を設定したり、sa\_user\_defined\_counter\_add システムプロシージャを使って増分したりできます。sa\_user\_defined\_counter\_add システムプロシージャは、負の値を加算して、プロパティを減分するためにも使用できます。次のユーザ定義カウンタがあります。

- UserDefinedCounterRaw01
- UserDefinedCounterRaw02
- UserDefinedCounterRaw03
- UserDefinedCounterRaw04
- UserDefinedCounterRaw05
- UserDefinedCounterRate01
- UserDefinedCounterRate02
- UserDefinedCounterRate03
- UserDefinedCounterRate04
- UserDefinedCounterRate05

ユーザ定義プロパティの値は、それぞれの接続、データベース、サーバで個別に管理されます。プロパティの現在の値は、CONNECTION\_PROPERTY、DB\_PROPERTY、および PROPERTY 関数をそれぞれに使うことで検索できます。

これらのプロパティを使用する例として、キープールに残っている現在のキーの数を示す未加工プロパティを持つことがあります。このプロパティの値は、時間の経過に従って上下に変動します。同時に速度プロパティもキープールからキーがどの程度の速さで使用されたかを示すことができます。

### 関連情報

[統計のモニタに使用される SQL 関数 \[1432 ページ\]](#)

[Windows パフォーマンスモニタ \[1434 ページ\]](#)

## 1.5.7.5 データベースサーバプロパティの追跡

長期間にわたってデータベースサーバプロパティの数値の変更を追跡できるように、データベースサーバにデータベースサーバプロパティの数値を格納できます。

長期間にわたってデータベースサーバプロパティ値を追跡することは、データベースサーバの全体的な正常性を評価するのに役立ちます。たとえば、CPU 使用率が短時間増加することは問題ではない場合がありますが、CPU 使用率が長期間にわたって 100% の場合、ハードウェアが十分でないことを示している可能性があります。

データベースサーバのプロパティ値を定期的な間隔でポーリング、分析、格納するのではなく、数値を返すデータベースサーバプロパティを追跡するようにデータベースを設定します。これらの値は、より長い間隔でポーリングを実行し、データベースサーバの負荷を削減するように、一定期間インメモリに格納されます。

データベースサーバプロパティの追跡が有効な場合、プロパティ値は一定の間隔で追跡されます。sp\_property\_history system プロシージャを使用することで、過去のプロパティ値を問い合わせることができます。

このセクションの内容:

[すべての追跡可能なデータベースサーバプロパティ値の表示 \[889 ページ\]](#)

追跡が可能なデータベースサーバプロパティ値のリストを表示します。

[データベースサーバプロパティの追跡の設定 \[890 ページ\]](#)

データベースサーバを設定し、データベースサーバプロパティの数値を追跡します。

## 関連情報

[-sf データベースサーバオプション \[471 ページ\]](#)

[-phi データベースサーバオプション \[459 ページ\]](#)

[-phs データベースサーバオプション \[460 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[データベースプロパティの一覧 \[874 ページ\]](#)

[システム権限 \[1502 ページ\]](#)

### 1.5.7.5.1 すべての追跡可能なデータベースサーバプロパティ値の表示

追跡が可能なデータベースサーバプロパティ値のリストを表示します。

#### コンテキスト

数値のデータベースサーバプロパティのみを追跡できます。

sa\_eng\_properties システムプロシージャを実行するか、PROPERTY\_NUMBER 関数を呼び出すことで、データベースサーバプロパティの PropNum を確認できます。

#### 手順

次の文を実行します。

```
SELECT * FROM sa_eng_properties ()  
WHERE PROPERTY_IS_TRACKABLE( PropNum ) = 1;
```

## 結果

追跡が可能なデータベースサーバプロパティのリストが返されます。

## 関連情報

[-sf データベースサーバオプション \[471 ページ\]](#)

[-phl データベースサーバオプション \[459 ページ\]](#)

[-phs データベースサーバオプション \[460 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[データベースプロパティの一覧 \[874 ページ\]](#)

[システム権限 \[1502 ページ\]](#)

## 1.5.7.5.2 データベースサーバプロパティの追跡の設定

データベースサーバを設定し、データベースサーバプロパティの数値を追跡します。

## コンテキスト

数値のデータベースサーバプロパティのみを追跡できます。

## 手順

次のオプションのいずれか、または両方を選択してください。

オプション	アクション
データベースサーバで追跡するデータベースサーバプロパティを指定する	<ul style="list-style-type: none"><li>データベースサーバを起動する際、データベースサーバオプション <code>-phl</code> および <code>-phs</code> を使用して、指定したデータベースサーバのプロパティリストの履歴追跡をオンにし、プロパティ履歴の追跡で使用する最大メモリ量を指定します。たとえば、次のコマンドを実行します。 <code>dbsrv17 -n myserver -phl ProcessCPUSystem, ProcessCPUUser -phs 250K</code></li><li>データベースサーバがすでに実行されている場合、<code>sa_server_option</code> システムプロシージャを使用して、データベースサーバのプロパティ追跡を設定します。たとえば、次の文を実行します。 <code>CALL sa_server_option( 'PropertyHistoryList, ProcessCPUSystem, ProcessCPUUser, PropertyHistorySize, 250K' );</code></li></ul>

オプション	アクション
データベースで追跡するデータベースサーバプロパティを指定する	<p>sa_db_option システムプロシージャを使用して、データベースのデータベースサーバプロパティのプロパティ追跡を設定します。たとえば、次の文を実行します。</p> <pre>CALL sa_db_option( 'PropertyHistoryList,ProcessCPUSystem,ProcessCPUUser' );</pre>

## 結果

指定したデータベースサーバプロパティの値が、指定した時間または指定した最大メモリ量に達するまでの間、追跡されます。

## 関連情報

- [-sf データベースサーバオプション \[471 ページ\]](#)
- [-phi データベースサーバオプション \[459 ページ\]](#)
- [-phs データベースサーバオプション \[460 ページ\]](#)
- [データベースサーバプロパティの一覧 \[859 ページ\]](#)
- [データベースプロパティの一覧 \[874 ページ\]](#)
- [システム権限 \[1502 ページ\]](#)

## 1.5.8 データベースのサイズおよび数に関する物理的な制限

このトピックは、SQL Anywhere データベースにおけるオブジェクトのサイズと数の物理的な制限を示します。一般に、コンピュータのメモリ、CPU、ディスク容量から受ける制限の方が厳しいのが普通です。

項目	制限
データベースサイズ	13 ファイル/データベース。オペレーティングシステムとファイルシステムが許可する各ファイルの最大サイズ。
DB 領域サイズ	2 <sup>28</sup> x ページサイズ
テンポラリファイルのサイズ	2 <sup>28</sup> x ページサイズ
フィールドの大きさ	2 GB
ファイルサイズ (FAT 12)	16 MB

項目	制限
ファイルサイズ (FAT 16)	2 GB
ファイルサイズ (FAT 32)	4 GB
ファイルサイズ (NTFS、HP-UX 11.0 以降、Solaris 2.6 以降、Linux 2.4 以降、AIX、Mac OS X)	<ul style="list-style-type: none"> <li>● 512 GB (2 KB ページに対して)</li> <li>● 1 TB (4 KB ページに対して)</li> <li>● 2 TB (8 KB ページに対して)</li> </ul>
ファイルサイズ (その他のプラットフォームとファイルシステム)	2 GB
最大キャッシュサイズ (Windows Server 2003 Web Edition、Windows Server 2003 Standard Edition、Windows Server 2008、Windows Server 2008 R2、Windows 7)	1.8 GB
最大キャッシュサイズ (Windows Server 2003 Enterprise Edition、Windows Server 2003 Datacenter Edition)	2.7 GB
最大キャッシュサイズ (UNIX: Solaris、x86 Linux、IBM AIX、HP)	2 GB (32 ビットサーバに対して)
最大キャッシュサイズ (Win 64)	64 ビットサーバの物理メモリによる
最大キャッシュサイズ (Itanium HP-UX)	64 ビットサーバの物理メモリによる
最大インデックスエントリサイズ	制限なし
データベース数/サーバ	255
カラム数/テーブル	45000 カラム数を過度に多くすることは可能ですがパフォーマンスに影響します。
NULL 入力可のカラムの数/テーブル	45000 と ((ページサイズ - オーバヘッド) * 8) のうち小さい方
プロシージャの結果セット内のカラム数	45000
SELECT リスト内のカラム数	100000
GROUP BY リスト内のカラム数	100000
グループセットのある GROUP BY リスト内のカラム数	64
CUBE 内のカラム数	15
異なるグループセットの数	32768
カラムの DEFAULT の長さ	32768
カラムの COMPUTE の長さ	32768
プロシージャパラメータの DEFAULT の長さ	32768
ユーザ定義ドメインの DEFAULT の長さ	32768
検査制約の長さ	2 GB
インデックス数/テーブル	$2^{32}$
ロー数/データベース	$4096 \times 2^{28} \times 13$
ロー数/テーブル	$4096 \times 2^{28}$

項目	制限
ベーステーブルとグローバルテンポラリテーブル数/データベース	$2^{32} - 2^{20} - 1 = 4293918719$
テンポラリテーブル数/接続	$2^{20} = 1048576$
参照されるテーブル数/トランザクション	制限なし
ストアプロシージャ数/データベース	$2^{32} - 1 = 4294967295$
同時実行文の数/データベースサーバ	$20 \times \text{number-of-database-connections} + 65534$
イベント数/データベース	$2^{31} - 1 = 2147483647$
トリガ数/データベース	$2^{32} - 1 = 4294967295$
ローサイズ	ファイルサイズにより制限
テーブルの大きさ	最大ファイルサイズ。テーブルのユーザ定義インデックスは、テーブルとは別に保存できます。
文字データ型	$2^{31} - 1 = 2147483647$
バイナリデータ型	$2^{31} - 1 = 2147483647$
ローサイズ	45000 フィールド
配列サイズ	640 万要素
識別子 (ユーザ ID、テーブル名、カラム名を含む)	128 バイト
パスワード	255 バイト
データベースサーバ名	250 バイト (TCP/IP と共有メモリ)
データベース名	250 バイト
データベースあたりのセキュリティ機能キーの数	1000

このセクションの内容:

[SQL Anywhere ハードウェア要件 \[893 ページ\]](#)

SQL Anywhere にはいくつかのハードウェア要件があります。

## 関連情報

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

### 1.5.8.1 SQL Anywhere ハードウェア要件

SQL Anywhere にはいくつかのハードウェア要件があります。

ハードウェア要件の詳細については、次の URL を参照してください。 <http://scn.sap.com/docs/DOC-35653>

## 関連情報

[SQL Anywhere Minimum Hardware Requirements \(SQL Anywhere の最小ハードウェア要件\)](#)

## 1.6 データベースのメンテナンス

データベースを管理するためのツールや方法には、いくつかの種類があります。

このセクションの内容:

### [データベースのバックアップとリカバリ \[894 ページ\]](#)

バックアップを使用して、データベースが使用できなくなった時点までにコミットされたすべての変更をリストアします。

### [データベースの検証 \[942 ページ\]](#)

データベースファイルの破損は、データベースサーバがデータベース内の破損部分にアクセスするまで判明しないことがあります。データベースが有効であることを定期的にチェックします。

### [スケジュールとイベントの使用によるタスクの自動化 \[950 ページ\]](#)

スケジュールとイベントによるデータベース管理タスクの自動化

### [イベントのトレース \[964 ページ\]](#)

イベントのトレースでは、システム定義およびユーザ定義のトレースイベントに関する情報がイベントトレースデータ (ETD) ファイルに記録されます。

### [データベースのトラブルシューティング \[971 ページ\]](#)

データベースのトラブルシューティングに使用できる、多くのツールおよび方法があります。

### 1.6.1 データベースのバックアップとリカバリ

バックアップを使用して、データベースが使用できなくなった時点までにコミットされたすべての変更をリストアします。

バックアップとはデータベース内の情報の全部または一部のコピーであり、物理的に別の場所に格納されます。メインのデータベースファイルが使用不可の場合、バックアップからコピーをリストアすることができ、現在のトランザクションログを使用して、リストアしたデータベースには存在しない可能性のある追加データをリカバリすることができます。

いくつかのバックアップのタイプと、いくつかのデータのリストア方法およびリカバリ方法があります。

バックアップおよびリカバリにおける 2 つの非常に重要なベストプラクティスは、1) バックアップからリストア可能かの検証を含む適切なバックアップ方式に従うことと、2) バックアップファイル自体からではなく、バックアップファイルのコピーからリストアすることです。

このセクションの内容:

### [データベースのバックアップ \[895 ページ\]](#)

実行中のデータベースをバックアップすると、他のユーザによってデータベースが変更されている途中であっても、データが一貫した状態にあるデータベースのスナップショットを得ることができます。

### [データベースのリカバリ \[915 ページ\]](#)

リカバリは、インクリメンタルトランザクションログファイルを使用してデータベースファイル、トランザクションログ、DB領域をリストアし、データベースファイルをできるだけ最新の状態にすることです。

[タイムライン \[942 ページ\]](#)

タイムラインはデータベースに対して行われる変更の順番を確立します。

## 1.6.1.1 データベースのバックアップ

実行中のデータベースをバックアップすると、他のユーザによってデータベースが変更されている途中であっても、データが一貫した状態にあるデータベースのスナップショットを得ることができます。

ログの最後のページが読み込まれる前に、トランザクションログに記録されたすべての完了したトランザクションがバックアップされます。またバックアップが開始されたあと、選択したバックアップのオプションに依存して、追加オペレーションが開始されます。

バックアップの開始時に、データベースサーバはチェックポイントを発行します。バックアップの最後で、データベースサーバはデータベースのバックアップコピーに、リカバリが必要であることを示すフラグを指定します。この処理によって、バックアップの開始以降に実行されたオペレーションが、データベースのバックアップコピーの開始時に適用されます。また、チェックポイントの時点で完了していなかったオペレーションは、コミットされていなければ取り消されます。

データベースサーバでは、バックアップ処理中に次のオペレーションを実行できないようにします。

- 別のバックアップ (ライブバックアップを除く)。
- チェックポイント (バックアップ命令によって発生するチェックポイントは除く)。
- チェックポイントを発生させる文。

## サポートされるバックアップツールおよび方法

バックアップを実行するのに使用できるバックアップツールおよび方法がいくつかあります。

- BACKUP DATABASE 文
- バックアップユーティリティ (dbbackup)
- バックアップイメージ作成ウィザード
- データベースバックアップウィザード

データベースツールインタフェース (DBTools) の DBBackup メソッドを使用することもできますが、このメソッドはここでは説明しません。加えて、Microsoft ボリュームシャドウコピーサービス (VSS) のサポートも提供されています。Microsoft VSS を使用して、バックアップとしてのファイルのスナップショットを作成することができます。

データベースミラーリング (高可用性)、読み込み専用スケールアウト、およびトランザクションログミラーリングは、メディアの障害時に継続アクセスを保証する方法で、バックアップ方法としては考慮されません。

このセクションの内容:

[バックアップの種類 \[896 ページ\]](#)

バックアップを実行するタイミング、地点、方法を決定するには、さまざまな選択肢があります。決定内容によって、実行するバックアップのタイプ (サーバ側、クライアント側、フル、インクリメンタルなど) が決まります。

[バックアップタスク \[901 ページ\]](#)

バックアップの作成および管理に関連付けられているいくつかのタスクがあります。

#### [バックアップとリカバリ方式の設計 \[908 ページ\]](#)

データを保護するために、バックアップおよびリカバリ方式を実装することができます。

#### [バックアップ中のトランザクションログで発生すること \[913 ページ\]](#)

バックアップを作成中、指定しない限り、データベースサーバはトランザクションログのコピーを作成して、トランザクションログが大きくなり続けるままにします。

#### [同期やレプリケーションに関連するデータベースの考慮事項 \[914 ページ\]](#)

同期やレプリケーションの環境では、トランザクションログの名前を変更してトランザクションログを再開するようにバックアップオプションを使用します。

## 関連情報

[SQL Anywhere ボリュームシャドウコピーサービス \(VSS\) \[900 ページ\]](#)

[トランザクションログミラー \[288 ページ\]](#)

[高可用性と読み込み専用のスケールアウトシステム \[1659 ページ\]](#)

[データベースのリカバリ \[915 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

### 1.6.1.1.1 バックアップの種類

バックアップを実行するタイミング、地点、方法を決定するには、さまざまな選択肢があります。決定内容によって、実行するバックアップのタイプ (サーバ側、クライアント側、フル、インクリメンタルなど) が決まります。

バックアップのタイプは、相互排他的ではありません。たとえば、フル、オンライン、サーバ側、イメージバックアップの実行を決定することができます。つまり、データベースサーバコンピュータからバックアップを開始して、実行中データベースとそのトランザクションログのフルバックアップを実行し、データベースとトランザクションログのコピーを個別のファイルとしてデータベースサーバコンピュータに格納します。

#### ➔ ヒント

一般的で効率的なバックアップ戦略は、サーバ側のフルイメージバックアップを別のデバイスに定期的に行いながら、各フルバックアップの間に、フルバックアップよりも高い頻度でインクリメンタルバックアップを実行することです。実際の障害が発生したときにリカバリが問題なく実行されることを確認するために、バックアップからのリカバリを必ずテストします。

次の表に、実行可能なバックアップのタイプ、適用可能な時期、バックアップに使用できるツールを示します。多くのバックアップタイプに、関連する代替手段があります。これらのバックアップタイプは、その代替手段に沿って定義されます。

バックアップ用語	説明	次を使用して実行
サーバ側およびクライアント側のバックアップ	<p>サーバ側バックアップはデータベースサーバによって実行されます。この種類のバックアップは、データを他のコンピュータに転送する必要がないので、一般的にクライアント側より高速です。</p> <p>クライアント側バックアップはバックアップユーティリティ (dbbackup) によって実行されるバックアップです。ただし、これには例外があります。-r オプションで dbbackup を実行すると、データベースサーバが BACKUP DATABASE 文でバックアップを実行するので、サーバ側バックアップになります。</p>	<p>提供されているいずれかのバックアップツールを使用して、サーバ側バックアップを実行します。バックアップユーティリティ (dbbackup) を使用する際には、-s オプションを指定する必要があります。</p> <p>バックアップユーティリティ (dbbackup) を使用して、クライアント側バックアップを実行します。</p>
フルバックアップおよびインクリメンタルバックアップ	<p>フルバックアップはデータベースファイルのコピーとトランザクションログからなります。フルバックアップはすべてのデータを含むので有益ですが、特にデータベースのサイズが大きいと時間がかかります。一般的なバックアップの方法は、定期的に (たとえば週次で) フルバックアップを実行しながら、より高い頻度で (たとえば日次で) インクリメンタルバックアップを実行します。</p> <p>インクリメンタルバックアップは、トランザクションログのみのコピーからなります。インクリメンタルバックアップを実行する前に、フルバックアップを実行します。インクリメンタルバックアップ (フルバックアップ間の) はもっとも一般的なバックアップです。このバックアップはトランザクションログをコピーするだけなので、フルバックアップに比べてかなり短時間になります。</p>	<p>インクリメンタルは例外として、提供されているいずれかのバックアップツールを使用して、フルバックアップまたはインクリメンタルバックアップを実行します。</p> <p>ただし、SQL Central のデータベースバックアップウィザードを使用して、インクリメンタルアーカイブバックアップを実行することはできません。</p>
イメージバックアップおよびアーカイブバックアップ	<p>イメージバックアップでは、データベースファイルとトランザクションログをコピーして、それぞれ別のファイルとして保存します。アーカイブバックアップからよりも、イメージバックアップからのほうがリストアが容易にできます。これは、最も一般的に選択されるバックアップ形式です。</p>	<p>次を使用してイメージバックアップを実行します。</p> <ul style="list-style-type: none"> <li>● BACKUP DATABASE 文</li> <li>● バックアップユーティリティ (dbbackup)</li> <li>● SQL Central (イメージバックアップ) のバックアップイメージ作成ウィザード</li> <li>● The コックピット</li> </ul>

バックアップ用語	説明	次を使用して実行
オンラインバックアップとオフラインバックアップ	<p>オンラインバックアップは実行中のデータベースに対して行われます。実行中のデータベースをバックアップすると、他のユーザによってデータベースが変更されている途中であっても、データが一貫した状態にあるデータベースのスナップショットを得ることができます。通常、バックアップはオンラインバックアップで実行されます。</p> <p>オフラインバックアップはデータベースが実行していないときに作成されるデータベースファイルのコピーです。オフラインバックアップコピーは、データベースが通常どおり停止しており、データベースサーバが適切にシャットダウンしている状態で実行する必要があります。</p>	<p>ライブバックアップは、提供されているいずれかのバックアップツールを使用して実行します。</p> <p>オフラインバックアップは、サードパーティのバックアップソフトウェアなどのツールを使用するか、オペレーションシステムのファイル管理機能を使用して実行します。</p>
ライブバックアップまたは継続バックアップ	<p>ライブバックアップ (継続バックアップとも呼ばれる) は、データベースが実行中に継続的に作成されるトランザクションログのバックアップです。</p>	<p>ライブバックアップを行うには、バックアップユーティリティ (dbbackup) を使用して、-l オプションを指定します。</p>
並列バックアップ	<p>並列バックアップはサーバ側のバックアップで、物理デバイスレベルの並列処理を使用して、バックアップ操作の完了に必要な総時間を節約します。</p>	<p>BACKUP DATABASE 文、またはバックアップユーティリティ (dbbackup) を使用して、サーバ側のバックアップの一部として並列バックアップを実行します。</p>

このセクションの内容:

[ライブバックアップ \(継続バックアップ\) \[899 ページ\]](#)

ライブバックアップは、セカンダリコンピュータで実行されているデータベースの継続的なバックアップです。

[SQL Anywhere ボリュームシャドウコピーサービス \(VSS\) \[900 ページ\]](#)

SQL Anywhere は、Microsoft ボリュームシャドウコピーサービス (VSS) と互換性があります。

[並列データベースバックアップ \[900 ページ\]](#)

並列バックアップは、物理デバイスレベルの並列処理を使用して、バックアップ操作の完了に必要な総時間を節約します。

## 関連情報

[バックアップユーティリティ \(dbbackup\) を使用したデータベースのバックアップ \[902 ページ\]](#)

[BACKUP DATABASE 文を使用したデータベースのバックアップ \[904 ページ\]](#)

[SQL Central を使用したデータベースのバックアップ \[906 ページ\]](#)

## 1.6.1.1.1.1 ライブバックアップ (継続バックアップ)

ライブバックアップは、セカンダリコンピュータで実行されているデータベースの継続的なバックアップです。

メディア障害が発生した場合、セカンダリコンピュータでデータベースを再起動できます。データベースサーバが実行しているコンピュータの負荷とパフォーマンスによっては、ライブバックアップでは、メディア障害の前に発生したすべてのコミット済みトランザクションがバックアップされないことがあります。

ライブバックアップを実行するには、バックアップユーティリティ (dbbackup) を使用し、-l オプションを指定します。ライブバックアップは必ず別のコンピュータから実行します。

トランザクションログのバックアップは、必ず実際のトランザクションログと同じ長さか、それより短くなります。ライブバックアップの実行中に、別のバックアップがトランザクションログを再起動すると (dbbackup -r または dbbackup -x)、ライブバックアップは自動的にライブバックアップログをトランケートして、新しいトランザクションログの最初からライブバックアップを再起動します。

### ライブバックアップとトランザクションログミラーの比較

ライブバックアップとトランザクションログミラーは両方とも、トランザクションログのセカンダリコピーを作成します。しかし、ライブバックアップの使用とトランザクションログミラーの使用には、次のようないくつかの違いがあります。

トランザクションログミラーは、データベースを初期化したときに有効になる

バックアップユーティリティ (dbbackup) を使用してライブデータベースで開始されるライブバックアップとは異なり、トランザクションログミラーリングは、-m サーバオプションを使用し、データベースの初期化時に有効になります。

通常、ライブバックアップは別のコンピュータで実行される

別のコンピュータでバックアップユーティリティ (dbbackup) を実行すると、バックアップトランザクションログファイルへの書き込みはデータベースサーバによっては行われません。また、データ転送は SQL Anywhere クライアント/サーバ通信システムによって実行されます。したがって、パフォーマンスへの影響を低減でき、信頼性も向上します。

トランザクションログミラーの場合、別のコンピュータで実行することはお奨めしません。実行するとパフォーマンスに問題が発生したりデータが破壊されたりする可能性があり、コンピュータ間の接続に障害が発生すると、データベースサーバが停止します。

ライブバックアップは、コンピュータが使用できなくなる状況から保護する

トランザクションログミラーを別のデバイスに保存しても、コンピュータ全体が使用できなくなってしまうと、リカバリには時間がかかります。2 台のコンピュータで一連のディスクへのアクセスを共有するように構成するのも 1 つの方法です。

ライブバックアップはデータベースサーバより処理が遅れることがある

トランザクションログミラーには、コミットされたトランザクションを完全にリカバリするのに必要な情報がすべて含まれます。データベースサーバが処理しているロード量によっては、ライブバックアップはトランザクションログミラーよりも処理が遅く、コミットされたトランザクションがすべてバックアップされないことがあります。

### 関連情報

[トランザクションログミラー \[288 ページ\]](#)

[ライブバックアップからのデータベースの再開 \[933 ページ\]](#)

## 1.6.1.1.2 SQL Anywhere ボリュームシャドウコピーサービス (VSS)

SQL Anywhere は、Microsoft ボリュームシャドウコピーサービス (VSS) と互換性があります。

デフォルトでは、SQL Anywhere VSS ライタ (`dbvss17.exe`) が実行されている場合、すべての SQL Anywhere データベースでバックアップに VSS サービスを使用できます。VSS は、SQL Anywhere VSS ライターなしでデータベースのバックアップに使用できます。ただし、これらのデータベースをリストアするには、SQL Anywhere の完全なリカバリ手順を使用する必要があります。

データベースサーバが VSS サービスに参加しないようにするには、データベースサーバの起動時に `-vss-` を指定します。また、Windows 用サービスユーティリティ (`dbsvc`) を使用して VSS サービスの開始時刻を制御することもできます。

SQL Anywhere では VSS は次のように機能します。

1. バックアップアプリケーションが、スナップショットを作成するコマンドを VSS に送信します。
2. VSS が SQL Anywhere VSS ライタ (`dbvss17.exe`) に `identify` コマンドを発行します。
3. VSS が、すべてのトランザクションをサスペンドし、すべてのデータベースサーバにあるすべてのデータベースの変更されたすべてのページをディスクに書き込む `prepare to snapshot` コマンドを発行します。データベースのトランザクションが 10 秒以内にサスペンドしなかった場合は、コミットされていないトランザクションがスナップショットに含まれ、フルリカバリが必要になる場合があります。
4. VSS が、すべてのデータベースサーバにあるすべてのデータベースのすべてのアクティビティをチェックポイントしてからサスペンドする `freeze` コマンドを発行します。各 SQL Anywhere データベースサーバは、すべてのデータベースですべてのアクティビティがサスペンドされるまで最大 60 秒間待ちます。一般に、この処理には数秒間かかります。
5. VSS が、SQL Anywhere VSS ライターに `thaw` コマンドを発行し、すべてのデータベースサーバにあるすべてのデータベースに対するすべてのトランザクションを再開します。

VSS サービスを使用する場合でも、フルバックアップおよび定期インクリメンタルバックアップを伴うバックアップ戦略の採用を検討してください。まれに、VSS で許可されている最大時間内に、SQL Anywhere がトランザクションをサスペンドしたり、チェックポイントを完了したりできないことがあります。この状況が発生した場合に、SQL Anywhere バックアップをすでに実行している場合、トランザクションログファイルおよびフルリカバリプロセスを使用してデータをリカバリします。

### 関連情報

[Windows サービスの作成 \(SQL Central の場合\) \[341 ページ\]](#)

[Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

## 1.6.1.1.3 並列データベースバックアップ

並列バックアップは、物理デバイスレベルの並列処理を使用して、バックアップ操作の完了に必要な総時間を節約します。

`-s` オプションを指定したバックアップユーティリティ (`dbbackup`) を使用するか、`AUTO TUNE WRITERS` 句を含む `BACKUP DATABASE` 文を使用して、サーバ側のイメージバックアップを実行すると、データベースの並列バックアップが実行されます。

データベースサーバは、データベースファイルが保存されている各ドライブについてリーダースレッドを作成します。ライタースレッドは、バックアップディレクトリが存在するバックアップ先ドライブ用に作成されます。リーダーとライターを別々に使用することで、I/O 処理が順次的にではなく並列に実行されます。

並列バックアップのパフォーマンスは、システムで最も低速なコンポーネントによって制限されます。これは通常は物理ディスクですが、I/O コントローラやシステムバスなど、他のコンポーネントの場合もあります。これらの各コンポーネントのデータ転送レートには上限があります。

BACKUP DATABASE 文とバックアップユーティリティ (dbbackup) には、並列バックアップの動作を設定できるよう次のオプションが用意されています。

- チェックポイントログをコピーするタイミングと方法
- データベースサーバから dbbackup へのデータ転送時に使用できる最大ページ数 (dbbackup の使用時のみ使用可)
- ライターの追加 (BACKUP DATABASE 文の場合のみ)

バックアップは、必ず別の物理ドライブに作成してください。これにより、I/O 並列処理によるパフォーマンス改善が実現され、ハードウェア障害が発生したときのデータの安全性が向上します。

## 関連情報

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

### 1.6.1.1.2 バックアップタスク

バックアップの作成および管理に関連付けられているいくつかのタスクがあります。

このセクションの内容:

[バックアップユーティリティ \(dbbackup\) を使用したデータベースのバックアップ \[902 ページ\]](#)  
バックアップユーティリティ (dbbackup) を使用してデータベースをバックアップします。

[BACKUP DATABASE 文を使用したデータベースのバックアップ \[904 ページ\]](#)  
BACKUP DATABASE 文を使用して、データベースをバックアップします。

[SQL Central を使用したデータベースのバックアップ \[906 ページ\]](#)  
SQL Central のウィザードを死傷してさまざまな種類のバックアップを実行します。

## 1.6.1.1.2.1 バックアップユーティリティ (dbbackup) を使用したデータベースのバックアップ

バックアップユーティリティ (dbbackup) を使用してデータベースをバックアップします。

### 前提条件

BACKUP DATABASE および VALIDATE ANY OBJECT システム権限が必要です。

ディスクサンドボックス機能が有効で、サンドボックスの外部にバックアップを行っている場合、データベースサーバにセキュリティ機能キーが必要です。

### コンテキスト

バックアップユーティリティ (dbbackup) では、バックアップするデータベースへ接続する必要があります。このため、オンラインバックアップのみで使用します。ただし、テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

データベースのロケーションとは異なるコンピュータ上のロケーションを選択します。これにより、メディア障害が発生する場合に備えることができます。

バックアップユーティリティ (dbbackup) を使用して、アーカイブバックアップを実行することはできません。

### 手順

1. データベースを検証します。たとえば、次と同じような dbvalid コマンドを実行します。

```
dbvalid -c "UID=DBA;PWD=sql;SERVER=demo17;DBN=demo"
```

2. 次の例のいずれかと同じような dbbackup コマンドを実行します。Syntax カラムの例は、複数の行に分割されています。ただし、実行するコマンドは 1 行である必要があります。

オプション	アクション
<ul style="list-style-type: none"><li>フル</li><li>イメージ</li><li>クライアント側</li></ul>	<pre>dbbackup -c "HOST=myHost;SERVER=demo17;DBN=demo; UID=DBA;PWD=sql" "c:¥temp¥SQLAnybackup"</pre>
<ul style="list-style-type: none"><li>フル</li><li>イメージ</li><li>サーバ側</li><li>トランザクションの完了を待機</li></ul>	<pre>dbbackup -c "HOST=myHost;SERVER=demo17;DBN=demo; UID=DBA;PWD=sql" - s -wa "c:¥temp¥SQLAnybackup"</pre>

オプション	アクション
<ul style="list-style-type: none"> <li>○ フル</li> <li>○ イメージ</li> <li>○ クライアント側</li> </ul>	<pre>dbbackup -c "HOST=myHost;SERVER=demo17;DBN=demo;UID=DBA;PWD=sql" "c:¥temp¥SQLAnybackup"</pre>
<ul style="list-style-type: none"> <li>○ インクリメンタル</li> <li>○ イメージ</li> <li>○ クライアント側</li> </ul>	<pre>dbbackup -c "HOST=myHost;SERVER=demo17;DBN=demo; UID=DBA;PWD=sql"- t -r -n "c:¥temp¥SQLAnybackup"</pre>
<ul style="list-style-type: none"> <li>○ ライブ</li> </ul>	<pre>フルバックアップ dbbackup -c "HOST=myHost;SERVER=demo17;DBN=demo" -wa "c:¥temp¥SQLAnybackup";  ライブバックアップ dbbackup -c "HOST=myHost;SERVER=demo17;DBN=demo" "c: ¥temp¥SQLAnybackup" -l "demo.log"</pre>

## 結果

バックアップが実行され、設定ファイル backup.syb にエントリが作成されます。

### 例

この例では、サンプルデータベース demo のライブバックアップを作成します。最初のコマンドでフルバックアップが作成されます。リカバリではバックアップトランザクションログファイルを適用するデータベースファイルが必要なため、ライブバックアップの作成前にフルバックアップを作成する必要があります。2 つめのコマンドでライブバックアップが開始します。

```
dbbackup -c "SERVER=demo17;DBN=demo;UID=DBA;PWD=sql" "c:¥temp¥SQLAnybackup"
dbbackup -c "SERVER=demo17;DBN=demo;UID=DBA;PWD=sql" "c:¥temp¥SQLAnybackup" -l
"demo.log"
```

ライブバックアップが開始されると、トランザクションログのライブバックアップが次のページの待機中であることを示すメッセージ表示されます。オペレーティングシステムによっては、バックアップトランザクションログファイル (demo.log) 上でアクティビティが発生していないように見えることがあります。これはライブバックアップ中ファイルが開いたままであり、オペレーティングシステムによっては、ファイルが閉じられるまでファイルのメタデータ (サイズ、最終変更のタイムスタンプなど) が変更されないからです。このケースでは、たとえばアクティブデータベースの障害後や、データベースへの接続が切断された場合など、ライブバックアップが終了するとファイルが閉じられます。

-l オプションでは、トランザクションログファイル名がフルバックアップ中に作成されたファイルと同じだった場合、上書きするかどうかを聞かれます。異なる名前を指定すれば、新しいバックアップトランザクションログがその名前で作成され、ライブバックアップ用に使用されます。データベースが障害中になると、ライブバックアップは停止します。その後、アクティブトランザクションログの名前は変更されていないと仮定して、バックアップトランザクションログをフルバックアップ中に作成したバックアップデータベースに適用することができます。

### 注記

このライブバックアップの例のリカバリは、アクティブトランザクションログの名前が変更されていない場合のみ正常に実行されます。

## 次のステップ

作成したバックアップからリカバリができることを確認します。

## 関連情報

[ライブバックアップ \(継続バックアップ\) \[899 ページ\]](#)

[バックアップの種類 \[896 ページ\]](#)

[データベースのリカバリ \[915 ページ\]](#)

[ディスクサンドボックス \[1604 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

[検証ユーティリティ \(dbvalid\) \[1200 ページ\]](#)

## 1.6.1.1.2 BACKUP DATABASE 文を使用したデータベースのバックアップ

BACKUP DATABASE 文を使用して、データベースをバックアップします。

### 前提条件

BACKUP DATABASE および VALIDATE ANY OBJECT システム権限が必要です。

ディスクサンドボックス機能が有効で、サンドボックスの外部にバックアップを行っている場合、データベースサーバにセキュリティ機能キーが必要です。

テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

### コンテキスト

BACKUP DATABASE 文はデータベースサーバのコンピュータ上で実行されます。実行中のデータベースへ接続する必要があるため、サーバ側からのオンラインバックアップのみを実行します。

ディスクにバックアップするときは、データベースのロケーションとは異なるコンピュータ上のロケーションを選択します。これにより、データベースが実行中のマシンでメディア障害が発生する場合に備えることができます。

## 手順

1. データベースを検証します。たとえば、次と同じような VALIDATE 文を実行します。

```
VALIDATE DATABASE;
```

2. 次の例のいずれかと同じような BACKUP DATABASE 文を実行します。

オプション	アクション
<ul style="list-style-type: none"><li>フル</li><li>イメージ</li></ul>	BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥SQLAnybackup';
<ul style="list-style-type: none"><li>インクリメンタル</li><li>イメージ</li></ul>	BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥SQLAnybackup' TRANSACTION LOG ONLY TRANSACTION LOG TRUNCATE;
<ul style="list-style-type: none"><li>フル</li><li>アーカイブ</li><li>ディスクへ</li></ul>	BACKUP DATABASE TO 'c:¥¥temp¥¥SQLAnybackup¥¥archive';
<ul style="list-style-type: none"><li>フル</li><li>イメージ</li><li>名前が変更されたトランザクションログ</li></ul>	BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥SQLAnybackup' TRANSACTION LOG RENAME;
<ul style="list-style-type: none"><li>インクリメンタル</li><li>イメージ</li><li>名前が変更されたトランザクションログ</li></ul>	BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥SQLAnybackup' TRANSACTION LOG ONLY TRANSACTION LOG RENAME MATCH;

## 結果

バックアップが実行され、設定ファイル backup.syb にエントリが作成されます。

### 例

次の文によって、バックアップを格納するために動的に構成されたディレクトリ名を使用して、日次でデータベースをバックアップするイベントが作成されます。

```
CREATE EVENT NightlyBackup
SCHEDULE
START TIME '23:00' EVERY 24 HOURS
HANDLER
BEGIN
DECLARE dest LONG VARCHAR;
DECLARE day_name CHAR(20);
SET day_name = DATENAME( WEEKDAY, CURRENT DATE );
SET dest = 'd:¥¥temp¥¥SQLAnybackup¥¥' || day_name;
BACKUP DATABASE DIRECTORY dest
TRANSACTION LOG RENAME;
END;
```

## 次のステップ

作成したバックアップからリカバリができることを確認します。

## 関連情報

[バックアップの種類 \[896 ページ\]](#)

[データベースのリカバリ \[915 ページ\]](#)

[ディスクサンドボックス \[1604 ページ\]](#)

### 1.6.1.1.2.3 SQL Central を使用したデータベースのバックアップ

SQL Central のウィザードを死傷してさまざまな種類のバックアップを実行します。

## 前提条件

BACKUP DATABASE および VALIDATE ANY OBJECT システム権限が必要です。

テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

## コンテキスト

[バックアップイメージ作成ウィザード](#)を使用して、イメージバックアップを作成します。このバックアップによってデータベースに関連付けられている各ファイルのコピーが作成されます。

[データベースバックアップウィザード](#)を使用して、データベースと任意の関連付けられたファイルを含むアーカイブバックアップ (単一ファイル) を作成します。

ディスクサンドボックス機能が有効で、サンドボックスの外部にバックアップを行っている場合、データベースサーバにセキュリティ機能キーが必要です。

ディスクにバックアップするときは、データベースのロケーションとは異なるコンピュータ上のロケーションを選択します。これにより、データベースが実行中のコンピュータでメディア障害が発生する場合に備えることができます。

## 手順

1. データベースを検証します。
  - a. データベースを右クリックし、[データベースの検証](#) をクリックします。
  - b. [データベース検証ウィザード](#)の指示に従います。ただし、[検証タイプ](#)の選択画面では、次のオプションが選択されていることを確認します。
    - [データベースの検証](#)を選択して、[フルチェック](#)を選択します。
    - [テーブルおよびマテリアライズドビューの検証](#)を選択して、[標準チェック](#)を選択します。
2. データベースをバックアップします。

オプション	アクション
イメージバックアップ	<ol style="list-style-type: none"><li>1. データベースを右クリックして、<a href="#">バックアップイメージの作成</a>をクリックし、ウィザードの指示に従います。</li><li>2. バックアップするファイルを確認するとき<ul style="list-style-type: none"><li>○ フルバックアップについては、<a href="#">すべてのデータベースファイルおよびトランザクションログファイル</a>をクリックします。</li><li>○ インクリメンタルバックアップについては、<a href="#">トランザクションログファイルのみ</a>を選択します。</li></ul></li></ol>
アーカイブバックアップ	<ol style="list-style-type: none"><li>1. データベースを右クリックして、<a href="#">データベースのバックアップ</a>をクリックし、ウィザードの指示に従います。アーカイブバックアップを作成するときのみフルバックアップを実行することができます。</li></ol>

## 結果

バックアップが実行され、設定ファイル `backup.syb` にエントリが作成されます。

## 次のステップ

作成したバックアップからリカバリができることを確認します。これを行わないと、データベースの障害時にバックアップからリカバリが行えず、データが消失することがあります。

## 関連情報

[バックアップの種類](#) [896 ページ]

[データベースのリカバリ](#) [915 ページ]

[アーカイブバックアップからのデータベースのリストア \(SQL Central\)](#) [930 ページ]

### 1.6.1.1.3 バックアップとリカバリ方式の設計

データを保護するために、バックアップおよびリカバリ方式を実装することができます。

良いバックアップ方式では、フルバックアップとインクリメンタルバックアップの組み合わせと、場合によってはライブバックアップも組み込まれています。また、バックアップの前にデータベースの検証をし、読み込み専用モードでバックアップを検証して、リカバリに使用できることを確認します。バックアップを作成する頻度は、データの重要性、データが変更される頻度などの要因によって異なります。一般的には、週ごとにフルバックアップを行い、1日1回トランザクションログのインクリメンタルバックアップを実行するというスケジュールから始めます。フルバックアップもインクリメンタルバックアップも、データベースが実行中(オンライン)に行うことができます。

バックアップとリカバリのプランを立てるときには、次のような点を検討する必要があります。

- バックアップが必要なファイル
- データベースファイルの場所
- バックアップファイルの格納場所
- 保持するフルバックアップの数新しいバックアップでバックアップを上書きすると、バックアップの最中にメディア障害が発生した場合、バックアップは失われてしまいます。複数のフルバックアップのコピーを保持してください。
- バックアップが実行中にデータベースサーバが使用可能であり続ける必要があるか火事、洪水、地震、盗難、その他の破壊行為に備えて、フルバックアップのコピーをオフサイトに保管してください。
- データベースへのアクセスなしに組織が機能できる時間(メディア障害時の最大のリカバリ時間)リカバリの所要時間は、使用可能なハードウェア、データベースファイルのサイズ、リカバリに使用する媒体、ディスク領域、予期しないエラーなどの外部要因の影響を受けます。バックアップ方式を計画するときには、必要に応じてリカバリコマンドの入力、テープの検索やロードなどのタスクにかかる時間も考慮してリカバリの所要時間を決めてください。
- データベースにレプリケーションが含まれるか(そうである場合、トランザクションログのミラーリングの使用を検討してください)。

#### バックアップおよびリカバリ方式の作成におけるハイレベルステップ

1. データベース検証のコマンドを含め、バックアップとリカバリのコマンドを作成して検証します。
2. バックアップおよびリカバリコマンドの実行にかかる時間を計ります。
3. バックアップコマンドを記録し、バックアップの格納先を記述した手順書を作成します。手順書には、使用する命名規則、実行するバックアップの種類も明記しておきます。
4. メンテナンスプランを使用して、プロダクションサーバのバックアッププロセスを自動化/スケジュールします。
5. 予期しないエラーを防止するために、バックアップ手順をモニタします。プロセスに変更があった場合はドキュメント化します。

このセクションの内容:

##### [メンテナンスプランによる検証とバックアップの自動化 \[909 ページ\]](#)

メンテナンスプランは、1つまたは複数のメンテナンスタスクを自動的に行うスケジュールです。

##### [バックアップおよびリカバリプランにおける検証 \[912 ページ\]](#)

定期的にデータベースを検証して、作成したバックアップが破損していないことを確認します。

## 関連情報

[バックアップの種類 \[896 ページ\]](#)

[スケジュールとイベントの使用によるタスクの自動化 \[950 ページ\]](#)

[同期やレプリケーションに関連するデータベースの考慮事項 \[914 ページ\]](#)

### 1.6.1.1.3.1 メンテナンスプランによる検証とバックアップの自動化

メンテナンスプランは、1 つまたは複数のメンテナンスタスクを自動的に行うスケジュールです。

SQL Central にメンテナンスプランを設定します。設定できるメンテナンスタスクの種類は次の通りです。

- 検証
- バックアップ
- 検証タスク前、またはバックアップタスク後のユーザ定義の操作 (SQL 文)

複数のメンテナンスプランを作成することが可能ですが、一度に実行できるのは 1 つのメンテナンスプランのみです。メンテナンスプランが実行されるたび、メンテナンスプランレポートがデータベースに格納されます。SQL Central のメンテナンスプランレポートを表示するか、レポートを電子メールで送信します。

このセクションの内容:

[メンテナンスプランの作成 \[909 ページ\]](#)

SQL Central を使用して、メンテナンスプランを作成します。

[メンテナンスプランレポートの表示 \[910 ページ\]](#)

SQL Central のメンテナンスプランレポートを表示します。

[メンテナンスプランの削除 \[911 ページ\]](#)

SQL Central を使用して、メンテナンスプランを削除します。

#### 1.6.1.1.3.1.1 メンテナンスプランの作成

SQL Central を使用して、メンテナンスプランを作成します。

#### 前提条件

メンテナンスプランの作成には、SELECT ANY TABLE システム権限が必要です。

また、maint\_plan テーブルの MANAGE ANY EVENT システム権限、および INSERT ANY TABLE システム権限または INSERT 権限が必要です。

このタスクを初めて実行する場合と、最後にメンテナンスプランがアクセスされて以降データベースがアップグレードされた場合、メンテナンスプランテーブルを作成または更新する必要があることを示すエラーが表示されます。これらのテーブルを作成

および更新するには、SELECT ANY TABLE システム権限と、CREATE ANY OBJECT システム権限か CREATE ANY INDEX と CREATE ANY TABLE システム権限の両方が必要です。[メンテナンスプランフォルダ](#)をクリックして、テーブルを作成または更新します。

## 手順

1. SQL Central の左ウィンドウ枠で [メンテナンスプラン](#) を右クリックし、**新規** ▶ [メンテナンスプラン](#) をクリックします。
2. ウィザードの指示に従います。

## 結果

メンテナンスプランが作成され、指定したスケジュールを使用して実行されます。

## 関連情報

[バックアップの種類 \[896 ページ\]](#)

### 1.6.1.1.3.1.2 メンテナンスプランレポートの表示

SQL Central のメンテナンスプランレポートを表示します。

## 前提条件

メンテナンスプランレポートを表示するには、SELECT ANY TABLE システム権限が必要です。

最後にメンテナンスプランがアクセスされて以降データベースがアップグレードされた場合、メンテナンスプランテーブルをアップグレードする必要があることを示すエラーが表示されます。これらのテーブルをアップグレードするには、SELECT ANY TABLE システム権限と、CREATE ANY OBJECT システム権限か CREATE ANY INDEX と CREATE ANY TABLE システム権限の両方が必要です。

## 手順

1. SQL Central の左ウィンドウ枠で、[メンテナンスプラン](#) をクリックします。
2. 左ウィンドウ枠で、[メンテナンスプラン](#) をクリックします。

3. 右ウィンドウ枠で、レポートをダブルクリックします。

## 結果

メンテナンスプランのプロパティ ウィンドウが表示されます。詳細 ウィンドウ枠に、メンテナンスプランのレポートが表示されます。

## 関連情報

[バックアップの種類 \[896 ページ\]](#)

### 1.6.1.1.3.1.3 メンテナンスプランの削除

SQL Central を使用して、メンテナンスプランを削除します。

## 前提条件

メンテナンスプランの削除には、SELECT ANY TABLE システム権限が必要です。

MANAGE ANY EVENT システム権限と DELETE ANY TABLE システム権限も必要です。

最後にメンテナンスプランがアクセスされて以降データベースがアップグレードされた場合、メンテナンスプランテーブルをアップグレードする必要があることを示すエラーが表示されます。これらのテーブルをアップグレードするには、SELECT ANY TABLE システム権限と、CREATE ANY OBJECT システム権限か CREATE ANY INDEX と CREATE ANY TABLE システム権限の両方が必要です。

## 手順

1. SQL Central の左ウィンドウ枠で、[メンテナンスプラン](#) をクリックします。
2. メンテナンスプランを右クリックして、[削除](#) をクリックします。
3. 削除するかどうかを確認する画面が表示された場合は、[はい](#) をクリックします。

## 結果

メンテナンスプランが削除され、メンテナンスプランの一部として実行されていた検証およびバックアップは再び実行されません。

## 関連情報

[バックアップの種類 \[896 ページ\]](#)

### 1.6.1.1.3.2 バックアップおよびリカバリプランにおける検証

定期的にデータベースを検証して、作成したバックアップが破損していないことを確認します。

ベストプラクティスは、通常のデータベースの検証のあとに自動的に実行されるメンテナンスプランを作成することです。

データベースファイル内のベーステーブルの破損が検証で見つかった場合は、メディア障害として対処し、前のバックアップからリカバリします。ただし、破損がインデックス上であれば、インデックスを削除して再作成する方が簡単です。

検証では、検証されるオブジェクトに対する排他アクセスが必要なので、データベース上に他のアクティビティが存在しないときに検証を行うことをお奨めします。

検証を行うことによって、バックアップからリカバリができることを確認する状況がいくつかあります。

#### バックアップ間

データベースファイルの破損は、データベースサーバがデータベース内の破損部分にアクセスするまで判明しないことがあります。バックアップとリカバリプランの一環として、定期的にバックアップ間のデータベースを検証します。

#### バックアップ前

すべてのバックアッププロシージャには、バックアップ前の検証ステップが必要です。

#### バックアップ後

バックアップ作成後すぐに、検証を行う必要があります。検証のためにデータベースのバックアップコピーを起動するときは、-ds データベースオプションを使用して、DB 領域ファイルとトランザクションログのロケーションを指定します。そのようにすることで、元のデータベースを稼働させたままで、データベースのバックアップコピーを元のデータベースと同じコンピュータ上で起動できます。

## バックアップ時に開いているトランザクションがあったデータベースの検証

インメモリ検証モードでデータベースのバックアップを開始し、その後データベースを検証します。インメモリモードによって、データベースへの書き込みを防ぎながら、データベースのリカバリをデータベースサーバ上で実行できます。データベースに保留中のトランザクションがある場合は、読み込み専用モード (-r データベースオプション) を使用することはできません。

## 関連情報

[データベースの検証 \[942 ページ\]](#)

[メンテナンスプランによる検証とバックアップの自動化 \[909 ページ\]](#)

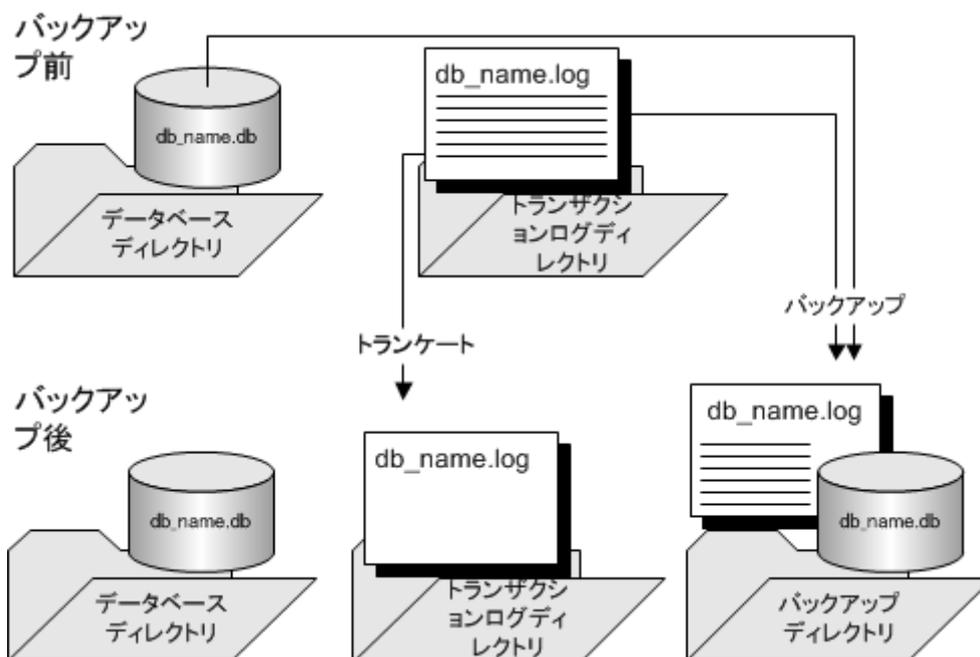
[トランザクションログの検証 \[281 ページ\]](#)

### 1.6.1.1.4 バックアップ中のトランザクションログで発生すること

バックアップを作成中、指定しない限り、データベースサーバはトランザクションログのコピーを作成して、トランザクションログが大きくなり続けるままにします。

ディスク領域は制限されており、トランザクションログを無制限に増大させることは現実的ではありません。バックアップが完了したらトランザクションログの内容を削除することを選択できます。ただし、データベースがレプリケーションに関連する場合は、レプリケーションではトランザクションログへのアクセスを必要とするので、このオプションを選択しないでください。

次の図で、トランザクションログファイルをトランケートするフルバックアップを説明します。



通常、トランザクションログのバックアップコピーは、オンライントランザクションログよりも小さくなります。この違いはデータベースサーバがアクティブなトランザクションログに 64 KB 単位で領域を割り当てるためで、このためサイズには空のページが含まれます。しかし、空でないページだけがバックアップされ、これがバックアップコピーのファイルサイズが小さくなる要因です。

各インクリメンタルバックアップ後にトランザクションログを削除すると、データベースファイルのメディア障害からリカバリするタスクが複雑になります。データベースを最新の状態にするには各トランザクションログを順番に適用する必要がありますが、最後のフルバックアップ以降に複数の異なるトランザクションログが作成されている可能性があります。

## 関連情報

[バックアップの種類 \[896 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) を使用したデータベースのバックアップ \[902 ページ\]](#)

[BACKUP DATABASE 文を使用したデータベースのバックアップ \[904 ページ\]](#)

[SQL Central を使用したデータベースのバックアップ \[906 ページ\]](#)

### 1.6.1.1.5 同期やレプリケーションに関連するデータベースの考慮事項

同期やレプリケーションの環境では、トランザクションログの名前を変更してトランザクションログを再開するようにバックアップオプションを使用します。

これにより、古いトランザクションの情報を保ちつつ、トランザクションログが無限に拡大するのを防ぐことができます。

データベースが SQL Remote インストール環境の一部である場合、Message Agent は古いトランザクションにアクセスする必要があります。統合データベースの場合、データベース内に SQL Remote インストール環境全体のマスターコピーが格納されるので、データを失わないためには完全なバックアップ手順が必要です。

データベースが Mobile Link の設定に関連していて、dbmlsync を使用している場合は、同じことに注意する必要があります。ただし、データベースが Mobile Link の統合データベースである場合は、古いトランザクションログは必要ありません。

リモートデータベースでは、バックアップ手順は統合データベースの場合ほど重要ではありません。データのバックアップを、統合データベースへのレプリケーションに頼る方法もあります。ただし、メディア障害が発生した場合は、統合データベースからリモートデータベースを再抽出しなければならず、レプリケートされていないオペレーションは失われます。ログ変換ユーティリティを使用して、失われたオペレーションのリカバリを実行することは可能です。

レプリケーションに頼ってリモートデータベースのデータを保護する場合でも、トランザクションログが大きくなりすぎるのを防ぐために、リモートデータベースでバックアップを定期的に行う必要があります。統合データベースで使用するのと同じオプション (トランザクションログの名前変更と再起動) を使用して Message Agent を実行し、Message Agent が名前変更されたログファイルにアクセスできるようにします。リモートデータベースで delete\_old\_logs オプションを On に設定すると、不要になった古いトランザクションログファイルが Message Agent によって自動的に削除されます。

### SQL Remote でのトランザクションログの自動名前変更

Message Agent の -x オプションを使用すると、データベースサーバが停止した際に、リモートコンピュータでトランザクションログの名前を変更する必要がなくなります。-x オプションは、トランザクションログが出力メッセージ用にスキャンされた後で、トランザクションログの名前を変更します。

## 関連情報

[バックアップの種類 \[896 ページ\]](#)

[トランザクションログの名前変更またはトランケート \(SQL Central の場合\) \[284 ページ\]](#)

[ログ変換ユーティリティ \(dbtran\) \[1133 ページ\]](#)

[delete\\_old\\_logs オプション \[Mobile Link\]\[SQL Remote\] \[695 ページ\]](#)

## 1.6.1.2 データベースのリカバリ

リカバリは、インクリメンタルトランザクションログファイルを使用してデータベースファイル、トランザクションログ、DB 領域をリストアし、データベースファイルをできるだけ最新の状態にすることです。

バックアップおよびリカバリプランの一部として、常にバックアップを検証します。有効なデータベースのバックアップコピーからのみリカバリします。

### 自動リカバリ

データベースの起動時に、データベースサーバは直前のセッションの最後にデータベースが正しく停止したかどうかをチェックします。正しく停止されていない場合、データベースサーバは自動リカバリ処理を実行して、最後にコミットされたトランザクションまでのすべての変更をリストアします。データベースとトランザクションログのバックアップコピーには、リカバリ前にどのような変更も加えてはいけません。

### 手動リカバリ

1つ以上のトランザクションログをデータベースバックアップのコピーに適用して、データベースを前のバージョンのデータベースにリカバリします。リカバリ処理で実行する手順は、バックアップ処理でインクリメンタルバックアップのトランザクションログに変更を加えなかったかどうかによって異なります。バックアップの処理中にトランザクションログの削除または名前の変更を行った場合、複数のトランザクションログに加えられた変更を適用する必要があります。バックアップ処理でトランザクションログに変更が加えられていない場合は、リカバリのときにはオンライントランザクションログだけを使用すれば済みます。

トランザクションログが複数ある場合、トランザクションが複数のトランザクションログにまたがっている可能性があります。リカバリのときには、トランザクションログを正しい順序で適用します。適用しない場合、複数のトランザクションログにまたがっているトランザクションがロールバックされます。トランザクションログの正しい適用順序をデータベースサーバに判断させる場合は、データベースサーバオプションとして `-ad` を指定します。

ポイントインタイムリカバリ (PITR) を実行することによって、データベースをリストアすることができます。データベースサーバが実行する操作は、関連付けられているオフセットの数値により、トランザクションログの中でユニークに識別される必要があります。トランザクションログの COMMIT および CHECKPOINT 操作には、これらに関連付けられるタイムスタンプがあります。ポイントインタイムリカバリを実行して、トランザクションログのオフセットまたはタイムスタンプにリカバリすることができます。より高い精度を求める場合は、オフセットのリストアが最適です。リストアするトランザクションを指定することができます。

### データベースリカバリ中のパフォーマンス

データベースリカバリには時間がかかることがあります。データベースのリカバリ時間に影響するいくつかの要因があります。

負荷の同時実行レベル 負荷の同時実行レベルが大きいほど、リカバリ時間が短縮されます。

リカバリを実行するマシンで使用可能なハードウェアリソース CPU コアおよびディスク I/O の使用可能なスピンドルが多いほど、リカバリに必要な時間が短くなります。

トランザクションログでの DDL 操作と DML 操作の組合せ DML 操作は、シリアル化が必要でデータベースのリカバリにかかる時間を増大させる DDL 操作よりも、並列リカバリによる利点を得ることができます。

プライマリキーの存在 プライマリキーのあるテーブルでの操作では、リカバリプロセスが大幅に高速化します。

このセクションの内容:

#### [自動リカバリ処理 \[917 ページ\]](#)

データベースが停止すると、データベースサーバでチェックポイントが実行され、データベース内の情報がデータベースファイル内に格納されます。

#### [ポイントインタイムリカバリ \[917 ページ\]](#)

ポイントインタイムリカバリ (PITR) によって、タイムスタンプの値、またはトランザクションログのオフセットとして指定した特定時点のデータベースのリストアが可能になります。

#### [コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

SQL Central を使用して、コミットされていない操作をリカバリし、すべてのコミットされたトランザクションをデータベースに再適用します。

#### [コミットされていない操作のリカバリ \(コマンドライン\) \[919 ページ\]](#)

dbtran ユーティリティを使用して、コミットされていない操作をリカバリし、すべてのコミットされたトランザクションをデータベースに再適用します。

#### [チュートリアル: タイムスタンプへのデータベースのリストア \[920 ページ\]](#)

トランザクションログのタイムスタンプへ、データベースをリストアします。

#### [チュートリアル: トランザクションログのオフセットへのデータベースのリストア \[925 ページ\]](#)

トランザクションログのオフセットへ、データベースをリストアします。

#### [イメージバックアップからのリストア \[929 ページ\]](#)

適用するインクリメンタルトランザクションログ がないデータベースを、イメージバックアップからリストアします。

#### [アーカイブバックアップからのデータベースのリストア \(SQL Central\) \[930 ページ\]](#)

SQL Central を使用して、アーカイブバックアップからデータベースをリストアします。

#### [アーカイブバックアップからのデータベースのリストア \(Interactive SQL\) \[931 ページ\]](#)

Interactive SQL を使用して、アーカイブバックアップからデータベースをリストアします。

#### [ライブバックアップからのデータベースの再開 \[933 ページ\]](#)

運用データベースを実行するプライマリコンピュータとは別のコンピュータに作成されたライブバックアップから、データベースを再開します。

#### [トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

データベースをリカバリする必要があり、複数のトランザクションログがある場合は、正しい順序でトランザクションログファイルをバックアップコピーに適用します。

#### [メディア障害からのリカバリ \[937 ページ\]](#)

システム障害とメディア障害の 2 種類の障害によって、データベースが使用不可になることがあります。

## 関連情報

### 1.6.1.2.1 自動リカバリ処理

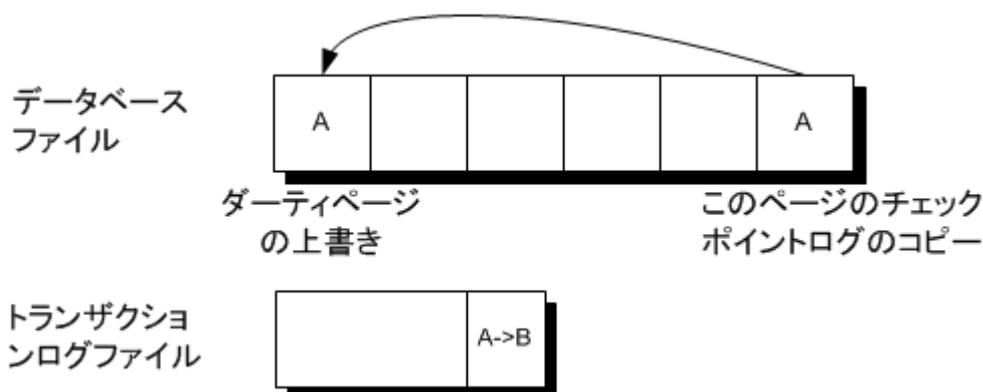
データベースが停止すると、データベースサーバでチェックポイントが実行され、データベース内の情報がデータベースファイル内に格納されます。

このプロセスはクリーンシャットダウンと呼ばれます。

データベースを起動するたびに、データベースサーバは最後の停止がクリーンだったのか、システム障害の結果だったのかをチェックします。データベースの停止がクリーンでなかった場合は、システム障害からリカバリするために、次の手順が自動的に実行されます。

最新のチェックポイントにリカバリする

すべてのページを最新のチェックポイント時の状態にリストアするために、チェックポイントログページのコピーでチェックポイント以降に加えられた変更が上書きされます。



チェックポイント以降に加えられた変更を適用する

チェックポイントからシステム障害が発生するまでの間に加えられた変更が適用されます。この変更内容はトランザクションログに格納されています。

コミットされていないトランザクションのロールバック

コミットされていないトランザクションが、ロールバックログを使用してロールバックされます。

### 1.6.1.2.2 ポイントインタイムリカバリ

ポイントインタイムリカバリ (PITR) によって、タイムスタンプの値、またはトランザクションログのオフセットとして指定した特定時点のデータベースのリストアが可能になります。

操作はトランザクションログに逐次記録され、すべての操作がファイルにおける数値のオフセットでユニークに識別されます。それまでデータベースが正常だった時点の操作を識別できる場合は、そのオフセットに最も近い COMMIT または CHECKPOINT 操作にリカバリすることが推奨されています。この操作はオフセットへのポイントインタイムリカバリと呼ばれます。

トランザクションログの COMMIT および CHECKPOINT 操作には、実行された時間を記録したタイムスタンプがあります。データベースがそれ以前には正常に稼働していた時刻を知っている場合、その時刻にリストアすることができます。データベースサーバは指定されたタイムスタンプの後、最も近い COMMIT または CHECKPOINT にリカバリします。この操作はオフセットタイムスタンプへのポイントインタイムリカバリと呼ばれます。

ポイントインタイムリカバリを実行するときに推奨される方法は、変換ユーティリティ (dbtran) を使用してトランザクションログの操作を検査し、オフセットを使用して操作のリカバリを行うことです。

ポイントインタイムリカバリは典型的なリカバリ操作と同じように、-a オプションのいずれか (-a、-ad、-ar など) を指定して開始されます。データベースサーバでは、複数のディレクトリにある該当するトランザクションログファイルからトランザクションを見つけて適用することができます。

ポイントインタイムリカバリは 1 つの重大な点において他のリカバリ方法とは異なります。リカバリ中は、トランザクションログを一度に適用することのみできます。リカバリの最後に適用すべきトランザクションログとトランザクションを見つけた場合、リカバリをもう一度実施し、-ad 句で指定したディレクトリのリスト内の新しいトランザクションログの場所を入れる必要があります。

ポイントインタイムリカバリの最後で、データベースは実施する最後の COMMIT 操作のタイムスタンプとオフセットを表示します。

## 関連情報

[データベースのリカバリ \[915 ページ\]](#)

[チュートリアル: トランザクションログのオフセットへのデータベースのリストア \[925 ページ\]](#)

[チュートリアル: タイムスタンプへのデータベースのリストア \[920 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

[-ru データベースオプション \[532 ページ\]](#)

[-ruo データベースオプション \[533 ページ\]](#)

### 1.6.1.2.3 コミットされていない操作のリカバリ (SQL Central)

SQL Central を使用して、コミットされていない操作をリカバリし、すべてのコミットされたトランザクションをデータベースに再適用します。

## コンテキスト

場合によっては、障害時に完了していないトランザクションの情報を検索する必要があります。データベースファイル上のメディア障害からリカバリするときには、トランザクションログが正常である必要があります。

[ログファイル変換ウィザード](#)を使用すると、SQL Central でトランザクションログファイルを .sql ファイルに変換できます。

#### i 注記

トランザクションログには、障害が発生する直前の変更が含まれていないこともあります。最後にコミットしたトランザクションより前にデータベースに加えられた変更は、トランザクションログに含まれます。

## 手順

1. SQL Central で、**ツール** ▶ **SQL Anywhere17** ▶ **ログファイルの変換** をクリックします。
2. ウィザードの指示に従います。
3. 変換したトランザクションログ (SQL スクリプトファイル) をテキストエディタで編集し、必要な説明を特定します。

## 結果

コミットされていない操作がリカバリされます。

## 関連情報

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[メディア障害からのリカバリ \[937 ページ\]](#)

[イメージバックアップからのリストア \[929 ページ\]](#)

[アーカイブバックアップからのデータベースのリストア \(SQL Central\) \[930 ページ\]](#)

[ライブバックアップからのデータベースの再開 \[933 ページ\]](#)

### 1.6.1.2.4 コミットされていない操作のリカバリ (コマンドライン)

dbtran ユーティリティを使用して、コミットされていない操作をリカバリし、すべてのコミットされたトランザクションをデータベースに再適用します。

## コンテキスト

場合によっては、障害時に完了していないトランザクションの情報を検索する必要があります。データベースファイル上のメディア障害からリカバリするときには、トランザクションログが正常である必要があります。

#### **i** 注記

トランザクションログには、障害が発生する直前の変更が含まれていないこともあります。最後にコミットしたトランザクションより前にデータベースに加えられた変更は、トランザクションログに含まれます。

## 手順

1. `-a` オプションを使用してコミットされていないトランザクションを含み、`dbtran` を実行してトランザクションログファイルを SQL スクリプトファイルに変換します。
2. 変換したトランザクションログ (SQL スクリプトファイル) をテキストエディタで編集し、必要な説明を特定します。

## 結果

コミットされていない操作がリカバリされます。

### 例

次のコマンドを実行して、トランザクションログファイル `sample.log` を `changes.sql` ファイルに変換します。

```
dbtran -a sample.log changes.sql
```

## 関連情報

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[メディア障害からのリカバリ \[937 ページ\]](#)

[イメージバックアップからのリストア \[929 ページ\]](#)

[アーカイブバックアップからのデータベースのリストア \(SQL Central\) \[930 ページ\]](#)

[ライブバックアップからのデータベースの再開 \[933 ページ\]](#)

[ログ変換ユーティリティ \(dbtran\) \[1133 ページ\]](#)

### 1.6.1.2.5 チュートリアル: タイムスタンプへのデータベースのリストア

トランザクションログのタイムスタンプへ、データベースをリストアします。

## 前提条件

このチュートリアルで使用するテーブルを作成するには、`CREATE TABLE` システム権限が必要です。

このチュートリアルのバックアップのステップを実行するには、`BACKUP DATABASE` システム権限が必要です。

データベースのリストアを実行するために必要な権限は、`-gu` データベースオプションの設定と `SERVER OPERATOR` システム権限の有無によって異なります。

## コンテキスト

おおよその特定時点へデータベースをリストアすることができます。たとえば、ある特定時点の後データベースが使用不可になった場合、その時間の直前のタイムスタンプへリカバリすることができます。

タイムスタンプへリカバリするとき、データベースサーバはタイムスタンプに関連する隣接の COMMIT 操作へリストアします。タイムスタンプへのポイントインタイムリカバリは、トランザクションログのオフセットへのリカバリよりも精度が低くなっています。

リストアコマンドを実行するとき、リストアに失敗すると、リストアしようとするデータベースのコピーがリストアに使用できなくなります。リストアはバックアップファイル自体で行うのではなく、バックアップファイルのコピーで行うことをおすすめします。そうすることにより、新しいコピーを作成することができます。

時間と手間を節約するために、リストアコマンドに `-im v` オプションを追加します (たとえば `dbsrv17 -im v myDB.db -as -ar -ru "2014-12-15 15:23:15.348"`)。このオプションによって、データベースを永久的に変更することなく、リストアコマンドが正常に実行されたことをテストすることができます。リストアが失敗した場合、バックアップファイルの新しいコピーを作成する必要はありません。リストアコマンドを変更して、再度実行することができます。コマンドが正常に実行された場合、読み込み専用のリストアされたデータベースに接続し、データベースが正しい特定時点にリストアされたことを確認できます。データベースが正しい特定時点にリストアされた場合、データベースをシャットダウンして、`-im v` オプションなしでリストアコマンドを再度実行して、永久的に指定した特定時点にデータベースをリストアします。

## 手順

1. [Interactive SQL](#) でサンプルデータベース (demo.db) に接続します。
2. 次の文を実行してバックアップディレクトリ `c:¥temp¥backup_dir¥` を作成し、データベースファイルをそこにバックアップします。

```
BACKUP DATABASE DIRECTORY 'c:¥temp¥backup_dir' DBFILE ONLY;
```

`c:¥temp¥backup_dir¥` ディレクトリには demo.db データベースのバックアップコピーが格納されています。

3. 次の文を実行して `trackingTable` と呼ばれるテーブルを作成します。`trackingTable` は後でデータベースをどのタイムスタンプにリストアしたか確認するために使用します。INSERT 文は初期値を設定し、COMMIT 文はデータベースへの変更をコミットします。

```
CREATE TABLE trackingTable ( myValue INT );  
INSERT INTO trackingTable VALUES (1);  
COMMIT;
```

4. 次の文を実行して、現時刻のタイムスタンプを返します。この例では、タイムスタンプは 2014-12-15 15:23:15.348 です。タイムスタンプは異なるので、後で使用するために記録します。
5. タイムスタンプの値を初期化するには、[ツール](#) > [オプション](#) をクリックして、[SQL Anywhere](#) カテゴリを選択します。[日付および時刻の設定を初期化をローカルに設定し、結果の表示方法をテキストに設定します。](#)

```
SELECT CURRENT TIMESTAMP;
```

### 結果の例

```
current timestamp  
-----  
2014-12-15 15:23:15.348
```

(1 rows)

### i 注記

このチュートリアルでは単純化のため、CURRENT\_TIMESTAMP の特別値を問い合わせ、データベースをリストアする時刻を書き留めることにより、データベースをリストアするタイムスタンプを決定します。ただし、実際のデータベースのリストアでは、ログ変換ユーティリティ (dbtran) を使用してデータベースをリストアするタイムスタンプを決定するか、データベースをリストアする時刻に一致するタイムスタンプを指定します。

6. 次の文を実行して、以前のステップでデータベースをバックアップしたディレクトリに、トランザクションログをバックアップします。

```
BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥backup_dir' TRANSACTION LOG ONLY  
TRANSACTION LOG RENAME MATCH;
```

この文では、<integer>AA.log (たとえば、141212AA.LOG) のような名前を持つトランザクションログファイルのバックアップを作成します。

7. 次の文を実行して、trackingTable テーブルにローを追加して、データベースの変更をコミットします。

```
INSERT INTO trackingTable VALUES ( 2 );  
COMMIT;
```

8. 次の文を実行して、現時刻のタイムスタンプを返します。この例では、タイムスタンプは 2014-12-15 15:24:14.922 です。値は異なるので、書き留めます。

```
SELECT CURRENT_TIMESTAMP;
```

### 結果の例

```
current timestamp  
-----  
2014-12-15 15:24:14.922  
(1 rows)
```

9. 次の文を実行して、trackingTable テーブルに 2 つのローがあることを確認します。

```
SELECT * FROM trackingTable;
```

### 結果の例

```
myValue  
-----  
1  
2  
  
(2 rows)
```

10. 次の文を実行して、データベースをバックアップしたディレクトリに、トランザクションログの別のバックアップを作成します。

```
BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥backup_dir' TRANSACTION LOG ONLY  
TRANSACTION LOG RENAME MATCH;
```

これにより、<integer>AB.log (たとえば、141212AB.LOG) のような名前を持つログファイルのバックアップが作成されます。

c:¥temp¥backup\_dir ディレクトリには、データベース (demo.db) のバックアップコピーと、2 つのユニークに名付けられたログファイルのバックアップコピーが存在します。これで、データベースを以前書き留めた最初のタイムスタンプへリストアする準備ができました。

11. データベースを切断し、シャットダウンします。
12. c:¥temp¥backup\_dir を c:¥temp¥restore\_dir という名の新しいディレクトリにコピーします。このディレクトリからデータベースをタイムスタンプへリストアします。

### i 注記

バックアップファイルから直接ではなく、バックアップファイルのコピーからリストアを行うことをおすすめします。これにより、問題が発生した場合には新しいコピーを作成してリストアを再試行することができます。

13. c:¥temp¥restore\_dir で、次のコマンドを実行してデータベースをチュートリアルで書き留めた最初のタイムスタンプにリストアします (この例では、2014-12-15 15:23:15)。

```
dbsrv17 demo.db -as -ar -ru "2014-12-15 15:23:15 -05:00"
```

タイムスタンプの仕様にタイムゾーン (-05:00) を含めるのはベストプラクティスで、タイムスタンプへのポイントインタイムリカバリを行うときには強くおすすめします。このケースで -5:00 は、トランザクションログの時間が UTC マイナス 5 時間であることをデータベースサーバに教えています。タイムスタンプの仕様でタイムゾーンを省略するときは、リストアを行っているデータベースサーバのローカル時間が使用されます。データベースサーバとログが同じタイムゾーンの場合、これは問題になりません。ただし同じタイムゾーンでない場合、データベースはデータベースサーバのローカル時間に変換されたタイムスタンプにリストアしますが、これは正しくありません。

14. *Interactive SQL* を使用して、c:¥temp¥restore\_dir¥ の demo.db データベースに接続し、次の文を実行して trackingTable テーブルのローを問い合わせます。

```
SELECT * FROM trackingTable;
```

### 結果の例

```
myValue
-----
          1

(1 row)
```

書き留めた最初のタイムスタンプは 2 つめのロー (2) をテーブルに挿入する前であるため、trackingTable テーブルに 1 つのローしか存在しないという事実から、指定したタイムスタンプへのリストアが正常に行われたことが確認できます。

15. 次の文を実行して、trackingTable テーブルにローを追加して、データベースへの変更をコミットします。

```
INSERT INTO trackingTable VALUES ( 103 );
COMMIT;
```

16. 次の文を実行して、現時刻のタイムスタンプを返します。この例では、タイムスタンプは 2014-12-15 15:54:23.84 です。値は異なるので、後で使用するために記録します。

```
SELECT CURRENT_TIMESTAMP;
```

### 結果の例

```
current timestamp
-----
2014-12-15 15:54:23.84
```

```
(1 rows)
```

17. 次の文を実行して、trackingTable テーブルにローを追加して、データベースの変更をコミットします。

```
INSERT INTO trackingTable VALUES ( 104 );  
COMMIT;
```

18. 次の文を実行して、現時刻のタイムスタンプを返します。この例では、タイムスタンプは 2014-12-15 15:55:33.048 です。値は異なるので、後で使用するために記録します。

```
SELECT CURRENT_TIMESTAMP;
```

#### 結果の例

```
current timestamp  
-----  
2014-12-15 15:55:33.048  
(1 rows)
```

19. 次の文を実行して、以前のステップでデータベースをバックアップしたディレクトリにトランザクションログをバックアップします。

```
BACKUP DATABASE DIRECTORY 'c:¥temp¥backup_dir' TRANSACTION LOG ONLY  
TRANSACTION LOG RENAME MATCH;
```

この文によって c:¥temp¥backup\_dir にログファイルの 3 つめのバックアップが作成されます。

20. データベースを切断し、データベースサーバを停止します。
21. c:¥temp¥backup\_dir のコンテンツを、c:¥temp¥restore\_dir2 (末尾の 2 に注意) という名前の新しいディレクトリにコピーします。
22. c:¥temp¥restore\_dir2 で次のコマンドを実行して、trackingTable.myValue に値 103 を追加した後書き留めたタイムスタンプ (この例では 2014-12-15 15:54:23.84) へデータベースをリストアします。

```
dbsrv17 demo.db -as -ar -ru "2014-12-15 15:54:23.84 -05:00"
```

23. *Interactive SQL* で、c:¥temp¥restore\_dir2 の demo.db データベースに接続します。

24. 次の文を実行して、trackingTable テーブルのローを表示します。

```
SELECT * FROM trackingTable;
```

#### 結果の例

```
myValue  
-----  
1  
103  
(2 rows)
```

古いログの値 2 と、新しいログの値 104 が含まれていないことに注目してください。これは、値 2 が最初のバックアップ操作後に挿入されたため、c:¥temp¥restore\_dir¥ のリストアされたデータベースには存在しないためです。同じように、値 104 は 2 回目のバックアップ操作後に挿入されたため、c:¥temp¥restore\_dir2¥ のリストアされたデータベースには存在しません。

25. ポイントインタイムリカバリが完了したら、データベースをシャットダウンしていずれのバックアップに関連するオプション (たとえば、-a、-ad、および -ar) も付けずに、データベースサーバを再起動します。

## 関連情報

[ポイントインタイムリカバリ \[917 ページ\]](#)

[データベースのリカバリ \[915 ページ\]](#)

[チュートリアル: トランザクションログのオフセットへのデータベースのリストア \[925 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

[-ru データベースオプション \[532 ページ\]](#)

[-ruo データベースオプション \[533 ページ\]](#)

### 1.6.1.2.6 チュートリアル: トランザクションログのオフセットへのデータベースのリストア

トランザクションログのオフセットへ、データベースをリストアします。

#### 前提条件

このチュートリアルで使用するテーブルを作成するには、CREATE TABLE システム権限が必要です。

このチュートリアルのバックアップのステップを実行するには、BACKUP DATABASE システム権限が必要です。

データベースのリストアを実行するために必要な権限は、-gu データベースオプションの設定と SERVER OPERATOR システム権限の有無によって異なります。

#### コンテキスト

操作はトランザクションログに逐次記録され、すべての操作がトランザクションログにおける数値のオフセットでユニークに識別されます。それまでデータベースが正常だった時点の操作またはイベントを識別できる場合は、この操作のオフセットまでデータベースをリストアすることができます。

リストアコマンドを実行するとき、リストアに失敗すると、リストアしようとするデータベースのコピーがリストアに使用できなくなります。このためリストアはバックアップファイル自体で行うのではなく、バックアップファイルのコピーで行うことをおすすめします。そうすることにより、常に新しいコピーを作成することができます。

時間と手間を節約するために、リストアコマンドに `-im v` オプションを追加します (たとえば `dbsrv17 -im v myDB.db -as -ar -ruo 1245672`)。このオプションによって、データベースを永久的に変更することなく、リストアコマンドが正常に実行されたことをテストすることができます。リストアが失敗した場合、バックアップファイルの新しいコピーを作成する必要はありません。リストアコマンドを変更して、再度実行することができます。コマンドが正常に実行された場合、読み込み専用のリストアされたデータベースに接続し、データベースが正しい特定時点にリストアされたことを確認できます。データベースが正しい特定時点にリストアされた場合、データベースをシャットダウンして、`-im v` オプションなしでリストアコマンドを再度実行して、永久的に指定した特定時点にデータベースをリストアします。

## 手順

1. *Interactive SQL* でサンプルデータベース (demo.db) に接続します。
2. 次の文を実行してバックアップディレクトリ `c:¥temp¥backup_dir¥` を作成し、データベースファイルをそこにバックアップします。

```
BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥backup_dir' DBFILE ONLY;
```

`c:¥temp¥backup_dir¥` ディレクトリには、サンプルデータベース (demo.db) のバックアップコピーが格納されています。

3. 次の文を実行して `trackingTable` と呼ばれるテーブルを作成します。`trackingTable` は後でデータベースをどのオフセットにリストアしたか確認するために使用します。`INSERT` 文は初期値を設定し、`COMMIT` 文はデータベースへの変更をコミットします。

```
CREATE TABLE trackingTable ( myValue INT );
INSERT INTO trackingTable VALUES (1);
COMMIT;
```

4. 次の文を実行して、トランザクションログから実行する `COMMIT` 文の直後の操作のオフセットを得ます。この例における結果のオフセット値は 1245672 です。ただし、実際のオフセットの数値はこのチュートリアルで返されたものとは異なるので、オフセットの数値を書き留めます。そうすることにより、後でデータベースをリストアするオフセットを指定するときに、(このチュートリアルで示されたオフセットの数値ではなく) その値を指定することができます。

```
SELECT DB_PROPERTY('LastCommitRedoPos');
```

### 結果の例

```
db_property('LastCommitRedoPos')
-----
1245672
(1 rows)
```

### i 注記

チュートリアルでは単純化するために、`LastCommitRedoPos` データベースのプロパティを問い合わせる最後の `COMMIT` 文のオフセットを確認し、データベースをリストアするオフセットを決定します。ただし、実際のデータベースのリストアでは、ログ変換ユーティリティ (`dbtran`) を使用して、データベースをリストアする実際の操作のオフセットを決定します。

5. 次の文を実行して、以前のステップでデータベースをバックアップしたディレクトリに、トランザクションログをバックアップします。

```
BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥backup_dir' TRANSACTION LOG ONLY
TRANSACTION LOG RENAME MATCH;
```

この文では、`<date>AA.log` (たとえば、`141212AA.LOG`) のような名前を持つトランザクションログファイルのバックアップを作成します。

6. 次の文を実行して、`trackingTable` テーブルにローを追加して、データベースへの変更をコミットします。

```
INSERT INTO trackingTable VALUES ( 2 );
COMMIT;
```

7. 次の文を実行して、トランザクションログから実行する COMMIT 文のオフセットを得ます。この例における結果のオフセット値は 1249951 です。ここでも、オフセットの数値はこのチュートリアルで返されたオフセットの数値とは異なるので、オフセットの数値を書き留めます。

```
SELECT DB_PROPERTY('LastCommitRedoPos');
```

#### 結果の例

```
db_property('LastCommitRedoPos')
-----
1249951
(1 rows)
```

8. 次の文を実行して、trackingTable テーブルに 2 つのローがあることを確認します。

```
SELECT * FROM trackingTable;
```

#### 結果の例

```
myValue
-----
      1
      2
(2 rows)
```

9. 次の文を実行して、データベースをバックアップしたディレクトリに、トランザクションログの 2 つめのバックアップを作成します。

```
BACKUP DATABASE DIRECTORY 'c:¥¥temp¥¥backup_dir' TRANSACTION LOG ONLY
TRANSACTION LOG RENAME MATCH;
```

この文では、<integer>AB.log (たとえば、141212AB.LOG) のような名前を持つトランザクションログファイルのバックアップを作成します。

c:¥temp¥backup\_dir には、データベース (demo.db) のバックアップコピーと、2 つのユニークに名付けられたログファイルのバックアップコピーが存在します。これで、データベースを以前書き留めたオフセットへリストアする準備ができました。

10. データベースを切断しシャットダウンします。
11. c:¥temp¥backup\_dir を c:¥temp¥restore\_dir という名前の新しいディレクトリにコピーします。このディレクトリからデータベースをオフセットへリストアします。

#### i 注記

バックアップファイルから直接ではなく、バックアップファイルのコピーからリストアを行うことをおすすめします。これにより、問題が発生した場合には新しいコピーを作成してリストアを再試行することができます。

12. c:¥temp¥restore\_dir で、次のコマンドを実行してデータベースをチュートリアルで書き留めた最初のオフセットにリストアします (この例では、オフセット 1245672)。

```
dbsrv17 demo.db -as -ar -ruo 1245672
```

#### 注記

- -as オプションを指定すると、データベースサーバはデータベースをリストア後稼働状態にします。
- -ar オプションを指定すると、データベースサーバはローカルディレクトリからログを適用します。

- o -ruo オプションを指定すると、データベースサーバはオフセット 1245672 からデータベースのリストアを開始します。

13. *Interactive SQL* で、`c:¥temp¥restore_dir¥` の `demo.db` データベースに接続し、次の文を実行して `trackingTable` テーブルのローを表示します。

```
SELECT * FROM trackingTable;
```

#### 結果の例

```
myValue
-----
          1
(1 row)
```

最初のオフセットは 2 つめの値 (2) をテーブルにコミットする前であるため、`trackingTable` テーブルに 1 つのローしか存在しないという事実から、指定したオフセットへのリストアが正常に行われたことが確認できます。

14. 次の文を実行して、`trackingTable` テーブルにローを追加して、データベースへの変更をコミットします。

```
INSERT INTO trackingTable VALUES ( 103 );
COMMIT;
```

15. 次の文を実行して、トランザクションログから実行する `COMMIT` 文のオフセットを得ます。この例における結果のオフセット値は 1249890 です。オフセットの数値を書き留めます。

```
SELECT DB_PROPERTY('LastCommitRedoPos');
```

#### 結果の例

```
db_property('LastCommitRedoPos')
-----
1249890
(1 rows)
```

16. 次の文を実行して、`trackingTable` テーブルにローを追加して、データベースへの変更をコミットします。

```
INSERT INTO trackingTable VALUES ( 104 );
COMMIT;
```

17. 次の文を実行して、トランザクションログから実行する `COMMIT` 文のオフセットを得ます。この例における結果のオフセット値は 1249914 です。オフセットの数値を書き留めます。

```
SELECT DB_PROPERTY('LastCommitRedoPos');
```

#### 結果の例

```
db_property('LastCommitRedoPos')
-----
1249914
(1 rows)
```

18. 次の文を実行して、以前のステップでデータベースをバックアップしたディレクトリにトランザクションログをバックアップします。

```
BACKUP DATABASE DIRECTORY 'c:¥temp¥backup_dir' TRANSACTION LOG ONLY
TRANSACTION LOG RENAME MATCH;
```

この文によって `c:¥temp¥backup_dir` にトランザクションログファイルの 3 つめのバックアップが作成されます。

19. データベースを切断し、データベースサーバを停止します。

20. `c:¥temp¥backup_dir` を、`c:¥temp¥restore_dir2` (末尾の 2 に注意) という名前の新しいディレクトリにコピーします。

21. `c:¥temp¥restore_dir2¥` で次のコマンドを実行して、`trackingTable.myValue` に値 103 を追加した後書き留めたオフセット (この例では 1249890) へデータベースをリストアします。

```
dbsrv17 demo.db -as -ar -ruo 1249890
```

22. *Interactive SQL* で、`c:¥temp¥restore_dir2` の `demo.db` データベースに接続します。

23. 次の文を実行して、`trackingTable` テーブルのローを表示します。

```
SELECT * FROM trackingTable;
```

#### 結果の例

```
myValue
-----
          1
          103
(2 rows)
```

古いログの値 2 と新しいログの値 104 は含まれません。これは、値 2 が最初のバックアップ操作後に挿入されたため、`c:¥temp¥restore_dir¥` のリストアされたデータベースには存在しないためです。同じように、値 104 は 2 回目のバックアップ操作後に挿入されたため、`c:¥temp¥restore_dir2¥` のリストアされたデータベースには存在しません。

24. ポイントインタイムリカバリが完了したら、データベースをシャットダウンしていずれのバックアップに関連するオプション (たとえば、`-a`、`-ad`、および `-ar`) も付けずに、データベースサーバを再起動します。

## 関連情報

[ポイントインタイムリカバリ \[917 ページ\]](#)

[データベースのリカバリ \[915 ページ\]](#)

[チュートリアル: タイムスタンプへのデータベースのリストア \[920 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

[-ru データベースオプション \[532 ページ\]](#)

[-ruo データベースオプション \[533 ページ\]](#)

### 1.6.1.2.7 イメージバックアップからのリストア

適用するインクリメンタルトランザクションログがないデータベースを、イメージバックアップからリストアします。

## 前提条件

リストアプロセスを開始する前に、破損または損傷しているデータベースファイルやトランザクションログのアーカイブコピーを作成し、リカバリで問題が発生した場合に最初から作業をやり直せるようにします。

## 手順

1. データベースファイルを元のロケーションにコピーします。たとえば、次のコマンドを実行します。

```
copy c:\temp\backup\demo.db C:\Users\Public\Documents\SQL Anywhere 17\Samples
copy c:\temp\backup\demo.log C:\Users\Public\Documents\SQL Anywhere 17\Samples
```

2. データベースサーバを再起動します。

## 結果

データベースがリストアされました。

## 関連情報

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[自動リカバリ処理 \[917 ページ\]](#)

[バックアップの種類 \[896 ページ\]](#)

[コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

### 1.6.1.2.8 アーカイブバックアップからのデータベースのリストア (SQL Central)

SQL Central を使用して、アーカイブバックアップからデータベースをリストアします。

## 前提条件

リストアプロセスを開始する前に、破損または損傷しているデータベースファイルやトランザクションログのアーカイブコピーを作成し、リカバリで問題が発生した場合に最初から作業をやり直せるようにします。

このタスクでは、適用する必要があるトランザクションログのインクリメンタルバックアップがないと仮定しています。

空きページの削除を有効にして強力に暗号化されているデータベースを作成した場合は、データベースをリストアするときに暗号化キーを指定してください。

## 手順

1. 次のコマンドを実行して、パーソナルデータベースサーバを起動します。

```
dbeng17 -n server-name
```

2. SQL Central を起動し、ユーティリティデータベースに接続します。接続 ウィンドウで、次の項目に入力します。
  - a. ユーザ ID 項目に、**DBA** と入力します。
  - b. パスワード項目に、**passwd** と入力します。
  - c. データベース名 フィールドに、**utility\_db** と入力します。
3. **OK** をクリックします。
4. ツール > SQL Anywhere17 > データベースのリストア をクリックします。
5. ウィザードの指示に従います。

## 結果

データベースがリストアされました。

## 関連情報

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

[自動リカバリ処理 \[917 ページ\]](#)

[バックアップの種類 \[896 ページ\]](#)

[コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

### 1.6.1.2.9 アーカイブバックアップからのデータベースのリストア (Interactive SQL)

Interactive SQL を使用して、アーカイブバックアップからデータベースをリストアします。

#### 前提条件

リストアプロセスを開始する前に、破損または損傷しているデータベースファイルやトランザクションログのアーカイブコピーを作成し、リカバリで問題が発生した場合に最初から作業をやり直せるようにします。

このタスクでは、適用する必要があるトランザクションログのインクリメンタルバックアップがないと仮定しています。

空きページの削除を有効にして強かに暗号化されているデータベースを作成した場合は、データベースをリストアするときに暗号化キーを指定してください。

## 手順

1. 次のコマンドを実行して、パーソナルデータベースサーバを起動します。

```
dbeng17 -n server-name
```

2. Interactive SQL を起動して、ユーティリティデータベースに接続します。次のフィールドを完成させます。
  - a. ユーザ ID 項目に、**DBA** と入力します。
  - b. パスワード項目に、**passwd** と入力します。
  - c. データベース名フィールドに、**utility\_db** と入力します。
3. **OK** をクリックします。
4. アーカイブルートを指定して **RESTORE DATABASE** 文を実行します。

この時点で、アーカイブされたデータベースを元のロケーション (デフォルト) にリストアするか、**RENAME** 句を使用して別のコンピュータに別のデバイス名を使用してリストアするかを選択します。

## 結果

データベースがリストアされました。

### 例

次の文では、データベースをファイル `c:\¥backup¥archive.1` のアーカイブバックアップからデータベースファイル `c:\¥newdb¥newdb.db` にリストアします。トランザクションログの名前と場所はデータベース内に指定されます。

```
RESTORE DATABASE 'c:\¥¥newdb¥¥newdb.db'  
FROM 'c:\¥¥backup¥¥archive';
```

次の文では、データベースをファイル `c:\¥backup¥archive.1` のアーカイブバックアップからデータベースファイル `c:\¥newdb¥newdb.db` にリストアします。トランザクションログの名前と場所はデータベース内に指定されます。データベースの暗号化キーが指定されます。

```
RESTORE DATABASE 'c:\¥¥newdb¥¥newdb.db'  
FROM 'c:\¥¥backup¥¥archive'  
KEY '3Km57y1z';
```

## 関連情報

[ユーティリティデータベース \(utility\\_db\) \[296 ページ\]](#)

[自動リカバリ処理 \[917 ページ\]](#)

[バックアップの種類 \[896 ページ\]](#)

[コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

[-gu データベースサーバオプション \[435 ページ\]](#)

## 1.6.1.2.10 ライブバックアップからのデータベースの再開

運用データベースを実行するプライマリコンピュータとは別のコンピュータに作成されたライブバックアップから、データベースを再開します。

### 前提条件

セカンダリサーバに SQL Anywhere がインストールされている必要があります。

BACKUP DATABASE システム権限が必要です。

### 手順

1. フルバックアップトランザクションログファイルとライブバックアップトランザクションログを、データベースファイルのバックアップコピーに適用できる新しいディレクトリにコピーします。
2. 現在のトランザクションログファイル名が予期されるトランザクションログファイル名と一致する場合は、名前を変更します。
3. データベースサーバを `-ad` オプションを指定して起動し、新しいディレクトリにあるトランザクションログを適用し、データベースを最新にします。

```
dbsrv17 "database-name.db" -ad directory-name
```

データベースサーバは、トランザクションログが適用されると自動的に停止します。

4. データベースサーバを通常どおり起動して、ユーザアクセスを許可します。新しいアクティビティは、すべて新しいトランザクションログに書き込まれます。
5. データベースのフルバックアップを実行します。

```
dbbackup -c "connection-string" path
```

6. セカンダリコンピュータにトランザクションログのライブバックアップを開始します。

```
dbbackup -c "connection-string" -l path¥filename.log
```

### 結果

データベースが再起動されました。

## 関連情報

- [ライブバックアップ \(継続バックアップ\) \[899 ページ\]](#)
- [自動リカバリ処理 \[917 ページ\]](#)
- [バックアップの種類 \[896 ページ\]](#)
- [コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

### 1.6.1.2.11 トランザクションログが複数の場合のデータベースのリカバリ

データベースをリカバリする必要があり、複数のトランザクションログがある場合は、正しい順序でトランザクションログファイルをバックアップコピーに適用します。

トランザクションログを正しい順序で適用するには、次のいずれかの方法を使用します。

- `-a` サーバオプションを使用して、各トランザクションログをデータベースのバックアップコピーに個別に適用します。トランザクションログユーティリティ (`dblog`) を使用して、トランザクションログファイルを生成する順番を判断します。このユーティリティは、トランザクションログの開始時のログオフセットが表示されます。これによって複数のログファイルを適用する順序を効果的に判断できます。
- `-ad` サーバオプションを使用して、トランザクションログファイルのロケーションを指定します。データベースサーバは、データベースのバックアップコピーに対するトランザクションログの正しい適用順序をトランザクションログオフセットに基づいて判断します。
- `-ar` サーバオプションを使用して、トランザクションログと同じディレクトリにある、データベースに関連付けられているログファイルを適用することを、データベースサーバに指示します。トランザクションログのディレクトリはデータベースから取得されます。データベースサーバは、データベースのバックアップコピーに対するトランザクションログの正しい適用順序をログオフセットに基づいて判断します。
- 正しい順番を決定後、ログ変換ユーティリティ (`dbtran`) を使用して、1 つ以上のトランザクションログを、データベースのバックアップコピーに適用可能な `.sql` ファイルに変換します。

このセクションの内容:

- [-ad サーバオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ \[935 ページ\]](#)
  - ad サーバオプションを使用して、指定したディレクトリのすべてのトランザクションログをデータベースのバックアップコピーに適用することで、データベースをリカバリします。
- [-a サーバオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ \[936 ページ\]](#)
  - a サーバオプションを使用して、単一のトランザクションログファイルをデータベースのバックアップコピーに適用し、その後データベースサーバをシャットダウンします。

## 関連情報

- [-a データベースオプション \[520 ページ\]](#)
- [-ad データベースオプション \[521 ページ\]](#)
- [-ar データベースオプション \[523 ページ\]](#)
- [トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

## 1.6.1.2.11.1 -ad サーバオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ

-ad サーバオプションを使用して、指定したディレクトリのすべてのトランザクションログをデータベースのバックアップコピーに適用することで、データベースをリカバリします。

### 前提条件

リカバリプロセスを開始する前に、破損または損傷しているデータベースファイルやトランザクションログのバックアップコピーを作成し、リカバリで問題が発生した場合に最初から作業をやり直せるようにします。

### 手順

1. バックアップトランザクションログと現在のトランザクションログをディレクトリにコピーします。
2. データベースサーバを起動し、トランザクションログをデータベースのバックアップコピーに適用します。

```
dbsrv17 backup-copy-of-database -ad directory
```

データベースサーバは、トランザクションログのトランザクションログオフセットを使用して、トランザクションログファイルの正しい適用順序を判断します。

### 結果

データベースサーバは、トランザクションログをデータベースのバックアップコピーに適用した後、停止します。

#### 例

データベースサーバを起動し、c:¥backuplogs ディレクトリのトランザクションログを backupdemo.db という名前のデータベースのバックアップコピーに適用します。

```
dbsrv17 backupdemo.db -ad c:¥backuplogs
```

### 関連情報

[自動リカバリ処理 \[917 ページ\]](#)

[-a サーバオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ \[936 ページ\]](#)

[コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

[-ad データベースオプション \[521 ページ\]](#)

## 1.6.1.2.11.2 -a サーバオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ

-a サーバオプションを使用して、単一のトランザクションログファイルをデータベースのバックアップコピーに適用し、その後データベースサーバをシャットダウンします。

### 前提条件

リカバリプロセスを開始する前に、破損または損傷しているデータベースファイルやトランザクションログのアーカイブコピーを作成し、リカバリで問題が発生した場合に最初から作業をやり直せるようにします。

### コンテキスト

複数のトランザクションログがある場合は、もっとも古いものから最新のものへ、正しい順番で1つずつ適用します。

### 手順

1. データベースサーバを起動し、バックアップトランザクションログをデータベースのバックアップコピーに適用します。

```
dbsrv17 backup-database.db -a backup-transaction-log.log
```

データベースサーバは、バックアップトランザクションログをデータベースのバックアップコピーに適用した後、停止します。

2. データベースサーバを起動し、アクティブトランザクションログをデータベースのバックアップコピーに適用します。

```
dbsrv17 backup-database.db -a current-transaction-log.log
```

データベースサーバは、現在のトランザクションログをデータベースのバックアップコピーに適用した後、停止します。

### 結果

バックアップトランザクションログと、データベースの障害時にアクティブだったトランザクションログを適用することによって、データベースはリストアされ最新の状態になります。

#### 例

1. データベースサーバを起動し、backupdemo.log という名前のバックアップトランザクションログを backupdemo.db という名前のデータベースのバックアップコピーに適用します。

```
dbsrv17 backupdemo.db -a backupdemo.log
```

データベースサーバは、バックアップトランザクションログをデータベースのバックアップコピーに適用した後、停止します。

2. データベースサーバを起動し、`demo.log` という名前の現在のトランザクションログをデータベースのバックアップコピーに適用します。

```
dbsrv17 backupdemo.db -a demo.log
```

データベースサーバは、現在のトランザクションログをデータベースのバックアップコピーに適用した後、停止します。

## 関連情報

[自動リカバリ処理 \[917 ページ\]](#)

[-ad サーバオプションを使用した、複数のトランザクションログがあるデータベースのリカバリ \[935 ページ\]](#)

[コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

[-a データベースオプション \[520 ページ\]](#)

### 1.6.1.2.12 メディア障害からのリカバリ

システム障害とメディア障害の 2 種類の障害によって、データベースが使用不可になることがあります。

システム障害は、トランザクションの途中で、コンピュータまたはオペレーティングシステムが停止した場合に発生します。この種類の障害は、正しい手順でコンピュータを終了または再起動しなかったり、他のアプリケーションによってオペレーティングシステムが停止したり、停電した場合に起こる可能性があります。システム障害が発生した後、次にデータベースを起動するときに、データベースサーバは自動的にリカバリします。システムエラー以前にコミットされたトランザクションの結果は保存されます。システム障害の前にコミットされていなかったトランザクションによる変更はすべてキャンセルされます。

メディア障害はより深刻で、データベースファイルを保存しているデバイスが使用不可となるか、データベースおよびトランザクションログが破損した場合に発生します。

メディア障害からのフルリカバリは、障害が影響するのがデータベースファイルか、またはトランザクションログかに依存します。データベースファイルが使用できないがトランザクションログは使用できる場合、以前のフルバックアップを使用しトランザクションログの変更を適用して、すべてのコミットした変更をリカバリすることができます。

障害が影響するのがトランザクションログで、トランザクションログミラーを使用している場合は、すべてのデータをリカバリすることができます。トランザクションログミラーはないが、トランザクションログをバックアップしている (インクリメンタルバックアップ) 場合、最後のトランザクションログバックアップまですべてのデータをリカバリすることができます。トランザクションログミラーがなく、トランザクションログもバックアップしていない場合、直近のデータベースのチェックポイントから障害発生時までに入力された情報はリカバリできません。ライブバックアップを行っている場合、すべての完了したトランザクションをリカバリすることができます。

データをメディア障害から守るためにできるいくつかの方法を、次に示します。

- 良いバックアッププランを実装します。たとえば、フルバックアップを定期的に行い、その間にインクリメンタルバックアップを行います。
- バックアップをリカバリに使用できるかテストします。
- データベースがある場所とは別のデバイスにバックアップします。

- トランザクションログを失うと、重要な情報の損失やレプリケーションシステムの破損につながる可能性がある場合、設定のトランザクションログミラーを使用します。
- トランザクションログを自身のデータベースがあるのとは異なるデバイスに格納します。トランザクションログを別のデバイスに格納すると、ディスクのヘッドが、トランザクションログとメインデータベースファイルの間を移動しなくて済むことから、パフォーマンスの改善が期待できます。ただし、ネットワークディレクトリの別の場所にトランザクションログを格納しないでください。ネットワークを介してページに対する読み取りと書き込みを行うと、パフォーマンスの低下やファイルの破損を招く可能性があります。
- 2つまたはそれ以上のハードディスクドライブのあるコンピュータの中には、1つの物理ディスクドライブと複数の論理ディスクまたはパーティションを持つものがあります。
- ライブバックアップを維持します。

このセクションの内容:

[データベースファイルのメディア障害からのリカバリ \[938 ページ\]](#)

メディア障害で失われたファイルがデータベースファイルのとき、データベースをリカバリします。

[トランザクションログミラーのメディア障害からのリカバリ \[940 ページ\]](#)

メディア障害からのリカバリにトランザクションログミラーを使用します。

[ミラーされていないトランザクションログのメディア障害からのリカバリ \[941 ページ\]](#)

ミラーされていないトランザクションログを使用して、メディア障害からデータベースを部分的にリカバリします。

## 関連情報

[データベースの作成 \[265 ページ\]](#)

[バックアップの種類 \[896 ページ\]](#)

[ライブバックアップ \(継続バックアップ\) \[899 ページ\]](#)

[トランザクションログの場所の変更 \(SQL Central の場合\) \[281 ページ\]](#)

[トランザクションログの場所の変更 \(コマンドラインの場合\) \[283 ページ\]](#)

### 1.6.1.2.12.1 データベースファイルのメディア障害からのリカバリ

メディア障害で失われたファイルがデータベースファイルのとき、データベースをリカバリします。

## 前提条件

BACKUP DATABASE および VALIDATE ANY OBJECT システム権限が必要です。

## 手順

1. 現在のトランザクションログの追加バックアップコピーを作成します。データベースファイルが使用できないので、最後のバックアップ後に行われた変更の記録は、唯一トランザクションログに含まれています。
2. リカバリの処理中に使用するファイルを保存するリカバリディレクトリを作成します。
3. 最新のフルバックアップからデータベースファイルをリカバリディレクトリにコピーします。
4. バックアップされたトランザクションログに保持されているトランザクションをリカバリデータベースに適用します。次のいずれかを実行します。

オプション	アクション
手動で各トランザクションログを年代順に適用します。	<ol style="list-style-type: none"><li>1. トランザクションログファイルをリカバリディレクトリにコピーします。</li><li>2. トランザクションログ (-a) オプションでデータベースサーバを起動します。<pre>dbsrv17 database-name.db -a log-name.log</pre>データベースサーバは、トランザクションが適用されると自動的に停止します。</li><li>3. トランザクションログのバックアップをすべて適用したら、オンライントランザクションログをリカバリディレクトリにコピーします。 オンライントランザクションログのトランザクションをリカバリデータベースに適用します。<pre>dbsrv17 database-name.db -a log-name.log</pre></li></ol>
データベースサーバにトランザクションの正しい順序を決定させ、自動的に適用させます。	<ol style="list-style-type: none"><li>1. オンラインおよびオフラインのトランザクションログファイルをリカバリディレクトリにコピーします。</li><li>2. -ad オプションでデータベースサーバを起動し、トランザクションログの場所を指定します。データベースサーバは、トランザクションログの正しい適用順序をログオフセットに基づいて判断します。<pre>dbsrv17 database-name.db -ad log-directory</pre>データベースサーバは、トランザクションが適用されると自動的に停止します。</li></ol>

5. リカバリに使用する予定のデータベース上で妥当性検査を行います。
6. バックアップを作成します。
7. データベースファイルを運用ディレクトリに移します。
8. 運用データベースにアクセスできることをユーザに通知します。

## 結果

データベースがリカバリされます。

## 関連情報

[自動リカバリ処理 \[917 ページ\]](#)

[データベースの検証 \(SQL Central\) \[944 ページ\]](#)

[コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

## 1.6.1.2.12.2 トランザクションログミラーのメディア障害からのリカバリ

メディア障害からのリカバリにトランザクションログミラーを使用します。

### コンテキスト

データベースが SQL Remote インストール環境の統合データベースである場合、トランザクションログミラーまたは同様の機能を持つハードウェアを使用してください。

### 手順

1. トランザクションログが開始された時点におけるデータベースファイルの追加バックアップコピーを作成します。
2. どちらのファイルが破損しているのか特定します。トランザクションログとそのミラーに対してログ変換ユーティリティ (dbtran) を実行します。エラーメッセージが生成されるファイルが破損しています。

次のコマンドは `demo.log` という名前のトランザクションログを変換して、変換された出力を `demo.sql` に格納します。

```
dbtran demo.log
```

3. 正しいファイルをコピーして、破損しているファイルに上書きします。
4. データベースサーバを再起動します。

### 結果

データベースがリカバリされます。

### 関連情報

[トランザクションログミラー \[288 ページ\]](#)

[自動リカバリ処理 \[917 ページ\]](#)

[コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

## 1.6.1.2.12.3 ミラーされていないトランザクションログのメディア障害からのリカバリ

ミラーされていないトランザクションログを使用して、メディア障害からデータベースを部分的にリカバリします。

### コンテキスト

データベースが Mobile Link や SQL Remote インストール環境の統合データベースである場合、トランザクションログミラーまたは同様の機能を持つハードウェアを使用してください。

#### 警告

dbeng17 コマンドは、データベースが Mobile Link や SQL Remote システムに関連していない場合にだけ使用してください。データベースが SQL Remote レプリケーションシステム内の統合データベースである場合は、リモートデータベースを再抽出する必要があります。

### 手順

1. データベースファイルの追加バックアップコピーを作成します。トランザクションログがない場合、最後にバックアップが行われてから最新のチェックポイントまでの間に加えられた変更の記録は、唯一データベースファイルに含まれています。
2. トランザクションログファイルを削除するか、名前を変更します。
3. -f オプションを使って、データベースを再起動します。

```
dbsrv17 "database-name.db" -f
```

-f オプションを指定しないと、データベースサーバはトランザクションログがないことを知らせるエラーメッセージを表示します。-f オプションを指定すると、データベースサーバは最新のチェックポイント時の状態にデータベースをリストアし、チェックポイントの時点でコミットされていなかったトランザクションをすべてロールバックします。その後新しいトランザクションログが作成されます。

### 結果

データベースが部分的にリカバリされます。最新のチェックポイントでコミットされていないトランザクションは、ロールバックされます。

### 関連情報

[トランザクションログミラー \[288 ページ\]](#)

[トランザクションログ \[279 ページ\]](#)

[自動リカバリ処理 \[917 ページ\]](#)

[コミットされていない操作のリカバリ \(SQL Central\) \[918 ページ\]](#)

[i データベースサーバオプション \[408 ページ\]](#)

### 1.6.1.3 タイムライン

タイムラインはデータベースに対して行われる変更の順番を確立します。

タイムラインは、データベースファイルとログファイルのセットを識別して順番に適用する必要がある、複雑なデータベースのリストアで役に立ちます。タイムラインを正しく使うと、トランザクションログ中に重複するオフセットが存在しないことを保証できます。

タイムラインは変換ログユーティリティ (dblog) およびログ変換ユーティリティ (dbtran) の出力内に存在します。タイムラインのフォーマットは "GUIDtimestamp" で、GUID はタイムラインのユニークな識別子、timestamp はタイムラインが作成された時刻で、UTC 表記です。

タイムラインが生成されるときには、注目すべき時刻が 3 つあります。

- データベースが作成されたとき
- データベースのアップロードまたはダウンロードを実行したとき
- データベースでポイントインタイムリカバリを実行したとき

ポイントインリカバリでタイムラインが生成されたとき、データベースサーバは元のデータベースのタイムラインを記憶しており、以降のポイントタイムリカバリで再度リストアして元のタイムラインの情報を使用することがあります。

## 関連情報

[ポイントインタイムリカバリ \[917 ページ\]](#)

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

## 1.6.2 データベースの検証

データベースファイルの破損は、データベースサーバがデータベース内の破損部分にアクセスするまで判明しないことがあります。データベースが有効であることを定期的にチェックします。

オプションの指定によって、チェックサム、インデックスデータの正当性、すべてのテーブルページがデータベース内のオブジェクトに属するかどうかといった検証内容が異なります。パフォーマンスを飛躍的に向上するため、参照整合性チェックを無効にするエクスプレス検証を実行することもできます。

## データベースの検証の推奨内容

検証中、データベースに変更が行われないようにします。たとえば、データベースオプション `-r` を使用して、データベースを読み込み専用モードで起動するか、データベースサーバオプション `-im` でインメモリ検証モードを使用します。

検証が失敗した場合、メディア障害として対処し、データベースを以前のバックアップからリカバリします。

## テーブルの検証の推奨内容

デフォルトのオプションを使用して、テーブルを検証します。これがもっとも速いテーブルの検証方法です。テーブルが破損したことを示すエラーを受信した場合は、排他データロックまたはスナップショットアイソレーションで検証を再実行します。この検証が失敗した場合、メディア障害として対処し、データベースを以前のバックアップからリカバリします。

検証では、検証対象のテーブルに対する排他アクセスを取得しないため、検証中にテーブルが変更された場合には、誤って破損をレポートすることがあります。テーブルが破損したことを示す検証エラーを受信した場合は、排他データロックまたはスナップショットアイソレーションで検証を再実行します。たとえば、`VALIDATE` 文を `WITH DATA LOCK` 句または `WITH SNAPSHOT` 句とともに実行します。この検証が成功すると、元の破損エラーは `false` になります。この検証が失敗した場合はメディア障害として対処し、以前のバックアップからリカバリします。

## データベースバックアップの検証の推奨内容

データベースバックアップを検証するには、最初にインメモリ検証モードを使用してデータベースバックアップを開始してから、検証を実行します。インメモリ検証モードでは書き込みができず、トランザクションからリカバリが可能かを確認します。インメモリ検証モードでは、`-r` データベースサーバオプションとは異なり、保留中のトランザクションがあるデータベースに対応することができます。たとえば、`-im` データベースサーバオプションでインメモリ検証モードを使用します。

## インデックス検証の推奨内容

インデックスが破損している場合は、インデックスなしでデータベースをアンロードしてから、データベースを再ロードします。また、`ALTER INDEX` 文の `REBUILD` 句を使用してインデックス破損を修正することもできます。

- エラーがレポートされた場合は、テーブル上のすべてのインデックスとキーを一度削除してから再作成できます。テーブルに対する外部キーも再作成する必要があります。
- 疑わしいインデックスがある場合は、`ALTER INDEX...REBUILD` 文を実行して、破損したインデックスを再構築できます。
- `VALIDATE TABLE` 文によってレポートされるエラーを解決するには、データベース全体をいったんアンロードし、再ロードする方法もあります。アンロード処理で、破損した可能性のあるインデックスがデータの順序付けに使用されないように、`dbunload` の `-u` オプションを使用してください。

このセクションの内容:

[データベースの検証 \(SQL Central\) \[944 ページ\]](#)

SQL Central を使用してデータベースを検証します。

#### [データベースの検証 \(SQL の場合\) \[945 ページ\]](#)

VALIDATE 文または sa\_validate システムプロシージャを使用して、データベースのバックアップを含むデータベースを検証します。

#### [データベースの検証 \(dbvalid の場合\) \[946 ページ\]](#)

データベース検証ユーティリティ (dbvalid) を使用してデータベースを検証します。

#### [チェックサムを使用した破損の検出 \[947 ページ\]](#)

チェックサムは、データベースページがディスク上で変更されたかどうかを判断するために使用します。

## 関連情報

[トランザクションログの検証 \[281 ページ\]](#)

[テーブルの検証 \(SQL Central\) \[949 ページ\]](#)

[検証ユーティリティ \(dbvalid\) \[1200 ページ\]](#)

[-r データベースサーバオプション \[467 ページ\]](#)

## 1.6.2.1 データベースの検証 (SQL Central)

SQL Central を使用してデータベースを検証します。

### 前提条件

VALIDATE ANY OBJECT システム権限が必要です。

### 手順

1. 検証中、データベースに変更が行われないようにします。たとえば、データベースオプション -r を使用して、データベースを読み込み専用モードで起動するか、データベースサーバオプション -im でインメモリ検証モードを使用します。
2. *SQL Anywhere 17* プラグインを使用してデータベースに接続します。
3. 左ウィンドウ枠でデータベースを選択します。
4. ファイルメニューで、データベースの同期をクリックします。
5. データベース検証ウィザードの指示に従います。

### 結果

データベースが検証されます。

データベースの検証がテーブルの破損以外の理由で失敗した場合、メディア障害として対処し、データベースを以前のバックアップからリカバリします。

テーブルが破損したことを示す検証エラーを受信した場合は、[データベース検証ウィザード](#)を再実行し、[排他データロックを使用オプション](#)、または[スナップショットアイソレーションを使用オプション](#)を選択します。この 2 番目の検証が成功すると、元の破損エラーは false になります。この検証が失敗した場合、メディア障害として対処し、以前のバックアップからリカバリします。

## 関連情報

[検証ユーティリティ \(dbvalid\) \[1200 ページ\]](#)

[-r データベースサーバオプション \[467 ページ\]](#)

### 1.6.2.2 データベースの検証 (SQL の場合)

VALIDATE 文または sa\_validate システムプロシージャを使用して、データベースのバックアップを含むデータベースを検証します。

## 前提条件

VALIDATE ANY OBJECT システム権限が必要です。

## 手順

1. 検証中に同時トランザクションが起こらないように、データベースを起動します。次のオプションのうちの 1 つを選択してください。

オプション	アクション
データベースを検証するには、次の手順に従います	データベースオプション <code>-r</code> を使用して、データベースを読み込み専用モードで起動するか、データベースサーバオプション <code>-im</code> でインメモリ検証モードを使用します。
データベースのバックアップを検証するには、次の手順に従います	インメモリ検証モードを使用して、データベースのバックアップを開始します。 インメモリ検証モードは書き込みを防ぎ、トランザクションがリカバリできることを確認し、保留中のトランザクションがあるデータベースを処理することができます。

2. データベースに接続して、次のうち 1 つのオプションを選択します。

オプション	アクション
sa_validate ストアドプロシージャ	<p>sa_validate ストアドプロシージャを実行します。</p> <pre>CALL sa_validate;</pre> <p>このストアドプロシージャは Messages という 1 つのカラムを返します。すべてのテーブルが有効である場合、カラムには No errors detected と表示されます。</p>
VALIDATE DATABASE 文	VALIDATE DATABASE 文を実行します。

## 結果

データベースが検証されます。

データベースの検証がテーブルの破損以外の理由で失敗した場合、メディア障害として対処し、データベースを以前のバックアップからリカバリします。

テーブルが破損したことを示す検証エラーを受信した場合は、排他データロックまたはスナップショットアイソレーションで検証を再実行します。たとえば、VALIDATE 文を USE DATA LOCK 句または USE SNAPSHOT 句とともに指定します。この 2 番目の検証が成功すると、元の破損エラーは false になります。この検証が失敗した場合、メディア障害として対処し、以前のバックアップからリカバリします。

## 関連情報

[検証ユーティリティ \(dbvalid\) \[1200 ページ\]](#)

[r データベースサーバオプション \[467 ページ\]](#)

### 1.6.2.3 データベースの検証 (dbvalid の場合)

データベース検証ユーティリティ (dbvalid) を使用してデータベースを検証します。

## 前提条件

VALIDATE ANY OBJECT システム権限が必要です。

## 手順

dbvalid ユーティリティを実行し、-ws または -wl オプションを指定して、検証中データベースに変更が加えられないようにします。-ws オプションにより、テーブルにスナップショットアイソレーションが適用され、-wl オプションによりテーブルに排他ロックが適用されます。次に例を示します。

```
dbvalid -c "connection-string-ws"
```

```
dbvalid -c "connection-string-wl"
```

## 結果

データベースが検証されます。

検証が失敗した場合、メディア障害として対処し、データベースを以前のバックアップからリカバリします。

## 関連情報

[検証ユーティリティ \(dbvalid\) \[1200 ページ\]](#)

[-r データベースサーバオプション \[467 ページ\]](#)

### 1.6.2.4 チェックサムを使用した破損の検出

チェックサムは、データベースページがディスク上で変更されたかどうかを判断するために使用します。

2つのチェックサムの種類がサポートされています。グローバルチェックサムと書き込みチェックサムです。

#### グローバルチェックサム

グローバルチェックサムは、データベースの作成時に有効にします。グローバルチェックサムを有効にしてデータベースを作成した場合、チェックサムはページがディスクに書き込まれる直前に計算されます。そのページが次にディスクから読み出されるときに、ページのチェックサムが再計算されて、ページに保存されているチェックサムと比較されます。チェックサムが異なる場合は、ディスク上でページが変更されており、エラーが発生します。

次の文を実行することによって、データベースがグローバルチェックサムを有効にして作成されたかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'Checksum' );
```

グローバルチェックサムが有効な場合、このクエリは ON を返します。それ以外の場合は OFF を返します。

ALTER DATABASE 文の CHECKSUM 句を使用すると、既存のデータベースのグローバルチェックサムを無効にすることができます。チェックサムを無効にすることはお奨めしません。グローバルチェックサムを一度無効にすると、有効にすることが

できなくなります。-wc+ オプションを使用してデータベースの書き込みチェックサムを有効にするか、またはチェックサムを有効にしてデータベースを再構築することができます。

## 書き込みチェックサム

デフォルトでは、バージョン 10 または 11 の SQL Anywhere で作成されたデータベースではグローバルチェックサムは有効になっていません。バージョン 12 のデータベースサーバで SQL Anywhere 11 を使用して作成されたデータベースを起動すると、デフォルトでは、データベースサーバで書き込みチェックサムが作成されます。書き込みチェックサムは、ページがディスクに書き込まれるときにのみページに追加されるチェックサムです。

バージョン 12 以降のデータベースでは、デフォルトでグローバルチェックサムが有効になっています。新しいデータベースを作成し、そのデータベースのグローバルチェックサムを無効にした場合は、-wc+ オプションまたは START DATABASE 文を使用して書き込みチェックサムを有効にすることができます。

次の文を実行することによって、データベースが書き込みチェックサムを有効にして起動されたかどうかをチェックできます。

```
SELECT DB_PROPERTY ( 'WriteChecksum' );
```

このクエリは、ページがディスクに書き込まれるときにのみチェックサムが有効になっている場合は ON を返し (グローバルチェックサムまたは書き込みチェックサムが有効になっているため)、それ以外の場合は OFF を返します。

## 書き込みチェックサムの自動作成

次のような場合には、データベース作成時に指定したグローバルチェックサム設定に関係なく、データベースの書き込みチェックサムが有効になります。

### 重要なページ

グローバルチェックサムが有効かどうかに関係なく、データベースサーバはすべてのデータベース内の重要なデータベースページの書き込みチェックサムを計算します。計算されたチェックサムは、オフラインの破損を検出するために使用します。チェックサムによって、重要なページの破損が原因で他のデータが破損するのを防ぐことができます。データベースサーバがチェックサムを計算するので、チェックサムが有効に指定されていないデータベースが壊れた場合、データベースサーバは致命的なエラーとともに停止します。

また、グローバルチェックサムが有効に指定されていないデータベースを検証した場合でも、そのデータベースに重要なページの破損があると、dbvalid はチェックサム検証に関する警告を返します。

### 一部のストレージメディアで動作しているデータベース

信頼性の低いストレージメディア (ネットワークドライブやリムーバブルドライブなど) 上でデータベースが実行されている場合、そのデータベースに対する書き込みチェックサムが自動的に有効になります。データベースがそのようなデバイス上にある間は書き込みチェックサムが有効なままになり、ページが書き込まれるたびに、ページにチェックサムが追加されます。信頼性の高いストレージデバイスにデータベースを移動すると、チェックサムが追加されたページがデータベースサーバのキャッシュに入ったときに、そのページに対してチェックサムの検証が実行されます。

## チェックサムの検証

グローバルチェックサムまたは書き込みチェックサムを使用するデータベースのディスクページを検証できます。データベースでグローバルチェックサムが有効になっている場合は、すべてのデータベースページが検証されます。データベースで書き込みチェックサムのみが使用されている場合は、チェックサムを含むページのみが検証されます。

チェックサムを有効にしたデータベースの場合、チェックサムは各データベースページで計算され、ページがディスクに書き込まれる際にその値が格納されます。検証ユーティリティ (dbvalid)、VALIDATE 文、sa\_validate システムプロシージャ、または SQL Central の [データベース検証ウィザード](#) を使用してチェックサムの検証を実行できます。この検証では、ディスクからのデータベースページの読み込みとそのページに対するチェックサムの計算が行われます。計算されたチェックサムが格納されているページのチェックサムと一致しない場合、ディスク上で、または書き込み中に、そのページが変更または破壊されています。1 つまたは複数のページが破壊されている場合、エラーが返されて無効なページに関する情報がデータベースサーバメッセージウィンドウに表示されます。

このセクションの内容:

[テーブルの検証 \(SQL Central\) \[949 ページ\]](#)

SQL Central を使用してテーブルを検証します。

## 関連情報

[-wc データベースサーバオプション \[499 ページ\]](#)

[-wc データベースオプション \[536 ページ\]](#)

[検証ユーティリティ \(dbvalid\) \[1200 ページ\]](#)

### 1.6.2.4.1 テーブルの検証 (SQL Central)

SQL Central を使用してテーブルを検証します。

## 前提条件

VALIDATE ANY OBJECT システム権限が必要です。検証は、テーブル上に他のアクティビティがないときに行うことをおすすめします。

## 手順

1. [SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[テーブル](#)をダブルクリックします。

オプション	アクション
特定のテーブルを検証します。	<ol style="list-style-type: none"> <li>1. 左ウィンドウ枠で、<b>テーブル</b>をダブルクリックします。</li> <li>2. テーブルを右クリックし、<b>検証</b>をクリックします。</li> <li>3. <b>OK</b>をクリックします。</li> </ol>
すべてのテーブルおよびマテリアライズドビューの検証	<ol style="list-style-type: none"> <li>1. 左ウィンドウ枠でデータベースを選択します。</li> <li>2. <b>ファイルメニュー</b>で、<b>データベースの同期</b>をクリックします。</li> <li>3. データベースを選択して <b>次へ</b> をクリックします。</li> <li>4. <b>テーブルおよびマテリアライズドビューの検証</b>オプションを有効にして、ウィザードの指示に従います。</li> </ol>

## 結果

テーブルが検証されます。

この検証が失敗した場合はメディア障害として対処し、以前のバックアップからリカバリします。

## 関連情報

[検証ユーティリティ \(dbvalid\) \[1200 ページ\]](#)

## 1.6.3 スケジュールとイベントの使用によるタスクの自動化

スケジュールとイベントによるデータベース管理タスクの自動化

### タスクの自動化

データベースにイベントを追加し、イベントのスケジュールを設定することによってルーチンタスクを自動化します。スケジュールに設定されている時刻になると、いつでもイベントハンドラと呼ばれる一連のアクションがデータベースサーバによって実行されます。

バックアップをスケジュールするために、メンテナンスプランを使用します。

### タスクのトリガ

また、データベース管理では、ある状態が発生したときにアクションを実行することも必要です。たとえば、トランザクションログが格納されているディスクの空き領域が少なくなってきたときには、適切な処置を行うよう、システム管理者に電子メールで通知することが考えられます。これらのタスクも各システムイベントに対してイベントハンドラを定義することによって自動化できます。

このセクションの内容:

[イベント \[951 ページ\]](#)

データベースにイベントを追加し、イベントのスケジュールを設定することによってルーチンタスクを自動化できます。

[スケジュール \[952 ページ\]](#)

アクティビティをスケジュールすると、事前に設定した時刻に一連のアクションを確実に実行できます。

[システムイベント \[954 ページ\]](#)

各システムイベントには一連のアクションをプログラムするためのフックがあります。

[イベントハンドラ \[957 ページ\]](#)

イベントハンドラは、イベントをトリガしたアクションとは別の接続を使って動作し、クライアントアプリケーションとの対話は行いません。

## 関連情報

[データベースサーバによるスケジュールされたイベントのチェック \[960 ページ\]](#)

[イベントハンドラの実行 \[960 ページ\]](#)

[イベントのトリガ条件 \[956 ページ\]](#)

[イベントハンドラのデバッグ \[963 ページ\]](#)

### 1.6.3.1 イベント

データベースにイベントを追加し、イベントのスケジュールを設定することによってルーチンタスクを自動化できます。

次のイベントのタイプがサポートされています。

スケジュールされたイベント

スケジュールに関連付けられており、指定の時間に実行されます。

システムイベント

データベースサーバによって追跡される特定のタイプの条件に関連付けられています。

手動イベント

TRIGGER EVENT 文を使用して明示的に起動されます。

ユーザトレースイベント

アプリケーションに関する情報をイベントトレースセッションに記録するのに使用されます。これらのイベントは、データベースへの接続のすべてから参照できます。

各イベントハンドラの実行後、エラーが発生しない場合、COMMIT が発生します。エラーが発生した場合は、ROLLBACK が発生します。

## 関連情報

[スケジュール \[952 ページ\]](#)

[システムイベント \[954 ページ\]](#)

[イベントのトレース \[964 ページ\]](#)

[イベントハンドラのトリガ \[962 ページ\]](#)

### 1.6.3.2 スケジュール

アクティビティをスケジュールすると、事前に設定した時刻に一連のアクションを確実に実行できます。

スケジュール情報とイベントハンドラはいずれもデータベース自体に格納されます。

通常は必要ありませんが、1つの名前付きイベントに2つ以上のスケジュールを関連付けることによって複雑なスケジュールを定義できます。たとえば、営業時間が曜日によって変わるような販売店で、営業時間中1時間に1回イベントを発生させることができます。それぞれ別のスケジュールを持つ複数のイベントを定義して、共通のストアードプロシージャを呼び出すことで同じような効果が得られます。

イベントをスケジュールするとき、英語の曜日をフルネーム (Monday、Tuesday など) で使用することも、省略形 (Mon、Tue など) で使用することもできます。英語以外の言語で稼働するサーバで曜日名を認識する必要がある場合は、フルネームの英語の曜日を使用する必要があります。

次に便利なスケジュールの例を示します。

#### 例

インクリメンタルバックアップを毎日午前1時に実行します。

```
CREATE EVENT IncrementalBackup
SCHEDULE
  START TIME '1:00 AM' EVERY 24 HOURS
HANDLER
BEGIN
  BACKUP DATABASE DIRECTORY 'c:¥¥backup'
  TRANSACTION LOG ONLY
  TRANSACTION LOG RENAME MATCH
END;
```

終業時に受注の要約を作成します。

```
CREATE EVENT Summarize
SCHEDULE
  START TIME '6:00 pm'
  ON ( 'Monday', 'Tuesday', 'Wednesday', 'Thursday',
      'Friday' )
HANDLER
BEGIN
  INSERT INTO OrderSummary
  SELECT CURRENT DATE,
         COUNT( * ),
         SUM( amount )
  FROM Orders
  WHERE date_ordered = current date
END;
```

このセクションの内容:

[スケジュール定義 \[953 ページ\]](#)

柔軟に設定を可能にするために、スケジュール定義にはいくつかの構成要素が用意されています。

### 1.6.3.2.1 スケジュール定義

柔軟に設定を可能にするために、スケジュール定義にはいくつかの構成要素が用意されています。

#### 名前

各スケジュール定義には名前があります。2 つ以上のスケジュールを 1 つのイベントに割り当てることができます。これは複雑なスケジュールを設計するのに便利です。

#### 開始時刻

イベントの起動時刻を定義できます。これは、イベントの実行が開始する時刻です。

#### 範囲

起動時刻の代わりに、イベントがアクティブになる時刻の範囲を指定できます。イベントは、指定した起動時刻と終了時刻の間に発生します。頻度は指定した周期で決定します。

#### 周期

各スケジュールには周期を定義できます。イベントは、何日または何曜日ごとの何時何分何秒ごとという形で指定できる頻度でトリガされます。反復するイベントには、EVERY または ON 句が含まれています。

このセクションの内容:

[イベントのスケジュールの作成 \[953 ページ\]](#)

あらかじめ設定された時間にアクションが実行されるように、イベントのスケジュールを定義します。

#### 1.6.3.2.1.1 イベントのスケジュールの作成

あらかじめ設定された時間にアクションが実行されるように、イベントのスケジュールを定義します。

#### 前提条件

ALTER ANY OBJECT または MANAGE ANY EVENT のどちらかのシステム権限が必要です。

#### 手順

1. [SQL Anywhere 17 プラグイン](#)を使用してデータベースに接続します。

2. 左ウィンドウ枠で、**イベント** をダブルクリックします。
3. 右ウィンドウ枠で、スケジュールを作成するイベントをダブルクリックします。
4. **スケジュール** タブをクリックします。
5. **ファイル** > **新規** > **スケジュール** をクリックします。
6. **スケジュール作成ウィザード** の指示に従います。

## 結果

イベントに対してスケジュールが作成されます。

### 1.6.3.3 システムイベント

各システムイベントには一連のアクションをプログラムするためのフックがあります。

データベースサーバはイベントを追跡し、定義されたトリガ条件をシステムイベントが満たしたときに (イベントハンドラに定義された) アクションを実行します。

イベントハンドラを定義し、選択したシステムイベントタイプが発生して、定義したトリガ条件を満たしたときにイベントハンドラが実行されるようにします。このようにすると、データのセキュリティと安全性が向上し、管理が容易になります。イベントハンドラの動作は、実行中にエラーが検出されなければコミットされ、エラーが検出された場合はロールバックされます。

このセクションの内容:

[システムイベントタイプ \[954 ページ\]](#)

システムイベントにはいくつかのタイプがあります。

[イベントのトリガ条件 \[956 ページ\]](#)

各イベント定義には、関連するシステムイベントおよび 1 つ以上のトリガ条件があります。

#### 1.6.3.3.1 システムイベントタイプ

システムイベントにはいくつかのタイプがあります。

**監査イベント**

**BackupEnd**

BackupEnd イベントタイプを使用すると、バックアップ終了時にアクションを実行できます。

**接続イベント**

接続が確立されたとき (Connect) または接続できなかったとき (ConnectFailed)。これらのイベントはセキュリティの目的で使用できます。イベントハンドラに接続する代わりに、ログインプロシージャを使用することもできます。

**DatabaseStart**

DatabaseStart イベントタイプを使用すると、データベース起動時にアクションを実行できます。

## Deadlock

Deadlock イベントを使用すると、デッドロック発生時にアクションを実行できます。イベントハンドラでは、sa\_report\_deadlocks プロシージャを使用して、デッドロックが発生するに至った状況に関する情報を取得できます。Deadlock イベントを使用するときは、データベースサーバがデッドロック情報を取得するように設定する必要があります。これを行うには、log\_deadlocks オプションを On にし、sa\_server\_option または -zl サーバオプションを使用して RememberLastStatement 機能を有効にします。

Deadlock イベントは、接続デッドロックとスレッドデッドロックの発生時に起動します。デッドロックイベントは、sa\_report\_deadlocks システムプロシージャによって取得できるもの以上の情報は提供しません。しかし、このイベントを使用すると、デッドロックにすぐに対処できます。データベースサーバに保持されるデッドロック関連の情報は量が限られているため、迅速な対応が重要なこともあります。

### 切断

Disconnect イベントを使用すると、ユーザまたはアプリケーションの切断時にアクションを実行できます。

### ディスクの空き領域

データベースファイル (DBDiskSpace)、トランザクションログファイル (LogDiskSpace)、テンポラリファイル (TempDiskSpace) を格納しているデバイスの使用可能なディスク領域を追跡します。

ディスク領域イベントを使用すると、ディスク領域が不足したときに管理者に警告することができます。

データベースサーバの起動時に -fc オプションを指定して、データベースサーバでファイルシステムがいっぱいになった場合のコールバック関数を実装できます。

### ファイルサイズ

ファイルが指定したサイズに達したとき。これはデータベースファイル (GrowDB)、トランザクションログ (GrowLog)、テンポラリファイル (GrowTemp) に使用できます。

ファイルサイズイベントを使用すると、データベース上での異常なアクションを追跡したり、バルクオペレーションをモニターすることができます。

## GlobalAutoincrement

GLOBAL AUTOINCREMENT で定義されたカラムの残りの値がこの範囲の 1% を下回ると、GlobalAutoincrement イベントが起動します。これは、このイベント用のパラメータとして指定された残りの値のテーブルと数字に基づいて、global\_database\_id オプション用の新しい値を要求するのに使用できます。このイベントにおけるテーブルの残りの値を取得するには、EVENT\_PARAMETER 関数を使用し、RemainingValues パラメータと TableName パラメータを指定します。RemainingValues は、そのカラム用に生成できる残りの値の数を返します。TableName は、範囲の終わりに近づいている GLOBAL AUTOINCREMENT カラムがあるテーブルを返します。

## RAISERROR エラー

RAISERROR イベントタイプを使用すると、RAISERROR 文が実行されたときにアクションを実行できます。RAISERROR 文で使用するエラー番号は、EVENT\_CONDITION 関数 (たとえば、EVENT\_CONDITION( 'ErrorNumber' )) を使用してイベントハンドラ内に定義できます。

### アイドル時間

データベースサーバが指定した時間アイドル状態にあったとき (ServerIdle)。このイベントタイプを使用すると、定型の管理操作をアクセスの少ない時間に行えます。

### データベースミラーリングシステムのイベントタイプ

次のシステムイベントがデータベースミラーリングでサポートされています。

## MirrorFailover

このイベントは、ミラーリングされたデータベースの所有権をデータベースサーバが取得するたびに発生します。たとえば、初めてサーバが起動され、データベースを所有する必要があると判断した場合に起動します。また、前回ミラーとして動作していたサーバが、プライマリサーバがダウンしており、監視サーバに確認した結果、所有権を取得する必要があると判断した場合にも起動します。

#### MirrorServerDisconnect

プライマリサーバとミラーサーバ間の接続が失われるか、プライマリサーバと監視サーバ間の接続が失われると、MirrorServerDisconnect イベントが発生します。このイベントのハンドラ内部では、EVENT\_PARAMETER('MirrorServerName') の値は、接続が失われたサーバの名前です。このイベントはプライマリサーバでのみ発生します。

## 関連情報

[login\\_procedure オプション \[720 ページ\]](#)

[log\\_deadlocks オプション \[716 ページ\]](#)

[-fc データベースサーバオプション \[409 ページ\]](#)

### 1.6.3.3.2 イベントのトリガ条件

各イベント定義には、関連するシステムイベントおよび1つ以上のトリガ条件があります。

システムイベントに対するトリガ条件が満たされるとイベントハンドラがトリガされます。

トリガ条件は CREATE EVENT 文の WHERE 句に含まれていて、AND キーワードを使用して結合できます。各トリガ条件は次のフォームで定義します。

```
event_condition( condition-name ) comparison-operator value
```

condition-name 引数は、さまざまなイベントタイプに対応できるようにあらかじめ設定されている文字列から1つを選択します。たとえば、DBSize(メガバイト単位のデータベースファイルサイズ)を使用して GrowDB システムイベントに適したトリガ条件を構築することができます。データベースサーバは、条件名とイベントタイプの対応をチェックしません。イベントタイプのコンテキストで条件に意味があるかどうかを確認する必要があります。

#### 例

- トランザクションログのサイズを 10 MB に制限します。

```
CREATE EVENT LogLimit
TYPE GrowLog
WHERE event_condition( 'LogSize' ) > 10
HANDLER
BEGIN
  IF EVENT_PARAMETER( 'NumActive' ) = 1 THEN
    BACKUP DATABASE
    DIRECTORY 'c:¥¥logs'
    TRANSACTION LOG ONLY
    TRANSACTION LOG RENAME MATCH;
  END IF;
END;
```

- データベースファイルを格納しているデバイスの空き領域が 10 % を下回ると管理者に通知しますが、ハンドラは 5 分間 (300 秒) に 2 回以上実行しません。

```
CREATE EVENT LowDBSpace
TYPE DBDiskSpace
WHERE event_condition( 'DBFreePercent' ) < 10
AND event_condition( 'Interval' ) >= 300
HANDLER
BEGIN
CALL xp_sendmail( recipient='DBAdmin',
subject='Low disk space',
"message"='Database free disk space '
|| EVENT_PARAMETER( 'DBFreeSpace' ) );
END;
```

- データベースに侵入しようとする者を発見すると、管理者に通知します。

```
CREATE EVENT SecurityCheck
TYPE ConnectFailed
HANDLER
BEGIN
DECLARE num_failures INT;
DECLARE mins INT;
INSERT INTO FailedConnections( log_time )
VALUES ( CURRENT_TIMESTAMP );
SELECT COUNT( * ) INTO num_failures
FROM FailedConnections
WHERE log_time >= DATEADD( minute, -5,
current_timestamp );
IF( num_failures >= 3 ) THEN
SELECT DATEDIFF( minute, last_notification,
current_timestamp ) INTO mins
FROM Notification;
IF( mins > 30 ) THEN
UPDATE Notification
SET last_notification = current_timestamp;
CALL xp_sendmail( recipient='DBAdmin',
subject='Security Check', "message"=
'over 3 failed connections in last 5 minutes' )
END IF
END IF
END;
```

- サーバが 10 分間以上アイドル状態にあると、処理を実行します。1 時間に 2 回以上は実行しません。

```
CREATE EVENT Soak
TYPE ServerIdle
WHERE event_condition( 'IdleTime' ) >= 600
AND event_condition( 'Interval' ) >= 3600
HANDLER
BEGIN
MESSAGE ' Insert your code here ... '
END;
```

### 1.6.3.4 イベントハンドラ

イベントハンドラは、イベントをトリガしたアクションとは別の接続を使って動作し、クライアントアプリケーションとの対話はいりません。

イベントハンドラは、イベントの作成者の権限で実行されます。

イベントハンドラは、スケジュールされたイベント用か、システムイベント処理用かにかかわらず、複合文を含んでいて、多くの点でストアドプロシージャに似ています。ループや条件付き実行などを追加することができます。

SQL Anywhere デバッガを使用してイベントハンドラのデバッグすることもできます。

各イベントハンドラの実行後、エラーが発生しない場合、COMMIT が発生します。エラーが発生した場合は、ROLLBACK が発生します。

## イベントハンドラのためのコンテキスト情報

ストアドプロシージャとは異なり、イベントハンドラには引数がありません。EVENT\_PARAMETER 関数を使用して、イベントがトリガされたコンテキストに関する情報にアクセスできます。返される情報には、イベントがトリガされた接続 ID とユーザ ID、イベント名、実行回数が含まれます。

## イベントハンドラのテスト

開発中は、好きなときにイベントハンドラをトリガできた方が便利です。TRIGGER EVENT 文を使うと、トリガ条件やスケジュールした時刻に関係なく、明示的にイベントを実行できます。ただし、無効なイベントハンドラを TRIGGER EVENT によって実行することはできません。

運用データベース上でイベントハンドラを開発するのはよいことではありませんが、明示的に ALTER EVENT 文を使ってイベントハンドラを無効にすることができます。

## コードの共有

複数のイベントを処理するアクションを 1 つにまとめておくと便利です。たとえば、データベースファイルまたはログファイルを格納しているデバイスのディスク領域が少なくなってきたときに、通知アクションを実行することができます。これを実行するには、ストアドプロシージャを作成し、各イベントハンドラの本文から呼び出します。このとき、必要なコンテキスト情報をパラメータとしてプロシージャに渡します。

## イベントハンドラのデバッグ

イベントハンドラのデバッグは、ストアドプロシージャのデバッグによく似ています。イベントハンドラは、イベントリストに表示されます。

## イベントハンドラを隠す

ALTER EVENT 文と SET HIDDEN 句を使用して、イベントハンドラの定義を隠すことができます。SET HIDDEN 句を指定すると、ISYSEVENT システムテーブルの action カラムに格納されているイベントハンドラの定義が永続的に難読化されます。

## アクティブなイベントの制限

また、EVENT\_PARAMETER 関数と NumActive コンテキスト名を使用して、現在アクティブになっている特定のイベントハンドラのインスタンスの数を判断できます。この関数は、一定時間に 1 つのイベントハンドラで 1 つのインスタンスだけを実行させるように制限する場合に利用できます。

このセクションの内容:

### データベースサーバによるシステムイベントのチェック [959 ページ]

システムイベントは、CREATE EVENT 文の中で指定されるイベントタイプによって分類されます。

### データベースサーバによるスケジュールされたイベントのチェック [960 ページ]

イベントのスケジュール時刻の計算は、データベースサーバの起動時と、スケジュールされた各イベントハンドラの完了時に行われます。

### イベントハンドラの実行 [960 ページ]

イベントハンドラがトリガされると、一時的に内部接続が確立され、その上でイベントハンドラが実行されます。

### データベースイベントの作成 [961 ページ]

イベントを作成してルーチンタスクを自動化します。

### イベントハンドラのトリガ [962 ページ]

イベントをトリガします。

### イベントハンドラのデバッグ [963 ページ]

イベントハンドラをデバッグします。

### イベントハンドラを隠す [963 ページ]

ALTER EVENT 文を使用してイベントハンドラの定義を隠します。

## 1.6.3.4.1 データベースサーバによるシステムイベントのチェック

システムイベントは、CREATE EVENT 文の中で指定されるイベントタイプによって分類されます。

イベントタイプには 2 つの種類があります。

### アクティブイベントタイプ

イベントタイプには、データベースサーバ自体のアクションの結果であるものがあります。こうしたアクティブなイベントタイプには、データベースファイルサイズ、さまざまなデータベースアクションの開始時、終了時 (BackupEnd など)、RAISERROR などが含まれます。

データベースサーバは、アクションを実行するときに、WHERE 句に定義されたトリガ条件が満たされているかどうかをチェックし、条件が満たされていればイベントタイプに対して定義されたイベントをトリガします。

### ポーリングイベントタイプ

ディスクの空き領域 (DBDiskSpace など) や IdleTime などのイベントタイプは、データベースのアクションだけでトリガされません。

このタイプのイベントに対して、データベースサーバは 30 秒ごとにポーリングします。ポーリングはデータベースの開始後、約 30 秒後から開始されます。

IdleTime イベントタイプの場合、データベースサーバはサーバが 30 秒間アイドル状態にあったかどうかをチェックします。その間まったく要求が開始されず、現在アクティブな要求もなければ、秒単位のアイドルチェック間隔時間をアイドル

時間の合計に追加します。そうでない場合はアイドル時間の合計が 0 にリセットされます。したがって、IdleTime の値は、常に 30 秒の倍数になります。IdleTime がトリガ条件に指定した間隔より長くなると、IdleTime に関連付けられたイベントハンドラが起動します。

### 1.6.3.4.2 データベースサーバによるスケジュールされたイベントのチェック

イベントのスケジュール時刻の計算は、データベースサーバの起動時と、スケジュールされた各イベントハンドラの完了時に行われます。

次のスケジュール時刻の計算は、スケジュール定義に指定された増分に基づいて、増分を前回の起動時刻に追加することで行われます。指定した増分よりイベントハンドラの実行時間が長くなり、現在の処理が終わらないうちに次のスケジュール時刻が来てしまう場合、データベースサーバは、現在の処理の後に次のスケジュール時刻が来るように増分します。

たとえば、実行に 65 分かかるイベントハンドラが 9 時 ~ 5 時の間の 1 時間ごとに起動するように要求された場合、実際には 9 時、11 時、1 時と 2 時間ごとに実行されます。

次の実行まで待機時間を設ける処理を 9 時 ~ 5 時の間で実行するには、各実行の合間に WAITFOR 文を使って、指定した完了時間が経過するまでループするようにハンドラを定義できます。

データベースサーバを断続的に実行していて、スケジュール時刻にデータベースサーバが実行中でない場合、イベントハンドラが起動時に実行されることはありません。その代わりに、次のスケジュール時刻は起動時に計算されます。たとえば、毎晩 1 時にバックアップを実行するようにスケジュールしていても、終業時にはいつもデータベースサーバを停止している場合、バックアップが実行されることはありません。

次にスケジュールされているイベント実行が 1 時間以上後の場合、データベースサーバは時間単位で次のスケジュール時間を計算します。これにより、夏時間の開始または終了のためにシステムクロックが調整されたときに、イベントが予定どおりに起動されます。

### 1.6.3.4.3 イベントハンドラの実行

イベントハンドラがトリガされると、一時的に内部接続が確立され、その上でイベントハンドラが実行されます。

ハンドラは、そのハンドラがトリガされるに至った接続で実行されるわけではありません。このため、クライアントアプリケーションとの対話に使用される MESSAGE...TO CLIENT などの文は、イベントハンドラ内では意味を持ちません。同様に、結果セットを返す文は使用できません。

ハンドラが実行される一時的な接続は、ライセンス契約の接続制限には数えられません。login\_procedure に指定したプロシージャはイベント接続では実行されません。

イベントハンドラは、イベント所有者の権限に基づいて別の接続上で実行されます。また、イベントハンドラからプロシージャを呼び出すこともできます。その場合、プロシージャは、その所有者の権限を使用して実行されます。イベントハンドラが実行される接続は、パーソナルデータベースサーバの最大接続数である 10 には含められません。

イベントエラーが発生すると、データベースサーバメッセージログに記録されます。

#### **i** 注記

イベントハンドラ内のトランザクションは、実行中にエラーが検出されなかった場合はコミットされ、エラーが検出された場合はロールバックされます。

### 1.6.3.4.4 データベースイベントの作成

イベントを作成してルーチンタスクを自動化します。

#### 前提条件

MANAGE ANY EVENT または CREATE ANY OBJECT のどちらかのシステム権限が必要です。

#### コンテキスト

スケジュールされたイベント、システムイベント、手動イベントを作成できます。

イベントハンドラを開発する場合、スケジュールやシステムイベントを追加して、後からイベントのトリガを制御できます。これには SQL Central または ALTER EVENT 文を使用します。

トリガするスケジュールもシステムイベントも設定しないでイベントハンドラを作成した場合、それは手動でトリガしたときのみ実行されます。

#### 手順

1. *SQL Anywhere 17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で **イベント** を右クリックし、**新規** > **イベント** をクリックします。
3. **イベント作成ウィザード** の指示に従います。

#### 結果

イベントがデータベースに追加されます。

## 関連情報

[スケジュールとイベントの使用によるタスクの自動化 \[950 ページ\]](#)

[イベントハンドラのトリガ \[962 ページ\]](#)

### 1.6.3.4.5 イベントハンドラのトリガ

イベントをトリガします。

#### 前提条件

MANAGE ANY EVENT システム権限が必要です。

#### コンテキスト

すべてのイベントハンドラはスケジュールやシステムイベントによって起動されるほか、手動でトリガすることができます。

イベントハンドラの開発中や、運用環境におけるいくつかのイベントに関しては、手動によるイベントのトリガが便利です。たとえば、毎月の売り上げレポートをスケジュールしている場合、月末でなくても売り上げレポートが必要になることがあります。

#### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、**イベント** をダブルクリックします。
3. イベントを右クリックして、**[トリガ]** をクリックします。

イベントは、有効にしておかなければトリガすることはできません。イベントを有効にするには、イベントを右クリックして **[有効化]** をクリックします。

4. **パラメータフィールド**に、イベントで使用するパラメータをカンマで区切って入力します。次に例を示します。

```
parameter=value,parameter=value
```

5. **OK** をクリックします。

#### 結果

イベントハンドラがトリガされます。

## 1.6.3.4.6 イベントハンドラのデバッグ

イベントハンドラをデバッグします。

### 前提条件

DEBUG ANY PROCEDURE システム権限が必要です。

### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. ▶ **モード** ▶ **デバッグ** ▶ をクリックします。
3. 左ウィンドウ枠で、**イベント** をダブルクリックします。
4. 右ウィンドウ枠で、デバッグするイベントをダブルクリックします。
5. 右ウィンドウ枠の **[SQL]** タブで、[F9] キーを押してブレークポイントを設定します。

### 結果

TRIGGER EVENT 文を使用してイベントハンドラをトリガします。設定したブレークポイントで実行が停止します。

## 1.6.3.4.7 イベントハンドラを隠す

ALTER EVENT 文を使用してイベントハンドラの定義を隠します。

### 前提条件

MANAGE ANY EVENT または ALTER ANY OBJECT のどちらかのシステム権限が必要です。

### 手順

1. データベースに接続します。

2. SET HIDDEN 句を使用して ALTER EVENT 文を実行します。

```
ALTER EVENT event-name SET HIDDEN
```

## 結果

イベントハンドラが、ISYSEVENT システムテーブルのアクションカラムに格納されているイベントハンドラ定義の中で永続的に難読化されます。

### 例

次の文は、イベントハンドラ secret\_event の定義を隠します。

```
ALTER EVENT secret_event SET HIDDEN
```

## 1.6.4 イベントのトレース

イベントのトレースでは、システム定義およびユーザ定義のトレースイベントに関する情報がイベントトレースデータ (ETD) ファイルに記録されます。

トレースセッションは、ターゲットに記録される情報を収集するトレースイベント (データベースサーバソフトウェアまたは SQL アプリケーション内の特定の位置) で構成されています。ターゲットとは、データベースサーバがトレースイベントを記録する場所 (ファイルなど) です。

イベントトレースセッション中に記録される情報には、指定されたシステム定義のトレースイベントとユーザ定義のトレースイベントからの情報が含まれます。イベントのトレースを使用すると、運用データベース上で、パフォーマンスの問題などを含むデータベースサーバの問題やアプリケーションの問題を診断できます。

### システムトレースイベント

これらはデータベースサーバから生成されるトレースイベントです。トレースイベントは、チェックポイントの開始や終了、データベースの起動や停止などの操作に対して生成されます。使用可能なシステムイベントとシステムイベントフィールドをクエリするには、それぞれ sp\_trace\_events システムプロシージャと sp\_trace\_event\_fields システムプロシージャを使用します。イベントフィールドには、イベントに関連する情報が含まれています。システムイベントはデータベースの再構築時には保持されず、どのシステムイベントセットが使用できるかはデータベースサーバのバージョンによって異なります。システムトレースイベントは、メジャーリリースごとに変更されることがあります。

監査イベントはシステムトレースイベントのサブセットで、データベースの監査ログに対するログ情報です。使用可能な監査イベントと監査イベントフィールドをクエリするには、それぞれ sp\_trace\_events システムプロシージャと sp\_trace\_event\_fields システムプロシージャを使用します。ただし、MANAGE AUDITING システム権限が必要です。

### i 注記

監査イベントは、sa\_enable\_auditing\_type システムプロシージャの使用可能なタイプパラメータの値にそのまま対応している訳ではありません。sa\_enable\_auditing\_type システムプロシージャを使用して監査を有効にするときは、監査する情報のタイプを指定します。実際の監査イベントはデータベースサーバによって生成されるシステムイベントです。

## ユーザトレースイベント

これらのトレースイベントは、アプリケーションの情報をイベントトレースセッションに記録します。ユーザトレースイベントは、データベースへの接続のすべてから参照できます。ユーザトレースイベントを作成するには CREATE TEMPORARY TRACE EVENT 文を使用します。

## イベントトレースターゲット

サポートされているイベントトレースターゲットは、イベントトレースデータ (ETD) ファイルのみです。ETD ファイルはプラットフォームに依存せず、dbmanageetd ユーティリティを使用して処理できます。sp\_read\_etd システムプロシージャを使用して、ETD ファイルからデータを取得することもできます。

## トレースセッション

トレースセッションは、データベースのトレースイベントとターゲットの集まりです。トレースセッションは、データベースへの接続のすべてから参照できます。また、トレースセッションはデータベースが停止するまで持続します。

## イベント重大度

次の SQL 重大度レベルがすべてのトレースイベントタイプに対して定義されています。

レベル	重要度の値の範囲
ALWAYS	0
CRITICAL	1-50
ERROR	51-100
WARNING	101-150
INFORMATION	151-200
DEBUG	201-255

## データベースのイベントトレースの設定

次の手順を実行して、データベースのイベントトレースを設定します。

1. 診断ログに含める情報を得るため、アプリケーションの中でユーザトレースイベントを作成します。
2. NOTIFY TRACE EVENT 文を呼び出すプロシージャを作成します。
3. トレースセッションを作成してトレースイベントをセッションに追加します。
4. イベントトレースセッションを開始します。

このセクションの内容:

#### [ユーザトレースイベントの作成 \[966 ページ\]](#)

アプリケーション内部の情報を追跡するユーザトレースイベントを作成します。

#### [トレースセッションの作成 \[967 ページ\]](#)

イベントトレース情報を診断ログに記録できるトレースセッションを作成します。

#### [トレースセッションの開始または停止 \[968 ページ\]](#)

トレースセッションを開始して、イベントトレース情報をトレース対象のターゲットに記録します。情報の記録を終えたらトレースイベントセッションを停止します。

#### [イベントトレースデータ \(ETD\) 診断ログファイルの内容の表示 \[969 ページ\]](#)

イベントトレースデータ (ETD) ファイル管理ユーティリティ (dbmanageetd) を使用して診断ログファイルの内容を表示します。

#### [使用可能なトレースイベントのリスト表示 \[970 ページ\]](#)

診断ログに記録するユーザ定義のトレースイベントとシステム定義のトレースイベントを、イベントのフィールドと一緒に表示します。

## 関連情報

[イベントトレースデータ \(ETD\) 診断ログファイルの内容の表示 \[969 ページ\]](#)

### 1.6.4.1 ユーザトレースイベントの作成

アプリケーション内部の情報を追跡するユーザトレースイベントを作成します。

## 前提条件

MANAGE ANY TRACE SESSION システム権限と NOTIFY TRACE EVENT システム権限が必要です。

## 手順

1. データベースに対して必要なユーザトレースイベントを作成します。

次に例を示します。

```
CREATE TEMPORARY TRACE EVENT my_event( id INTEGER, information LONG VARCHAR );
```

データベースの起動時に実行され、必要なユーザトレースイベントをすべて作成する、ストアプロシージャを作成できます。

2. データベースに対してトレースしているユーザ定義イベントごとに、NOTIFY TRACE EVENT 文を実行します。

次に例を示します。

```
NOTIFY TRACE EVENT my_event( 1, 'Hello world' );
```

NOTIFY TRACE EVENT 文は、トレースイベントを必要としているすべてのトレースセッションに、トレースイベントに関する情報を記録します。NOTIFY TRACE EVENT の実行時にイベントが定義されていない場合は、データベースサーバからエラーが生成されます。イベントが存在しない場合は、TRY ブロックの中に NOTIFY TRACE EVENT 文を記述して動作を定義できます。

## 結果

トレースイベントはデータベースが停止するまで存在します。

## 次のステップ

データベースのトレースセッションを作成します。

## 関連情報

[トレースセッションの作成 \[967 ページ\]](#)

### 1.6.4.2 トレースセッションの作成

イベントトレース情報を診断ログに記録できるトレースセッションを作成します。

## 前提条件

MANAGE ANY TRACE SESSION システム権限が必要です。

データベースに対して NOTIFY TRACE EVENT 文で参照するユーザトレースイベントは、すべて存在している必要があります。ただし、ユーザトレースイベントが存在しない場合にエラーをすべて処理する TRY ブロックの中に NOTIFY TRACE EVENT 文が含まれている場合は除きます。

## 手順

CREATE TEMPORARY TRACE EVENT SESSION 文を実行します。

次に例を示します。

```
CREATE TEMPORARY TRACE EVENT SESSION my_session
  ADD TRACE EVENT my_event, -- user event
  ADD TRACE EVENT SYS_ConsoleLog_Information -- system event
  ADD TARGET FILE ( SET filename_prefix='my_trace_file' );
```

ADD TRACE EVENT 句は、セッションの一部となっているユーザトレースイベントとシステムトレースイベントを指定します。ADD TARGET FILE 句は、記録するターゲットの名前を指定し、オプションで、そのターゲットに関連するパラメータを指定します。

## 結果

データベースのトレースセッションが作成されますが、手動で開始する必要があります。

## 次のステップ

データベースのトレースセッションを開始します。

## 関連情報

[ユーザトレースイベントの作成 \[966 ページ\]](#)

### 1.6.4.3 トレースセッションの開始または停止

トレースセッションを開始して、イベントトレース情報をトレース対象のターゲットに記録します。情報の記録を終えたらトレースイベントセッションを停止します。

## 前提条件

MANAGE ANY TRACE SESSION システム権限が必要です。

## 手順

トレースセッションの必要な状態を指定する ALTER TRACE EVENT SESSION 文を実行します。START または STOP  
たとえば、次の文は my\_session という名前のトレースセッションを開始します。

```
ALTER TRACE EVENT SESSION my_session  
STATE = START;
```

次の文は my\_session という名前のトレースイベントセッションを停止します。

```
ALTER TRACE EVENT SESSION my_session  
STATE = STOP;
```

## 結果

いったん開始されたセッションは、明示的に停止されるかデータベースが停止するまで、実行を続けます。

## 次のステップ

dbmanageetd ユーティリティを使用して診断ログファイルの内容を表示します。

## 関連情報

[イベントトレースデータ \(ETD\) ファイル管理ユーティリティ \(dbmanageetd\) \[1104 ページ\]](#)

### 1.6.4.4 イベントトレースデータ (ETD) 診断ログファイルの内容の表示

イベントトレースデータ (ETD) ファイル管理ユーティリティ (dbmanageetd) を使用して診断ログファイルの内容を表示します。

## 前提条件

トレースセッションから ETD ファイルに記録したイベントトレース情報が必要です。

## 手順

1. dbmanageetd ユーティリティを実行し、テキスト形式で診断ログの内容を表示します。次に例を示します。

```
dbmanageetd diagnostic-log.etd -o diagnostic-log.txt
```

2. ファイルをテキストエディタで開きます。

下の表に、イベントトレースの重大度レベルを示します。

レベル	重要度の値の範囲
ALWAYS	0
CRITICAL	1-50
ERROR	51-100
WARNING	101-150
INFORMATION	151-200
DEBUG	201-255

## 結果

診断ログが生成され、テキストエディタで表示できます。

## 関連情報

[イベントトレースデータ \(ETD\) ファイル管理ユーティリティ \(dbmanageetd\) \[1104 ページ\]](#)

### 1.6.4.5 使用可能なトレースイベントのリスト表示

診断ログに記録するユーザ定義のトレースイベントとシステム定義のトレースイベントを、イベントのフィールドと一緒に表示します。

## 前提条件

MANAGE ANY TRACE SESSION システム権限が必要です。

## 手順

1. すべてのトレースイベントのリストを表示するには、sp\_trace\_events システムプロシージャを実行します。
2. トレースイベントのフィールドのリストを表示するには、sp\_trace\_event\_fields システムプロシージャを実行します。

下の表に、イベントトレースの重大度レベルを示します。

レベル	重要度の値の範囲
ALWAYS	0
CRITICAL	1-50
ERROR	51 ~ 100
WARNING	101-150
INFORMATION	151-200
DEBUG	201-255

## 結果

サポートされているトレースイベントのリストを示す結果セットがデータベースサーバから返されます。

### 1.6.5 データベースのトラブルシューティング

データベースのトラブルシューティングに使用できる、多くのツールおよび方法があります。

このセクションの内容:

[トラブルシューティング: データベースサイズの予期しない変化について \[972 ページ\]](#)

データベースファイルのサイズが、予期せずに増えたり減ったりすることがあります。

[トラブルシューティング: エラーのレポート \[973 ページ\]](#)

クラッシュレポートや診断情報レポートなどのパフォーマンスデータを送信して、製品の品質向上にご協力ください。

[トラブルシューティング: データベースサーバの起動 \[974 ページ\]](#)

データベースサーバの起動のトラブルシューティングには、いくつかの方法があります。

[トラブルシューティング: データベースサーバのパフォーマンス警告 \[975 ページ\]](#)

データベースサーバのパフォーマンス警告は、データベースサーバメッセージログに出力されます。

[ネットワーク通信のトラブルシューティング \[979 ページ\]](#)

ネットワーク通信のトラブルシューティングにはいくつかの方法があります。

## 関連情報

[パフォーマンス向上のヒント \[1366 ページ\]](#)

[SQL Anywhere プロファイラ \[1411 ページ\]](#)

[SQL Anywhere モニタ \[1207 ページ\]](#)

[パフォーマンスの向上、診断、モニタリング \[1205 ページ\]](#)

## 1.6.5.1 トラブルシューティング: データベースサイズの予期しない変化について

データベースファイルのサイズが、予期せずに増えたり減ったりすることがあります。

データベースページは、ページのすべてのレコードが削除されると解放されます。解放されたデータベースページは再利用可能になりますが、ファイルから削除できません。ただし、今後 INSERT 文や UPDATE 文で解放されたページを使用できます。

挿入、更新、削除などの変更がデータベースに加えられるたびに、システム DB 領域内に格納されているロールバックログにエントリが追加されます。これらの操作が多数実行されてからコミットが実行されると、ロールバックログは非常に大きくなり、データベースのサイズが増大する可能性があります。

チェックポイントログは、システム DB 領域の最後に格納されています。データベースサーバが停止すると、チェックポイントログはトランケートされ、システム DB 領域が小さくなります。DELETE または TRUNCATE 操作によって解放されたページは、今後再利用できるようにデータベースファイル内に残り、ファイルから削除できません。

データベースファイルのサイズが増大している場合、または期待どおりに減少しない場合は、次の操作を行います。

- INSERT 文、UPDATE 文、または DELETE 文を使用する場合は、COMMIT を頻繁に実行します。COMMIT が実行されると、ロールバックログに割り付けられたページは、システム DB 領域内で再利用できるように解放されます。
- UPDATE 文または DELETE 文を使用する場合、または INSERT 文を使用し、大きいインデックスが関連する場合は、CHECKPOINT を時々実行します。チェックポイントログ内のページは、各チェックポイント後にチェックポイントログで再利用できるようになります。
- TRUNCATE TABLE を実行します。ページレベルの削除が実行されます。この場合、ページのコピーをチェックポイントログに追加したり、個々のローレベルの操作をロールバックログに追加したりする必要はありません。TRUNCATE TABLE によって解放されたページは、次のチェックポイントの後でのみ再利用できます。TRUNCATE TABLE を実行すると、次の条件に当てはまる場合、ページレベルの削除が実行されます。
  - トランケートするテーブルへの外部キー、またはテーブルからの外部キーがありません。
  - TRUNCATE TABLE がトリガ内で実行されていません。
  - TRUNCATE TABLE がアトミック操作で実行されていません。
  - チェックポイントログページがシステム DB 領域ファイルの最後に書き込まれています。これらのページは、データベースが停止すると削除されます。

データベースの再構築によって、データベースのサイズが減少する場合があります。これは、再構築されたデータベースの空きページが少なくなるためです。

## 関連情報

[情報ユーティリティ \(dbinfo\) \[1111 ページ\]](#)

## 1.6.5.2 トラブルシューティング: エラーのレポート

クラッシュレポートや診断情報レポートなどのパフォーマンスデータを送信して、製品の品質向上にご協力ください。

### 前提条件

必要な場合は、プロキシホストとポートの設定を設定してください。

### コンテキスト

レポートにネットワークユーザ名やパスワードなどの機密情報が記載されることのないよう、最大限の注意を払う必要があります。

エラーレポートは、削除するまでコンピュータのファイルシステムに格納されます。

### 手順

次のコマンドを実行します。

```
dbsupport -sa
```

### 結果

診断ディレクトリに保存されているすべてのクラッシュレポートと診断情報が、ソフトウェア開発チームに送信されます。

このセクションの内容:

[パフォーマンスデータの定期送信 \[974 ページ\]](#)

製品の改良に役立てるために、ソフトウェアには、パフォーマンスデータをソフトウェア開発チームにレポートする機能があります。

### 関連情報

[サポートユーティリティ \(dbsupport\) \[1165 ページ\]](#)

[Performance data collection details \(パフォーマンスデータ収集詳細\)](#)

## 1.6.5.2.1 パフォーマンスデータの定期送信

製品の改良に役立てるために、ソフトウェアには、パフォーマンスデータをソフトウェア開発チームにレポートする機能があります。

インストール中に、この機能がデフォルトで有効になります。また、次の機能も有効にすることができます。

- Interactive SQL で、**ツール** > **オプション** > **サポート** をクリック
- SQL Central で、**ヘルプ** > **SQL Anywhere17** > **更新チェッカーの設定** をクリックします。
- サポートユーティリティ (dbsupport) で、-cc 自動送信オプションを指定

### OEM ユーザ

デフォルトでは、管理ツールの展開時に、ソフトウェア開発チームにパフォーマンスデータを自動送信する機能は無効になっています。ただし、ユーザは、この機能を管理ツールで有効または無効にすることができます。ユーザのデータを送信するかどうかをユーザが変更できないようにするには、OEM.ini ファイルの showPerformanceDataUI オプションを false に設定します。

ユーザがパフォーマンスデータを SAP に送信できるようにするには、サポートユーティリティ (dbsupport) を -cc 自動送信オプションとともに使用する必要があります。

### 関連情報

[サポートユーティリティ \(dbsupport\) \[1165 ページ\]](#)

## 1.6.5.3 トラブルシューティング: データベースサーバの起動

データベースサーバの起動のトラブルシューティングには、いくつかの方法があります。

### トランザクションログファイルが有効であることを確認する

既存のトランザクションログが無効な場合、データベースサーバは起動しません。

たとえば、開発中に、データベースを新しいバージョンに置き換えると同時にトランザクションログを削除しなかったとします。これは、トランザクションログとデータベースが異なる原因となり、結果として無効なトランザクションログになります。

## テンポラリファイル用の十分なディスク領域があるかを確認する

一時ファイルを使用して、ソフトウェアが実行中の間、情報を格納します。

このファイルは、SATMP 環境変数で指定されるディレクトリ (通常は `c:\¥temp`) に保存されます。

テンポラリディレクトリに使用できる十分なディスク領域がないと、サーバを起動するときに問題が生じることがあります。

## ネットワーク通信の起動時の問題をデバッグする

TCP/IP または通信関係のエラーでサーバが起動に失敗した場合、`-z` サーバデバッグオプションを使用できます。

起動情報は、データベースサーバメッセージウィンドウに表示されます。また、`-o` オプションを使用して、結果を出力ファイルに記録することもできます。

## 関連情報

[-z データベースサーバオプション \[509 ページ\]](#)

[-o データベースサーバオプション \[452 ページ\]](#)

[SATMP 環境変数 \[553 ページ\]](#)

## 1.6.5.4 トラブルシューティング: データベースサーバのパフォーマンス警告

データベースサーバのパフォーマンス警告は、データベースサーバメッセージログに出力されます。

これらの警告は、データベースまたはデータベースサーバでそのパフォーマンスが影響を受ける状態になるとレポートされます。

このセクションの内容:

**パフォーマンス警告: サーバキャッシュサイズがデータベース "%1" には小さすぎます。 [976 ページ]**

データベースサーバの最大キャッシュサイズが、データベースのすべてのオープンな DB 領域の 10% 未満の場合、この警告が報告されます。

**パフォーマンス警告: データベース "%3" では、サーバ用に設定された最大値 %2 にページサイズ %1 が一致せず、キャッシュの非効率的な使用の原因となっています。 [977 ページ]**

データベースサーバのバッファプールのページサイズは、少なくともそのデータベースサーバで起動されているいずれかのデータベースのページサイズと同じである必要があります。

**パフォーマンス警告: データベース "%2" のテーブル "%1" にユニークなインデックスまたはプライマリキーがありません。 [977 ページ]**

テーブルが更新され、そのテーブルに 11 以上のローが格納されており、プライマリキーまたはユニークなインデックスがない場合に、このパフォーマンス警告がレポートされます。

パフォーマンス警告: ページサイズがデータベース "%1" には小さすぎます。[978 ページ]

2つの条件がそろると、ページサイズがデータベース "%1" には小さすぎることに対するパフォーマンス警告が表示されます。

パフォーマンス警告: データベースファイル "%1" は %2 のディスクフラグメントで構成されています。[978 ページ]

データベースサーバでデータベースが稼働を開始すると、SQL Anywhere は、オペレーティングシステムによってレポートされたメインデータベースファイルのファイルフラグメント数をレポートします。

パフォーマンス警告: データベース "%3" では、テーブル "%1" の AUTOINCREMENT カラム "%2" にインデックスが設定されていません。[978 ページ]

このパフォーマンス警告は、永続ベーステーブルに AUTOINCREMENT カラムが含まれ、そのカラムにインデックスが設定されていない場合にレポートされます。

パフォーマンス警告: 参照オブジェクトの 1 つで DDL 処理が実行されたため、データベース "%2" のビュー "%1" は無効になりました。[979 ページ]

この警告は、DDL 処理によって既存のビューが無効になった場合にレポートされます。

### 1.6.5.4.1 パフォーマンス警告: サーバキャッシュサイズがデータベース "%1" には小さすぎます。

データベースサーバの最大キャッシュサイズが、データベースのすべてのオープンな DB 領域の 10% 未満の場合、この警告が報告されます。

キャッシュサイズはデータベースサーバのバッファプールサイズの上限で、サーバによって計算されるか、または -ch オプションで明示的に指定されます。

たとえば、21 GB のデータベースを実行しているデータベースサーバは、サーバの最大キャッシュサイズが 2.1 GB 未満になると、この警告を発行します。キャッシュサイズはデータベースごとに計算されるため、この警告は、複数のデータベースが同じデータベースサーバで起動されている場合でも発行されない可能性があります。

#### 関連情報

ヒント: パフォーマンス向上のためのキャッシュの使用 [1370 ページ]

動的キャッシュサイズ決定 [1373 ページ]

キャッシュサイズをモニタリングする統計 [1375 ページ]

-ch データベースサーバオプション [392 ページ]

### 1.6.5.4.2 パフォーマンス警告: データベース "%3" では、サーバ用に設定された最大値 %2 にページサイズ %1 が一致せず、キャッシュの非効率的な使用の原因となっています。

データベースサーバのバッファプールのページサイズは、少なくともそのデータベースサーバで起動されているいずれかのデータベースのページサイズと同じである必要があります。

データベースサーバは、さらに大きいページサイズをキャッシュに使用できます。これは -gp オプションを使用して指定できます。ただし、SQL Anywhere データベースサーバでは、バッファプール内でデータベースページとページフレームが 1:1 でマッピングされている必要があります。メモリ内の各ページフレームには、1 つのデータベースページのみを格納でき、ページは 1 つのページフレームにのみ表示されます。キャッシュで 8 KB のページが使用され、データベースで 4 KB のページが使用されている場合、メモリ内の各キャッシュページフレームには 4 KB のデータのみが格納され、そのページフレームの残りの 4 KB は無駄になります。この状況では、キャッシュサイズは必要な容量の半分になります。

#### 関連情報

[最大ページサイズの考慮事項 \[319 ページ\]](#)

[-gp データベースサーバオプション \[427 ページ\]](#)

### 1.6.5.4.3 パフォーマンス警告: データベース "%2" のテーブル "%1" にユニークなインデックスまたはプライマリキーがありません。

テーブルが更新され、そのテーブルに 11 以上のローが格納されており、プライマリキーまたはユニークなインデックスがない場合に、このパフォーマンス警告がレポートされます。

トランザクションログに論理ローの操作 (UPDATE、INSERT、DELETE) が記録され、物理的なページへの変更は記録されません。トランザクションログ内のロー操作は、プライマリキーまたはユニークなインデックス値によって識別されます。したがって、データベースをリカバリする場合、トランザクションログ内のその操作について更新された値を正しいローに適用できます。プライマリキーまたはユニークなインデックスが存在しない場合は、ローのすべての値がローのプライマリキーとして使用されます。この動作によって、トランザクションログのサイズは大幅に増大します。テーブルにインデックスがない場合は、データベースのリカバリ時間が長くなる可能性があります。

#### 関連情報

[トランザクションログサイズの考慮事項 \[287 ページ\]](#)

[トランザクションログ \[279 ページ\]](#)

#### 1.6.5.4.4 パフォーマンス警告: ページサイズがデータベース "%1" には小さすぎます。

2つの条件がそろると、ページサイズがデータベース "%1" には小さすぎることに対するパフォーマンス警告が表示されます。

1. データベースのページサイズが 2 KB です。
2. データベース全体のページ数が 100000 (データベースサイズで約 200 MB に相当) より多くなっています。

領域の効率とパフォーマンスを向上させるために、データベースをアンロードし、ページサイズが 4 KB 以上の新しいデータベースに再ロードすることをお奨めします。

#### 1.6.5.4.5 パフォーマンス警告: データベースファイル "%1" は %2 のディスクフラグメントで構成されています。

データベースサーバでデータベースが稼働を開始すると、SQL Anywhere は、オペレーティングシステムによってレポートされたメインデータベースファイルのファイルフラグメント数をレポートします。

データベースが格納されているディスクボリュームのディスクが過度に断片化されると、I/O 処理中にディスクの遅延時間が長くなる可能性があるため、パフォーマンスの問題が発生することがあります。

ファイルシステムの断片化の問題は、データベース内のテーブルページの断片化とは無関係です。これらの別々の問題を分析して解決するには、それぞれ異なる方法とツールを使用します。ファイルフラグメント数は、Windows ベースの SQL Anywhere データベースサーバの場合のみレポートされます。UNIX ファイルシステム (Linux ファイルシステムを含む) では、Windows ベースのシステムよりも断片化の問題は大幅に少なくなります。

Windows プラットフォームでは、通常、データベースファイルは複数のフラグメントで構成されています。データベースについてこのメッセージを受け取った場合は、データベースファイルのサイズに対するフラグメント数を確認すると、システム管理者による対処が必要かどうかを判断できます。たとえば、25 個のフラグメントで構成されている 100 GB のデータベースファイルは、深刻な問題になるとは見なされません。ただし、300 個のディスクフラグメントで構成されている 50 MB のデータベースファイルは、パフォーマンスの問題の影響を受ける可能性があります。ファイルのフラグメンテーションの問題を解決するには、データベースをディスクパーティションに単独で入れ、使用可能な Windows ディスクデフラグユーティリティの 1 つを定期的に行います。

#### 1.6.5.4.6 パフォーマンス警告: データベース "%3" では、テーブル "%1" の AUTOINCREMENT カラム "%2" にインデックスが設定されていません。

このパフォーマンス警告は、永久ベーステーブルに AUTOINCREMENT カラムが含まれ、そのカラムにインデックスが設定されていない場合にレポートされます。

この警告が生成されるのは、ISYSTABCOL カタログテーブルの MAX\_IDENTITY カラムに格納されている、そのカラムの最大 ID 値が不明な場合、または有効でなくなる場合のみです。この状況は、既存のカラムが AUTOINCREMENT カラムに変更された場合、またはデータベースのグローバルオートインクリメント設定が変更された場合に発生する可能性があります。

## 1.6.5.4.7 パフォーマンス警告: 参照オブジェクトの 1 つで DDL 処理が実行されたため、データベース "%2" のビュー "%1" は無効になりました。

この警告は、DDL 処理によって既存のビューが無効になった場合にレポートされます。

ビューが変更されたスキーマオブジェクトに依存し、その後ビューの再コンパイルに失敗した場合、そのビューのステータスは INVALID のままです。その後クエリ内からビューが参照されるたびに、データベースサーバはビューを再コンパイルしようとします。

## 1.6.5.5 ネットワーク通信のトラブルシューティング

ネットワーク通信のトラブルシューティングにはいくつかの方法があります。

ネットワークソフトウェアにはさまざまなコンポーネントが含まれているため、問題が発生しやすくなります。ここで、ネットワークトラブルシューティングのヒントをいくつか説明しますが、ネットワークのトラブルシューティングを支援する一番の情報源は、社内の IT 部門またはネットワーク管理者です。

### ロギングの使用

-z および -o データベースサーバオプションを指定すると、トラブルシューティング目的で、診断通信メッセージやその他のメッセージがデータベースサーバメッセージウィンドウに記録されます。

クライアントでは、LogFile 接続パラメータを接続文字列に追加することによって、クライアント側の診断通信メッセージがファイルに記録されます。これらのメッセージを利用して、接続に失敗したときの状況、接続試行に使用された接続パラメータ、使用された通信リンクを特定できます。

### タイムアウト値の調整

接続が予期せずに切断してしまう場合は、活性タイムアウトまたはアイドルタイムアウトの値を調整することを検討してください。

### 関連情報

- z データベースサーバオプション [509 ページ]
- LogFile (LOG) 接続パラメータ [92 ページ]
- LivenessTimeout (LTO) 接続パラメータ [91 ページ]
- tl データベースサーバオプション [484 ページ]
- IdleTimeout (IDLE) 接続パラメータ [85 ページ]

## 1.7 データベース管理ツールとユーティリティ

データベースを管理するためのツールや方法には、いくつかの種類があります。

このセクションの内容:

### [Interactive SQL \[980 ページ\]](#)

Interactive SQL では、SQL 文の実行、SQL スクリプトファイルの実行、プランの表示と比較を行うことができます。

### [SQL Central \[1041 ページ\]](#)

SQL Central は、データベースサーバ、データベース、およびそれらに含まれているオブジェクトを管理するためのグラフィカルツールです。

### [ソフトウェアの更新 \[1062 ページ\]](#)

複数の方法でソフトウェアの更新を確認できます。

### [データベース管理ユーティリティ \[1064 ページ\]](#)

データベース管理タスクを実行するための複数のユーティリティプログラムが提供されています。

### 1.7.1 Interactive SQL

Interactive SQL では、SQL 文の実行、SQL スクリプトファイルの実行、プランの表示と比較を行うことができます。

Interactive SQL では、以下のデータベースがサポートされます。

- SQL Anywhere データベース
- Ultra Light データベース
- SAP HANA データベース
- SAP IQ データベース

このセクションの内容:

#### [Interactive SQL の起動 \[981 ページ\]](#)

Interactive SQL を使用して数種類のデータベースに接続します。

#### [SQL 文の実行 \(Interactive SQL\) \[983 ページ\]](#)

Interactive SQL から複数の SQL 文を実行します。

#### [結果セット \(Interactive SQL\) \[986 ページ\]](#)

Interactive SQL でクエリを実行した後、結果セットのソートや編集ができます。

#### [Interactive SQL の SQL 文 \[996 ページ\]](#)

SQL 文の中には、Interactive SQL でのみサポートされるものもあります。

#### [詳細とノードの統計を含むグラフィカルなプランの作成 \(Interactive SQL\) \[996 ページ\]](#)

プランビューアは、グラフィカルなプランと Ultra Light データベースのテキストプランを表示するためのグラフィカルツールです。

### [Interactive SQL のカスタマイズ \[998 ページ\]](#)

結果セットの表示方法を設定し、警告メッセージを無効にし、さらに .sql ファイルのデフォルトエディタとして、Interactive SQL を設定します。

### [クエリエディタ \(Interactive SQL\) \[1037 ページ\]](#)

クエリエディタは、SELECT 文の構築を支援する Interactive SQL のツールです。

### [式エディタ \(Interactive SQL\) \[1040 ページ\]](#)

式エディタによって、探索条件、計算カラム、サブクエリを作成できます。

### [管理ツールのセキュリティ \(Interactive SQL\) \[1041 ページ\]](#)

管理ツールでは、情報を保護するための多くのセキュリティ機能が使用できます。

## 1.7.1.1 Interactive SQL の起動

Interactive SQL を使用して数種類のデータベースに接続します。

### 前提条件

#### Linux

Linux デスクトップアイコンをサポートする Linux のバージョンが、SQL Anywhere のインストール時にインストール済みになっている必要があります。

#### UNIX および Mac OS X

SQL Anywhere ユーティリティは、事前にソース化されている必要があります。

### 手順

Interactive SQL を開き、データベースに接続します。

オプション	アクション
コマンドライン、UNIX、またはその両方	<p>次のコマンドを実行します。</p> <pre>dbisql</pre> <p>データベースの接続パラメータを指定する -c オプションを含めていない場合、または十分な接続パラメータが指定されていない場合、データベースの接続情報を入力できる接続ウィンドウが表示されます。</p>
Windows	<ol style="list-style-type: none"><li>1. <b>スタート</b> &gt; <b>プログラム</b> &gt; <b>SQL Anywhere17</b> &gt; <b>管理ツール</b> &gt; <b>Interactive SQL</b> をクリックします。</li><li>2. 接続ウィンドウで、データベースの接続情報を入力します。</li><li>3. 接続をクリックします。</li></ol>

オプション	アクション
SQL Central	<ol style="list-style-type: none"> <li>1. <b>ツール</b> &gt; <i>SQL Anywhere17</i> &gt; <i>Interactive SQL を開く</i> をクリックします。</li> <li>2. 接続ウィンドウで、データベースの接続情報を入力します。</li> <li>3. 接続をクリックします。</li> </ol> <p>➔ ヒント</p> <p>次のいずれかの方法で SQL Central から Interactive SQL にアクセスすることもできます。</p> <ul style="list-style-type: none"> <li>○ データベースを選択し、<b>ファイル</b> &gt; <i>Interactive SQL を開く</i> をクリックします。</li> <li>○ データベースを右クリックし、<i>Interactive SQL を開く</i> をクリックします。</li> <li>○ スタアドプロシージャを右クリックし、<i>Interactive SQL から実行</i> をクリックします。<i>SQL 文</i> ウィンドウ枠にプロシージャへの呼び出しが指定されて Interactive SQL が開き、スタアドプロシージャが実行されます。</li> <li>○ テーブルまたはビューを右クリックし、<i>Interactive SQL によるデータ表示</i> をクリックします。<code>SELECT * FROM table-name</code> が指定されて Interactive SQL が開き、クエリが実行されます。</li> </ul>
Mac OS X	<ol style="list-style-type: none"> <li>1. Finder で、/Applications/SQLAnywhere17 にある <i>Interactive SQL</i> をダブルクリックします。</li> <li>2. 接続ウィンドウで、データベースの接続情報を入力します。</li> <li>3. 接続をクリックします。</li> </ol>
Linux	<ol style="list-style-type: none"> <li>1. <b>アプリケーション</b> &gt; <i>SQL Anywhere17</i> &gt; <b>管理ツール</b> &gt; <i>Interactive SQL</i> をクリックします。</li> <li>2. 接続ウィンドウで、データベースの接続情報を入力します。</li> <li>3. 接続をクリックします。</li> </ol>

## 結果

Interactive SQL が起動し、データベースに接続されます。

### 例

- Interactive SQL を起動し、コマンドラインからサンプルデータベースに接続するには、次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;DSN=SQL Anywhere 17 Demo"
```

- Interactive SQL を起動し、SAP HANA データベースに接続するには、次のコマンドを実行します。

```
dbisql -hana -c "user=username;password=userpasswd;host=hana-server:30015"
```

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[UNIX および Mac OS X 環境変数 \[542 ページ\]](#)

[Interactive SQL ユーティリティ \(dbisql\) \[1125 ページ\]](#)

## 1.7.1.2 SQL 文の実行 (Interactive SQL)

Interactive SQL から複数の SQL 文を実行します。

### 手順

1. InteractiveSQL で、SQL 文ウィンドウ枠にクエリを入力します。1 つ以上の文を入力する場合、各文はコマンドデリミタ (デフォルトではセミコロン (;)) で終了するか、独立した各行でキーワード **GO** を使用して終了する必要があります。
2. 以下のオプションのいずれかを選択して、SQL 文を実行します。

オプション	アクション
すべての SQL 文を実行する	▶ SQL ▶ 実行 ▶ をクリックします。
選択した SQL 文を実行する	文を選択し、▶ SQL ▶ 選択の実行 ▶ をクリックします。
SQL 文を 1 文ずつ実行する	実行したい文にカーソルを置き、▶ SQL ▶ シングルステップ ▶ をクリックします。

3. SQL Anywhere または Ultra Light データベースでエラーが発生した場合、エラーウィンドウが表示されます。エラーは、参照目的で履歴タブにも記録されます。

### 結果

Interactive SQL によって、文の結果セットが表示されます。デフォルトでは、ロー番号が結果セットの左に表示されます。

このセクションの内容:

#### [文のキャンセル \(Interactive SQL\) \[984 ページ\]](#)

キャンセルオペレーションにより、現在の処理が停止されます。

#### [SQL スクリプトファイルの実行 \(Interactive SQL\) \[984 ページ\]](#)

Interactive SQL を使用して、スクリプトファイルを開いたり、表示、実行、保存を行ったりします。SQL スクリプトファイルは、SQL 文を含むテキストファイルであり、SQL 文を繰り返し実行する場合に便利です。

#### [複数の Interactive SQL タブを開く \[985 ページ\]](#)

それぞれ別個のデータベース接続に対応する複数の Interactive SQL タブを開きます。

### 関連情報

[Interactive SQL のカスタマイズ \[998 ページ\]](#)

[command\\_delimiter オプション \[Interactive SQL\] \[1007 ページ\]](#)

[データを表示した場合の予期しない記号のトラブルシューティング \[583 ページ\]](#)

## 1.7.1.2.1 文のキャンセル (Interactive SQL)

キャンセルオペレーションにより、現在の処理が停止されます。

Interactive SQL ツールバーで**停止**ボタンをクリックすると、文をキャンセルできます。

SQL スクリプトファイルが処理中の場合は、必要なアクション (コマンドファイルを**停止**するか、**続行**するか、*Interactive SQL* を**終了**する) を指定するためのプロンプトが表示されます。これらのアクションは、Interactive SQL の `on_error` オプションを使用して制御できます。

### 関連情報

[on\\_error オプション \[Interactive SQL\] \[1021 ページ\]](#)

## 1.7.1.2.2 SQL スクリプトファイルの実行 (Interactive SQL)

Interactive SQL を使用して、スクリプトファイルを開いたり、表示、実行、保存を行ったりします。SQL スクリプトファイルは、SQL 文を含むテキストファイルであり、SQL 文を繰り返し実行する場合に便利です。

### 手順

SQL スクリプトファイルは次のいずれかの方法で実行できます。

オプション	アクション
Interactive SQL の READ 文を使用する	たとえば、次の文はファイル <code>temp.sql</code> を実行します。 <pre>READ temp.sql;</pre>
スクリプトファイルをロードして実行する	1. <b>ファイル</b> > <b>開く</b> をクリックします。 2. ロードするスクリプトファイルを選択します。
スクリプトファイルロードしないで実行する	<b>ファイル</b> > <b>スクリプトの実行</b> をクリックします。
Interactive SQL のコマンドライン引数としてスクリプトファイルを指定します。	たとえば、次のコマンドは、サンプルデータベースに対して SQL スクリプトファイル <code>myscript.sql</code> を実行します。 <pre>dbisql -c "DSN=SQL Anywhere 17 Demo;PWD=sql" myscript.sql</pre>

## 関連情報

[お気に入りリストの管理 \[1001 ページ\]](#)

### 1.7.1.2.3 複数の Interactive SQL タブを開く

それぞれ別個のデータベース接続に対応する複数の Interactive SQL タブを開きます。

## コンテキスト

異なるデータベースサーバ上にある 2 つ (またはそれ以上) のデータベースに同時に接続したり、単一のデータベースへの同時接続を開始したりできます。

## 手順

1. Interactive SQL で、**ウインドウ > 新しいタブ** をクリックします。

#### → ヒント

SQLCONNECT 環境変数が設定されている場合や、すでにデータベースに接続している場合は、情報の入力を求める前に、データベースサーバがこの情報を使ってデータベースに接続しようとします。

同様に、ULSQLCONNECT 環境変数が設定されている場合や、すでに Ultra Lite データベースに接続している場合は、情報の入力を求める前に、データベースサーバがこの情報を使ってデータベースに接続しようとします。それに失敗した場合、またはデータベースにまだ接続していない場合に、**接続**タブが表示されます。

2. **接続**タブで、データベースの接続情報を入力し、**接続**をクリックします。

また、**SQL > 接続**、または **SQL > 切断** をクリックし、データベースに接続あるいはデータベースから切断できます。

## 結果

接続情報 (データベース名、ユーザ ID、データベースサーバ名など) が Interactive SQL のタイトルバーに表示されます。

## 関連情報

[SQLCONNECT 環境変数 \[557 ページ\]](#)

### 1.7.1.3 結果セット (Interactive SQL)

Interactive SQL でクエリを実行した後、結果セットのソートや編集ができます。

結果セットをソートするには、**結果タブの**カラムヘッダをクリックして、そのカラムを基準にして結果をソートします。**結果タブ**に結果セット全体が含まれていない場合、残りの結果をフェッチするよう要求されます。残りを取得しない場合、現在取得されている結果のみがソートされます。

また、結果セットからローを選択し、他のアプリケーションで使用できるようにコピーすることもできます。結果セットのフィールドデリミタ、引用文字、エスケープ文字は、それぞれ `isql_field_separator`、`isql_quote`、`isql_escape_character` オプションによって制御されます。これらのオプションは、Interactive SQL の**オプション**ウィンドウを使用するか、または SET OPTION 文を実行して表示および変更できます。

Interactive SQL は、ローの編集、挿入、削除をサポートしています。結果セットの編集には、UPDATE、INSERT、DELETE の各文を実行した場合と同様の効果があります。結果セットを編集すると、対応する INSERT 文、UPDATE 文、DELETE 文が Interactive SQL のコマンド履歴に追加されます。

結果セットのローまたは値を編集するには、値を変更したいテーブルまたはカラムに関する適切な権限が必要です。たとえば、ローを削除するには、ローが属するテーブルの DELETE 権限が必要です。

次の場合には、結果セットを編集できません。

- プライマリキーを持つテーブルからカラムを選択したが、一部のプライマリキーカラムを選択していない
- JOIN の結果セットを編集しようとした (たとえば、結果セットに複数のテーブルのデータがある場合)
- 編集が無効になっているテーブルを編集しようとした

次の場合には、結果セットの編集に失敗することがあります。

- 権限のないローやカラムを編集しようとした
- 無効な値を入力した (たとえば、数値カラムに文字列を入力したり、NULL を使用できないカラムに NULL を入力した場合)

編集に失敗すると、エラーを説明する Interactive SQL エラーメッセージが表示されます。データベーステーブルの値は変更されません。

このセクションの内容:

#### [Interactive SQL の結果セットでローを編集する \[987 ページ\]](#)

Interactive SQL を使用して、データベーステーブルの値を編集します。

#### [Interactive SQL でのテーブル編集の無効化 \[988 ページ\]](#)

Interactive SQL のオプションウィンドウを使用して、テーブル編集を無効にします。

#### [Interactive SQL の結果セットに新しいローを挿入する \[989 ページ\]](#)

Interactive SQL 内のテーブルにローを追加します。

#### [Interactive SQL の結果セットからローを削除する \[990 ページ\]](#)

Interactive SQL で結果セットからローを削除できます。

#### [Interactive SQL 結果セットからロー、カラム、セルをコピーする \[991 ページ\]](#)

Interactive SQL に表示される結果セットからセル、ロー、カラムをコピーし、他のアプリケーションに貼り付けます。

#### [Interactive SQL の結果セットから SQL 文を生成する \[992 ページ\]](#)

結果セット内で選択したローに対する INSERT 文、DELETE 文、UPDATE 文を作成します。

#### [イメージの表示 \(Interactive SQL\) \[993 ページ\]](#)

Interactive SQL を使用して、SVG、BLOB などのイメージや、5000000 ピクセル未満のイメージを表示します。

[HTML データと XML データの表示 \(Interactive SQL の場合\) \[994 ページ\]](#)

Interactive SQL で、クエリの結果セットから HTML および XML データを別ウィンドウに表示します。

## 関連情報

[Interactive SQL オプション \[1003 ページ\]](#)

[Interactive SQL の履歴を表示する \[1000 ページ\]](#)

### 1.7.1.3.1 Interactive SQL の結果セットでローを編集する

Interactive SQL を使用して、データベーステーブルの値を編集します。

#### 前提条件

変更するカラムに対する UPDATE 権限が必要です。

SQL Anywhere および Ultra Light データベースでは、テーブル編集を無効にしないでください。

#### コンテキスト

結果セットを編集するとき、同時に変更できるのは 1 つのローの値のみです。

#### 手順

1. Interactive SQL でクエリを実行します。
2. **結果**タブで、変更したい値をクリックします。
3. 値を右クリックし、**ローの編集**をクリックするか、F2 キーを押します。

その値を含むテーブルセルに ... ボタンが表示されます。

4. ... をクリックし、次のいずれかのオプションを選択します。

**[ウィンドウで編集]**

新しい値を入力できるウィンドウが開きます (文字データフィールドのみ)。

**[NULL に設定]**

カラムが NULL 入力不可の場合、このメニュー項目は表示されません。

#### [デフォルトに設定]

このメニュー項目が表示されるのは、新しいローをテーブルに追加するときに、カラムにデフォルト値が定義されている場合のみです。

#### [ファイルからロード]

ファイルを参照し、ファイルからセルの内容をロードできます。

新しい値を入力します。ローの他の値を変更するには、Tab キーまたは Shift + Tab キーを押して、他の値に移動します。

5. ローの値を編集したら、Enter キーを押してデータベースを更新します。

Esc キーを押して、選択した値に加えた変更をキャンセルします。

6. COMMIT 文を実行し、テーブルに対する変更を永続的なものにします。

## 結果

結果セットのローが編集されます。

### 1.7.1.3.2 Interactive SQL でのテーブル編集の無効化

Interactive SQL のオプションウィンドウを使用して、テーブル編集を無効にします。

## コンテキスト

Interactive SQL を配備するときに、テーブル編集を無効にすることができます。

## 手順

1. Interactive SQL で、**ツール** > **オプション** をクリックしてから、*SQL Anywhere*、*Ultra Light*、*IQ*、または *SAP HANA* のいずれかをクリックします。
2. **スクロール可能なテーブル**が選択されていることを確認し、**編集の無効化**をクリックします。
3. **OK** をクリックします。
4. クエリを実行します。

テーブル編集の変更を反映させるには、新しいクエリを実行する必要があります。

## 結果

テーブル編集が無効になります。

### 1.7.1.3.3 Interactive SQL の結果セットに新しいローを挿入する

Interactive SQL 内のテーブルにローを追加します。

#### 前提条件

ローを追加するには、テーブルの INSERT オブジェクトレベル権限が必要です。

結果セットを取得するにはクエリの実行を完了している必要があります。

#### コンテキスト

ローに値を追加するには、結果セット内のカラム間をタブで移動します。

カラムには、無効なデータ型は入力できません。たとえば、INT データ型を受け入れるカラムには、文字列は入力できません。

#### 手順

1. Interactive SQL で結果セットを右クリックし、**ローの追加**をクリックします。

新しい空白のローが表示され、そのローの最初の値に点滅するカーソルが表示されます。

2. 新しい値を入力し、Tab キーを押して次のカラムに移動します。

オプション	アクション
デフォルト値をカラムに挿入する	カラムがデフォルト値を受け入れる場合、セルには <b>(デフォルト)</b> が含まれます。デフォルト値を挿入する場合は、 <b>(デフォルト)</b> をクリックします。同様に、NULL 値を受け入れるカラムである場合、セルには <b>(NULL)</b> が表示されます。NULL 値を受け入れず、デフォルト値も持たないカラムである場合は、値を入力する必要があります。
値を計算カラムに挿入する	結果セット内に計算カラムがあり、計算カラムに値を指定しない場合、値はデータベースが更新されたときに計算されます。しかし、計算カラムに値を指定した場合は、指定した値でデータベースが更新され、値は計算カラムに対して計算されません。

すべてのカラム値が追加されるまで、この手順を繰り返します。

3. Enter キーを押してデータベースを更新します。
4. COMMIT 文を実行し、データベースの変更を確定します。

## 結果

これで新しいローがデータベースに挿入されました。

## 関連情報

[input\\_format オプション \[Interactive SQL\] \[1011 ページ\]](#)

### 1.7.1.3.4 Interactive SQL の結果セットからローを削除する

InteractiveSQL で結果セットからローを削除できます。

## 前提条件

ローを削除するには、テーブルの DELETE 権限が必要です。

結果セットを取得するにはクエリの実行を完了している必要があります。

## 手順

1. Interactive SQL で、削除したいロー (複数可) を選択します。ローを選択するには、次の手順に従います。
  - Shift キーを押しながらローをクリックします。
  - Shift + 上矢印または Shift + 下矢印キーを押します。

連続しないローを削除するには、各ローを個別に削除する必要があります。リストで複数の disjoint ローを選択するには、Control キーを押しながらクリックします。

2. Delete キーを押します。

選択したローがデータベーステーブルから削除されます。

3. COMMIT 文を実行し、変更を永続的なものにします。

## 結果

ローが結果セットから削除されます。

## 1.7.1.3.5 Interactive SQL 結果セットからロー、カラム、セルをコピーする

Interactive SQL に表示される結果セットからセル、ロー、カラムをコピーし、他のアプリケーションに貼り付けます。

### コンテキスト

ローやカラムをコピーすると、カラム見出しとテーブルデータもクリップボードにコピーされます。

コピーされたデータは、次の Interactive SQL オプションに従ってフォーマットされます。

- isql\_escape\_character オプション [Interactive SQL]
- isql\_field\_separator オプション [Interactive SQL]
- isql\_quote オプション [Interactive SQL]

これらのオプションは、Interactive SQL で **▶ オプション ▶ インポート/エクスポート** をクリックして変更することもできます。

これらのオプションがデフォルトに設定されている場合、コピーしたデータはカンマで区切られ、文字列は一重引用符で囲まれます。

### 手順

Interactive SQL で、コピーしたい結果セットを取得します。

オプション	アクション
1 ローのコピー	ローの任意のセルを右クリックし、 <b>▶ データのコピー ▶ ロー</b> をクリックします。
複数ローのコピー	Ctrl キーを押しながらローのセルをクリックし、さらに右クリックして <b>▶ データのコピー ▶ ロー</b> をクリックします。 選択したローが、カラム見出しを含めてクリップボードにコピーされます。
1 カラムのコピー	カラムの任意のセルを右クリックし、 <b>▶ データのコピー ▶ カラム</b> をクリックします。
複数カラムのコピー	Ctrl キーを押しながらカラムのセルをクリックし、さらに右クリックして <b>▶ データのコピー ▶ カラム</b> をクリックします。 結果ウィンドウ枠に結果セット全体が含まれていない場合、選択する前に残りの結果をフェッチするよう要求されます。残りをフェッチしないと、その時点でフェッチされている結果のみが選択されます。
セルのコピー	コピーするセルを右クリックして、 <b>▶ データのコピー ▶ セル</b> をクリックします。
複数セルのコピー	Ctrl キーを押しながらセルをクリックし、さらに右クリックして <b>▶ データのコピー ▶ セル</b> をクリックします。

オプション	アクション
	この操作を行った場合、カラム見出しはコピーされません。データのみがクリップボードにコピーされます。引用もされません。

## 結果

選択したデータがコピーされます。

## 次のステップ

選択した情報を、他のアプリケーションに貼り付けることができます。

## 関連情報

[isql\\_escape\\_character オプション \[Interactive SQL\] \[1014 ページ\]](#)

[isql\\_field\\_separator オプション \[Interactive SQL\] \[1015 ページ\]](#)

[isql\\_quote オプション \[Interactive SQL\] \[1018 ページ\]](#)

### 1.7.1.3.6 Interactive SQL の結果セットから SQL 文を生成する

結果セット内で選択したローに対する INSERT 文、DELETE 文、UPDATE 文を作成します。

## 前提条件

タスクを実行する結果セットを取得するには、クエリの実行を完了している必要があります。

## 手順

1. 文を生成するローを選択します。
2. 選択したローを右クリックし、**生成**をクリックしてから、**INSERT 文**、**DELETE 文**、または **UPDATE 文** をクリックします。

## 結果

文がクリップボードにコピーされます。

## 次のステップ

生成した文を実行します。INSERT、DELETE、または UPDATE 文を実行する権限が必要です。

### 1.7.1.3.7 イメージの表示 (Interactive SQL)

Interactive SQL を使用して、SVG、BLOB などのイメージや、5000000 ピクセル未満のイメージを表示します。

## 前提条件

クエリの実行に必要な権限が必要です。

イメージは次の形式のいずれかになっている必要があります。BMP (Windows 95 バージョン)、GIF、JPG、PNG、SVG、または WBMP (ワイヤレスビットマップ)。

## 手順

1. Interactive SQL からデータベースに接続します。
2. イメージが含まれる結果セットを返すクエリを実行します。
3. 結果セット内で、(イメージ) を含むセルをクリックし、... をクリックして、**ウィンドウで表示**をクリックします。

SVG を表示する場合は、次のショートカットを使用します。

アクション	オプション
Ctrl キーを押しながらマウスの左ボタンでドラッグ	視点を調整できるように境界をドラッグします。
Ctrl キーを押しながらマウスの右ボタンでドラッグ	中心を軸にイメージを回転させます。
Shift キーを押しながらマウスの左ボタンでドラッグ	イメージを新しいロケーションにパンします。
Shift キーを押しながらマウスの右ボタンでドラッグ	イメージをズームイン、ズームアウトします。

カラム `column-name` の値ウィンドウが表示されます。

4. **OK** をクリックして、ウィンドウを閉じます。

## 結果

イメージが表示されます。

### 例

サンプルデータベースに接続し、次の文を実行して JPEG イメージを返します。

```
SELECT * FROM GROUPO.Products;
```

コットン製のキャップに関連付けられた JPEG イメージをクリックすると、次のウィンドウが表示されます。



サンプルデータベースに接続し、次の文を実行すると、SVG イメージが返されます。

```
SELECT '<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect width="99" height="99"
style="fill:rgb(0,0,255);stroke-width:1;
stroke:rgb(0,0,0)"/>
</svg>';
```

### 1.7.1.3.8 HTML データと XML データの表示 (Interactive SQL の場合)

Interactive SQL で、クエリの結果セットから HTML および XML データを別ウィンドウに表示します。

#### 前提条件

クエリの実行に必要な権限が必要です。

## 手順

1. Interactive SQL で、データベースに接続します。
2. HTML データまたは XML データが含まれる結果セットを返すクエリを実行します。
3. 結果内で、HTML コンテンツまたは XML コンテンツを含むセルをクリックし、... をクリックして、**ウィンドウで表示**をクリックします。

## 結果

コラム `column-name` の値ウィンドウが表示されます。

### 例

HTML が含まれるサンプルの結果セットを表示するには、次のクエリを実行します。

```
SELECT '<html>
<head>
  <meta http-equiv=Content-Type content="text/html; charset=windows-1252">
  <title>Tee Shirt</title>
</head>
<body lang=EN-US>
  <p>
    <span style=font-size:10.0pt;font-family:Arial>
      We have improved the design of this perennial favorite.
      A sleek and technical shirt built for the trail, track, or
      sidewalk. UPF rating of 50+.
    </span>
  </p>
</body>
</html>';
```

適切な形式の XML ドキュメントが含まれるサンプルの結果セットを表示するには、次のクエリを実行します。

```
SELECT '<?xml version="1.0" encoding="UTF-8"?>
<parent>
  <children>
    <child name="Tim" />
    <child name="Katie" />
  </children>
</parent>';
```

## 次のステップ

**OK** をクリックして、ウィンドウを閉じます。

## 1.7.1.4 Interactive SQL の SQL 文

SQL 文の中には、Interactive SQL でのみサポートされるものもあります。

- CLEAR 文
- CONFIGURE 文
- CONNECT 文
- DESCRIBE 文
- DISCONNECT 文
- EXIT 文
- HELP 文
- INPUT 文
- OUTPUT 文
- PARAMETERS 文
- READ 文
- SET CONNECTION 文
- SET OPTION 文
- START SERVER 文
- START LOGGING 文
- STOP LOGGING 文
- SYSTEM 文

## 1.7.1.5 詳細とノードの統計を含むグラフィカルなプランの作成 (Interactive SQL)

プランビューアは、グラフィカルなプランと Ultra Light データベースのテキストプランを表示するためのグラフィカルツールです。

### 手順

1. Interactive SQL を開き、データベースに接続します。
2. **▶ ツール ▶ プランビューアを開く** をクリックするか、Shift + F5 キーを押します。

別ウィンドウにプランビューアが表示されます。プランビューアのウィンドウは、次のウィンドウ枠に分かれています。

#### SQL ウィンドウ枠

このウィンドウ枠には、プランを生成する SQL 文を入力します。

#### 【結果】ウィンドウ枠

このウィンドウ枠にはグラフィカルなプランが表示され、SQL Anywhere データベース専用です。

#### 【詳細】ウィンドウ枠

このウィンドウ枠には、SQL Anywhere データベースのプランに関する詳細テキストが表示されます。

Ultra Light データベースの場合、このウィンドウ枠にはテキストプランが表示されます。

3. [SQL](#) ウィンドウ枠に文を入力します。
4. [統計レベルリスト](#)で、[詳細とノードの統計](#)をクリックします。
5. [プランの取得](#)をクリックして、指定したクエリのプランを生成します。
6. [名前を付けて保存](#)をクリックします。
7. プランの保存先を指定し、ファイル名を入力します。[保存](#)をクリックします。

## 結果

[詳細とノードの統計](#)でグラフィカルなプランが作成されます。

## 次のステップ

▶ [ツール](#) ▶ [プランビューア](#) をクリックしてグラフィカルなプランを開きます。[開く](#)をクリックします。プランファイル (.saplan) を選択し、[開く](#)をクリックします。

このセクションの内容:

[グラフィカルなプランのカスタマイズ \[997 ページ\]](#)

グラフィカルなプラン内の項目の表示をカスタマイズします。

### 1.7.1.5.1 グラフィカルなプランのカスタマイズ

グラフィカルなプラン内の項目の表示をカスタマイズします。

## 手順

1. プランビューアを開きます。Interactive SQL で、▶ [ツール](#) ▶ [プランビューア](#) をクリックします。
2. プランが開きます。プランビューアの左下のウィンドウ枠でプランを右クリックし、[カスタマイズ](#)をクリックします。
3. 設定を変更します。
4. 変更が完了したら、[OK](#) をクリックします。
5. [プランの取得](#)をクリックします。

## 結果

カスタマイズされたグラフィカルなプランが生成されます。

### 1.7.1.6 Interactive SQL のカスタマイズ

結果セットの表示方法を設定し、警告メッセージを無効にし、さらに .sql ファイルのデフォルトエディタとして、Interactive SQL を設定します。

## コンテキスト

Interactive SQL で、**オプション**ウィンドウを使用してタブやウィンドウ枠の設定を変更します。Interactive SQL を配備するときに、**オプション**ウィンドウで表示または有効化する機能を制御できます。

## 手順

Interactive SQL を開き、データベースに接続します。

オプション	アクション
結果セットの表示方法のカスタマイズ	<ol style="list-style-type: none"><li>結果をスクロール可能なテーブルとして表示するか、またはテキストとして表示するかを選択します。 <b>データ</b>をクリックし、次のいずれかを選択します。  結果をスクロール可能なテーブルとして表示  このフォーマットの結果セットは編集できます。これはデフォルトです。 結果をテキストとして表示  このオプションを選択すると、結果セットは等幅フォントのテキストで表示されます。このフォーマットの結果セットは編集できません。</li><li>クエリを実行します。 テーブル編集の変更を反映させるには、新しいクエリを実行する必要があります。</li><li>結果セット内を右クリックし、次のいずれかを選択します。  カラムのサイズをデータに合わせる  値がトランケートされないように、テーブルカラムが広がります。 カラムのサイズをウィンドウに合わせる  すべてのカラムが水平スクロールバーのないウィンドウ内に収まるように、テーブルカラムは同じ幅になります。</li></ol>
警告メッセージの無効化	Interactive SQL に表示される警告メッセージの一部を無効にすることができます。たとえば、 <b>SQL 文</b> ウィンドウ枠でテキストを保存しないで、 <b>▶ ファイル ▶ 終了</b> をクリックした場合に表示される警告を非表示にできます。

オプション	アクション
	<ol style="list-style-type: none"> <li>1. <b>ツール &gt; オプション &gt; メッセージ</b> をクリックします。</li> <li>2. オプションのメッセージリストに表示されているメッセージをクリアします。</li> </ol>
Interactive SQL を .sql ファイルのデフォルトエディタに設定する	<ol style="list-style-type: none"> <li>1. <b>ツール &gt; オプション &gt; 一般</b> をクリックします。</li> <li>2. <i>Interactive SQL</i> を .SQL ファイルとプランファイルのデフォルトエディタにするをクリックします。</li> <li>3. <i>OK</i> をクリックします。</li> </ol>
実行ツールバーボタンを設定し、すべての文または選択した文を実行する	<ol style="list-style-type: none"> <li>1. <b>ツール &gt; オプション &gt; ツールバー</b> をクリックします。</li> <li>2. 次のオプションのうちの 1 つを選択します。 <ul style="list-style-type: none"> <li>○ すべての SQL 文を実行する場合は、<b>実行</b> をクリックします。これはデフォルト設定です。</li> <li>○ 選択した SQL 文のみを実行する場合は、<b>選択の実行</b> をクリックします。</li> </ul> </li> </ol>

このセクションの内容:

#### [Interactive SQL の履歴を表示する \[1000 ページ\]](#)

SQL 文の実行時に Interactive SQL によって自動的に保存される SQL 文の履歴を表示します。

#### [テーブル、カラム、プロシージャの検索 \(Interactive SQL\) \[1002 ページ\]](#)

Interactive SQL でテーブル、カラム、プロシージャを検索します。

#### [Interactive SQL オプション \[1003 ページ\]](#)

SET OPTION 文は、Interactive SQL のオプションの値を変更するために使用します。

#### [キーボードショートカット \(Interactive SQL\) \[1025 ページ\]](#)

Interactive SQL には、以下のキーボードショートカットがあります。

#### [テキスト補完 \(Interactive SQL および SQL Central\) \[1029 ページ\]](#)

Interactive SQL と SQL Central では、入力時に完全なキーワードとオブジェクト名を供給可能なテキスト補完オプションが提供されます。

#### [文のロギング \(Interactive SQL\) \[1031 ページ\]](#)

Interactive SQL で文をロギングします。

#### [スキーマロックの自動解放 \(Interactive SQL\) \[1032 ページ\]](#)

Interactive SQL を設定して、コミットされていない変更がない場合にデータベーススキーマロックを自動的に解放します。

#### [ソース制御の統合 \(Interactive SQL\) \[1033 ページ\]](#)

Interactive SQL はサードパーティのソース制御システムと統合でき、Interactive SQL 内からファイルに対する一般的なソース制御操作を実行できます。

## 関連情報

[データを表示した場合の予期しない記号のトラブルシューティング \[583 ページ\]](#)

## 1.7.1.6.1 Interactive SQL の履歴を表示する

SQL 文の実行時に Interactive SQL によって自動的に保存される SQL 文の履歴を表示します。

### コンテキスト

CREATE USER、CONNECT、CREATE EXTERNLOGIN などのパスワード情報を含む SQL 文を実行した場合、パスワードは常に履歴で難読化されます。

文の履歴がその後の Interactive SQL セッションで表示されるときには、パスワードの代わりに "..." が表示されます。

### 手順

1. SQL 文を実行します。
2. 履歴タブで、表示ドロップダウンメニューを使用して、このタブで実行された SQL、デモで実行された SQL、またはすべてのデータベースで実行された SQL のどれから文を表示するかを選択します。
3. 履歴タブで、文を選択し、次のいずれかのオプションを選択します。

オプション	アクション
SQL の挿入	文を右クリックし、▶ 送信先 ▶ SQL エディタ ▶ をクリックします。
履歴を保存する	文を右クリックし、コンテキストメニューで [履歴を保存] をクリックします。
お気に入りリストに文を追加する	文を右クリックし、▶ 送信先 ▶ SQL エディタ ▶ をクリックします。
文を削除する	<ul style="list-style-type: none"><li>○ 文を右クリックし、▶ 削除 ▶ 選択した文 ▶ をクリックします。</li><li>○ 履歴を削除するには、文を右クリックし、▶ 削除 ▶ 履歴 ▶ をクリックします。</li></ul>

このセクションの内容:

[お気に入りリストの管理 \[1001 ページ\]](#)

お気に入りのリストを作成、編集、共有します。

## 1.7.1.6.1.1 お気に入りリストの管理

お気に入りのリストを作成、編集、共有します。

### コンテキスト

お気に入りリストは各ユーザに固有のもので、共有しないかぎり別のユーザからは見えません。お気に入りリストをエクスポートすると、.fav ファイルとして保存されます。

### 手順

次のオプションのうちの 1 つを選択してください。

オプション	アクション
ファイル、文、接続をお気に入りリストに追加する	<ol style="list-style-type: none"><li>お気に入り追加する SQL スクリプトファイルを開きます。</li><li>▶ お気に入り ▶ お気に入り追加 をクリックします。</li><li>開いているファイル 'filename' を追加をクリックします。</li><li>名前フィールドで、.sql ファイルの名前を指定します。</li></ol>
SQL 文をお気に入り追加する	<ol style="list-style-type: none"><li>▶ お気に入り ▶ お気に入り追加 をクリックします。</li><li>SQL 文の追加をクリックし、名前フィールドでお気に入りの名前を指定します。</li></ol>
接続をお気に入り追加する	<ol style="list-style-type: none"><li>▶ お気に入り ▶ お気に入り追加 をクリックします。</li><li>接続パスワードの保存をクリックし、名前フィールドでお気に入りの名前を指定します。</li></ol>
ファイルの名前を編集する	<ol style="list-style-type: none"><li>▶ お気に入り ▶ お気に入り表示 をクリックします。 Interactive SQL ウィンドウの左側に、お気に入りウィンドウ枠が表示されます。</li><li>お気に入りを右クリックし、編集をクリックします。</li><li>文を編集し、保存をクリックします。</li></ol>
お気に入りリストの共有	<p>お気に入りをクリックし、次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"><li>お気に入りリストをエクスポートするには、次の手順に従います。<ol style="list-style-type: none"><li>お気に入りのエクスポートをクリックします。</li><li>.fav ファイルの名前を指定し、エクスポートをクリックします。</li></ol></li><li>お気に入りリストをインポートするには、次の手順に従います。<ol style="list-style-type: none"><li>お気に入りのインポートをクリックします。</li><li>インポートするお気に入りリストを参照し、インポートをクリックします。</li></ol></li></ul>

## 1.7.1.6.2 テーブル、カラム、プロシージャの検索 (Interactive SQL)

Interactive SQL でテーブル、カラム、プロシージャを検索します。

### 手順

1. Interactive SQL で、**ツール**をクリックし、次のいずれかを選択します。

オプション	アクション
テーブル名のルックアップ	テーブル名のルックアップをクリックするか、F7 キーを押します。 テーブルを検索し、選択します。
カラム名のルックアップ	テーブル名のルックアップをクリックするか、F7 キーを押します。 カラムを含むテーブルを検索し選択します。 カラムを表示をクリックします。
プロシージャ名のルックアップ	プロシージャ名のルックアップをクリックするか、F8 キーを押します。 プロシージャを検索し、選択します。

#### ➔ ヒント

- テーブル名またはプロシージャ名の最初の数文字を入力すると、項目のリストを絞り込みます。
- SQL のワイルドカード文字 % (パーセント記号) を使用するとゼロまたは複数文字の任意の文字列にマッチし、'\_ '(アンダースコア) を使用すると任意の 1 文字にマッチします。
- テーブル名の中にワイルドカード文字を挿入して検索するには、ワイルドカード文字の前にエスケープ文字を付けます。SQL Anywhere ODBC ドライバと SQL Anywhere JDBC ドライバを使用している場合は、~ (チルダ) をエスケープ文字として使用します。
- プロシージャの名前をルックアップしているが、目的のプロシージャであるかどうかははっきりしない場合、[詳細] ボタンをクリックしてプロシージャのソースを確認します。

2. **OK** をクリックして、名前を **SQL 文** ウィンドウ枠の現在のカーソル位置に挿入します。

### 結果

テーブル名、カラム名、またはプロシージャ名が検索され、**SQL 文** ウィンドウ枠に挿入されます。

### 関連情報

[テキスト補完 \(Interactive SQL および SQL Central\) \[1029 ページ\]](#)

### 1.7.1.6.3 Interactive SQL オプション

SET OPTION 文は、Interactive SQL のオプションの値を変更するために使用します。

オプション	値	デフォルト
auto_commit オプション [Interactive SQL]	On、Off	Off
auto_refetch オプション [Interactive SQL]	On、Off	On
bell オプション [Interactive SQL]	On、Off	On
command_delimiter オプション [Interactive SQL]	文字列	';
commit_on_exit オプション [Interactive SQL]	On、Off	On
default_isql_encoding オプション [Interactive SQL]	文字列	空の文字列
echo オプション [Interactive SQL]	On、Off	On
input_format オプション [Interactive SQL]	TEXT、EXCEL、FIXED	TEXT
isql_allow_read_client_file オプション [Interactive SQL]	On、Off、Prompt	Prompt
isql_allow_write_client_file オプション [Interactive SQL]	On、Off、Prompt	Prompt
isql_command_timing オプション [Interactive SQL]	On、Off	On
isql_escape_character オプション [Interactive SQL]	文字	'¥'
isql_field_separator オプション [Interactive SQL]	文字列	','
isql_maximum_displayed_rows オプション [Interactive SQL]	All または負でない整数	500
isql_print_result_set オプション [Interactive SQL]	Last、All、None	Last
isql_quote オプション [Interactive SQL]	文字列	'
isql_show_multiple_result_sets オプション [Interactive SQL]	On、Off	Off
nulls オプション [Interactive SQL]	文字列	'(NULL)'
on_error オプション [Interactive SQL]	Stop、Continue、Prompt、Exit、Notify_Continue、Notify_Stop、Notify_Exit	Prompt
output_format オプション [Interactive SQL]	TEXT、EXCEL、FIXED、HTML、SQL、XML	TEXT

オプション	値	デフォルト
output_length オプション [Interactive SQL]	整数	0
output_nulls オプション [Interactive SQL]	文字列	空の文字列
truncation_length オプション [Interactive SQL]	整数	256

このセクションの内容:

[auto\\_commit オプション \[Interactive SQL\] \[1005 ページ\]](#)

各文の後に COMMIT が実行されるかどうかを制御します。

[auto\\_refetch オプション \[Interactive SQL\] \[1006 ページ\]](#)

削除、更新、または挿入後にクエリ結果が再度フェッチされるかどうかを制御します。

[bell オプション \[Interactive SQL\] \[1007 ページ\]](#)

エラーが起こったときにベルを鳴らすかどうかを制御します。

[command\\_delimiter オプション \[Interactive SQL\] \[1007 ページ\]](#)

Interactive SQL で文の終わりを示す文字列を設定します。

[commit\\_on\\_exit オプション \[Interactive SQL\] \[1009 ページ\]](#)

Interactive SQL が切断または停止されるときに動作を制御します。

[default\\_isql\\_encoding オプション \[Interactive SQL\] \[1009 ページ\]](#)

READ、INPUT、OUTPUT 文で使用されるエンコードを指定します。

[echo オプション \[Interactive SQL\] \[1011 ページ\]](#)

START LOGGING 文を使用してロギングが有効になっている場合に、SQL 文およびメッセージのファイルへのロギングに対する詳細な制御を許可します。

[input\\_format オプション \[Interactive SQL\] \[1011 ページ\]](#)

INPUT 文が予想するデフォルトデータフォーマットを設定します。

[isql\\_allow\\_read\\_client\\_file オプション \[Interactive SQL\] \[1012 ページ\]](#)

接続でのクライアントファイルの読み込みを許可するかどうかを制御します。

[isql\\_allow\\_write\\_client\\_file オプション \[Interactive SQL\] \[1013 ページ\]](#)

接続でのクライアントファイルの書き込みを許可するかどうかを制御します。

[isql\\_command\\_timing オプション \[Interactive SQL\] \[1014 ページ\]](#)

Interactive SQL が対話型モードでコンソールアプリケーションとして実行されている場合に、実行時間を出力するかどうかを制御します。

[isql\\_escape\\_character オプション \[Interactive SQL\] \[1014 ページ\]](#)

テキストファイルにエクスポートされたデータに印刷不能な文字が含まれていた場合に、その代わりに使用するエスケープ文字を制御します。

[isql\\_field\\_separator オプション \[Interactive SQL\] \[1015 ページ\]](#)

ファイルにエクスポートするデータの値を区切るのに使用される、デフォルトの文字列を制御します。

[isql\\_maximum\\_displayed\\_rows オプション \[Interactive SQL\] \[1017 ページ\]](#)

Interactive SQL で結果ウィンドウ枠に表示されるローの最大数を指定します。

#### [isql\\_print\\_result\\_set オプション \[Interactive SQL\] \[1017 ページ\]](#)

.sql ファイルを実行したときに出力される結果セットを指定します。

#### [isql\\_quote オプション \[Interactive SQL\] \[1018 ページ\]](#)

テキストファイルにエクスポートするデータの文字列すべての先頭と最後に付く、デフォルトの文字列を制御します。

#### [isql\\_show\\_multiple\\_result\\_sets オプション \[Interactive SQL\] \[1019 ページ\]](#)

Interactive SQL で結果 ウィンドウ枠に複数の結果セットを表示可能にするかどうかを指定します。

#### [nulls オプション \[Interactive SQL\] \[1020 ページ\]](#)

Interactive SQL で結果を表示するときにデータベース内の NULL 値をどのように表示するかを指定します。

#### [on\\_error オプション \[Interactive SQL\] \[1021 ページ\]](#)

Interactive SQL の文を実行中にエラーが起こった場合の動作を制御します。

#### [output\\_format オプション \[Interactive SQL\] \[1022 ページ\]](#)

ファイルにリダイレクトされる SELECT 文で検索したデータや、OUTPUT 文を使用した出力のデフォルトの出力フォーマットを設定します。

#### [output\\_length オプション \[Interactive SQL\] \[1023 ページ\]](#)

Interactive SQL が、外部ファイルに情報をエクスポートするときに使用するカラム値の長さを制御します。

#### [output\\_nulls オプション \[Interactive SQL\] \[1023 ページ\]](#)

NULL 値をどのようにエクスポートするかを制御します。

#### [truncation\\_length オプション \[Interactive SQL\] \[1024 ページ\]](#)

表示内容を画面内に収めるために、幅の広いカラムのトランケーションを制御します。

### 1.7.1.6.3.1 auto\_commit オプション [Interactive SQL]

各文の後に COMMIT が実行されるかどうかを制御します。

#### 指定可能な値

On、Off

#### デフォルト

Off

#### 備考

auto\_commit が On の場合、各文の処理が成功した後でデータベースの COMMIT が実行されます。

デフォルトでは、COMMIT または ROLLBACK が実行されるのは、SQL 文が自動コミットを実行する場合 (CREATE TABLE 文など) を除き、ユーザが COMMIT 文または ROLLBACK 文を発行したときのみです。

#### **i** 注記

Interactive SQL の auto\_commit オプションを設定することによって、データベースサーバの auto\_commit オプションは設定されません。

## 例

Interactive SQL による自動コミットを無効にするには、次の文を実行します。

```
set option auto_commit='Off';
```

Interactive SQL の auto\_commit オプションの現在の設定は、*SQL Anywhere* オプションの**実行**タブで確認できます。

## 関連情報

[commit\\_on\\_exit オプション \[Interactive SQL\] \[1009 ページ\]](#)

[キーボードショートカット \(Interactive SQL\) \[1025 ページ\]](#)

### 1.7.1.6.3.2 auto\_refetch オプション [Interactive SQL]

削除、更新、または挿入後にクエリ結果が再度フェッチされるかどうかを制御します。

#### 指定可能な値

On、Off

#### デフォルト

On

## 備考

auto\_refetch が On の場合、Interactive SQL の結果ウィンドウ枠の結果タブに表示される現在のクエリ結果は、INSERT 文、UPDATE 文、または DELETE 文の後でデータベースから再フェッチされます。クエリの複雑さによって、時間のかかるものもあります。このため、これをオフにすることもできます。

### 1.7.1.6.3.3 bell オプション [Interactive SQL]

エラーが起こったときにベルを鳴らすかどうかを制御します。

#### 指定可能な値

On、Off

#### デフォルト

On

## 備考

このオプションは好みに応じて設定します。

### 1.7.1.6.3.4 command\_delimiter オプション [Interactive SQL]

Interactive SQL で文の終わりを示す文字列を設定します。

#### 指定可能な値

文字列

## デフォルト

セミコロン (;)

## 備考

通常は、コマンドデリミタを変更する必要はありません。セミコロンのままにしてください。

文のデリミタとしてセミコロンなどの文字列を使用する代わりに、セパレータ `go` を独立した行の先頭に配置することもできます。

独立した行の先頭に指定された `go` は、`command_delimiter` オプションの値にかかわらず常にコマンドデリミタとして認識されます。

`command_delimiter` オプションには任意の文字列を指定できます。ただし、次のような制限があります。

- & (アンパサンド)、\* (アスタリスク)、@ (アットマーク)、: (コロン)、. (ピリオド)、= (等号)、( (左カッコ)、) (右カッコ)、または | (パイプ記号) が含まれているデリミタには、それ以外の文字を追加することはできません。たとえば、\* は有効なデリミタですが、\*\* はデリミタとして無効です。
- 既存のキーワードをコマンドセパレータとして使用しないでください。
- コマンドデリミタには、数字、文字、句読点などの任意の文字列を使用できますが、埋め込みブランクを含めることはできません。  
コマンドデリミタとして設定されている文字列が、識別子として有効な文字で始まる場合は、前にスペースを付けてください。コマンドデリミタでは、大文字と小文字が区別されます。SET OPTION 文では、一重引用符で新しいコマンドデリミタを囲む必要があります。コマンドデリミタがセミコロン (デフォルト) の場合、セミコロンの前にスペースを入れる必要はありません。

### 例

次の例は、コマンドデリミタをチルダに設定します。

```
SET OPTION command_delimiter='~';  
MESSAGE 'hello'~
```

Interactive SQL の `-d` オプションを使用してコマンドデリミタを設定することもできます。この場合、`.sql` ファイルに SET OPTION `command_delimiter` 文を記述する必要はありません。たとえば、スクリプトファイル `test.sql` で、チルダ (`~`) をコマンドデリミタとして使用する場合は、次のようになります。

```
dbisql -d "~" test.sql
```

## 関連情報

[Interactive SQL ユーティリティ \(dbisql\) \[1125 ページ\]](#)

### 1.7.1.6.3.5 commit\_on\_exit オプション [Interactive SQL]

Interactive SQL が切断または停止されるときに動作を制御します。

#### 指定可能な値

On、Off

#### デフォルト

On

#### 備考

Interactive SQL を終了するときに COMMIT か ROLLBACK を行うかを制御します。commit\_on\_exit を On に設定すると、COMMIT が行われます。

#### 関連情報

[キーボードショートカット \(Interactive SQL\) \[1025 ページ\]](#)

### 1.7.1.6.3.6 default\_isql\_encoding オプション [Interactive SQL]

READ、INPUT、OUTPUT 文で使用されるエンコードを指定します。

#### 指定可能な値

識別子または文字列

#### デフォルト

システムエンコード (空の文字列) を使用します。

## スコープ

現在の接続の間、テンポラリオプションとしてのみ設定できます。

## 備考

このオプションは、ファイルを読み込みまたは書き込みするときに使用するエンコードを指定するのに使用されます。これを永続的に設定することはできません。デフォルトのエンコードは、実行しているプラットフォームのコードページです。英語版 Windows コンピュータでは、デフォルトのコードページは 1252 です。

Interactive SQL を実行している場合、INPUT 文、OUTPUT 文、または READ 文で使用されるエンコードは、次の順序で決定されます。

- INPUT 文、OUTPUT 文、または READ 文の ENCODING 句に指定されたエンコード (この句が指定されている場合)。
- default\_isql\_encoding オプションで指定されたエンコード (このオプションが設定されている場合)。
- 実行しているプラットフォームのデフォルトエンコード。英語版 Windows コンピュータでは、デフォルトエンコードは 1252 です。

OUTPUT 文を使用して入力ファイルを作成したときにエンコードを指定した場合は、INPUT 文でも同じ ENCODING 句を指定してください。

### 例

エンコーディングを UTF-16 (Unicode ファイルの読み込み用) に設定します。

```
SET TEMPORARY OPTION default_isql_encoding = 'UTF-16';
```

エンコードを cp437 (英語版 Windows OEM) に設定します。

```
SET TEMPORARY OPTION default_isql_encoding = 'cp437';
```

## 関連情報

[文字セット \[576 ページ\]](#)

[国際言語と文字セット \[571 ページ\]](#)

### 1.7.1.6.3.7 echo オプション [Interactive SQL]

START LOGGING 文を使用してロギングが有効になっている場合に、SQL 文およびメッセージのファイルへのロギングに対する詳細な制御を許可します。

#### 指定可能な値

On、Off

#### デフォルト

On

#### 備考

このオプションは、READ 文を使用して SQL スクリプトファイルを実行する場合、または Interactive SQL で **ファイル** > **スクリプトの実行** をクリックしてスクリプトファイルを実行する場合に最適です。このオプションを有効にするには、ロギングを有効にする必要があります。

### 1.7.1.6.3.8 input\_format オプション [Interactive SQL]

INPUT 文が予期するデフォルトデータフォーマットを設定します。

#### 指定可能な値

##### TEXT

入力行はテキスト文字であり、1 行あたり 1 つのローで構成され、値はカンマまたはセミコロンで区切られているものと見なされます。アルファベットの文字列をアポストロフィ (一重引用符) または二重引用符で囲むことができます。

デフォルトのデリミタは、小数点にピリオドを使用するロケールではカンマ (,) です。小数点にカンマを使用するロケールでは、デフォルトのデリミタはセミコロン (;) です。カンマまたはセミコロンを含む項目デリミタは、一重引用符または二重引用符のどちらかで囲む必要があります。一重か二重引用符を使用している場合、文字列内で使用するには引用符を 2 つ重ねてください。

##### EXCEL

シートの最初のローには、カラム名が含まれると見なされます。

FIXED

入力行は固定長フォーマットです。

デフォルト

TEXT

### 1.7.1.6.3.9 isql\_allow\_read\_client\_file オプション [Interactive SQL]

接続でのクライアントファイルの読み込みを許可するかどうかを制御します。

指定可能な値

On、Off、Prompt

デフォルト

Prompt

備考

このオプションは、データベースサーバがクライアントコンピュータ上のファイルを読み込めるかどうかを制御します。On に設定すると、読み込みが許可されます。Off に設定すると、読み込みは許可されません。Prompt に設定すると、プロンプトが表示され、ユーザが動作を指定できます。

このオプションは接続単位で保存され、その接続の間だけ持続します。このオプションは、SET TEMPORARY OPTION 文を使用して設定します。TEMPORARY キーワードを省略すると、Interactive SQL からエラーがレポートされます。

このオプションを使用すると、ストアードプロシージャまたはトリガから LOAD TABLE が実行された場合に、ユーザの介入なしにデータファイルを読み込みます。

クライアントコンピュータでファイルを読み込むには、READ CLIENT FILE システム権限が必要です。

## 関連情報

[allow\\_read\\_client\\_file オプション \[642 ページ\]](#)

[allow\\_write\\_client\\_file オプション \[645 ページ\]](#)

[isql\\_allow\\_write\\_client\\_file オプション \[Interactive SQL\] \[1013 ページ\]](#)

### 1.7.1.6.3.10 isql\_allow\_write\_client\_file オプション [Interactive SQL]

接続でのクライアントファイルの書き込みを許可するかどうかを制御します。

#### 指定可能な値

On、Off、Prompt

#### デフォルト

Prompt

#### 備考

このオプションは、データベースサーバからクライアントコンピュータ上のファイルに書き込めるかどうかを制御します。On に設定すると、書き込みが許可されます。Off に設定すると、書き込みは許可されません。Prompt に設定すると、プロンプトが表示され、ユーザが動作を指定できます。

このオプションは接続単位で保存され、その接続の間だけ持続します。このオプションは、SET TEMPORARY OPTION 文を使用して設定します。TEMPORARY キーワードを省略すると、Interactive SQL からエラーがレポートされます。

クライアントコンピュータでファイルに書き込むには、WRITE CLIENT FILE システム権限が必要です。

## 関連情報

[allow\\_write\\_client\\_file オプション \[645 ページ\]](#)

[allow\\_read\\_client\\_file オプション \[642 ページ\]](#)

[isql\\_allow\\_read\\_client\\_file オプション \[Interactive SQL\] \[1012 ページ\]](#)

### 1.7.1.6.3.11 isql\_command\_timing オプション [Interactive SQL]

Interactive SQL が対話型モードでコンソールアプリケーションとして実行されている場合に、実行時間を出力するかどうかを制御します。

#### 指定可能な値

On、Off

#### デフォルト

Off

#### 備考

このオプションは、Interactive SQL が非対話型 (バッチ) モードでウィンドウアプリケーションまたはコンソールアプリケーションとして実行される場合は適用されません。

### 1.7.1.6.3.12 isql\_escape\_character オプション [Interactive SQL]

テキストファイルにエクスポートされたデータに印刷不能な文字が含まれていた場合に、その代わりに使用するエスケープ文字を制御します。

#### 指定可能な値

任意の 1 文字

#### デフォルト

円記号 (¥)

## 備考

Interactive SQL が改行など印刷不能な文字を含む文字列をエクスポートする場合、印刷不能な文字は 16 進数フォーマットに変換され、その文字の前にはエスケープ文字が付加されます。OUTPUT 文に ESCAPE CHARACTER 句がない場合に、この設定で指定した文字が出力で使用されます。この設定は、テキストファイルにエクスポートする場合にだけ使用されます。

### 例

改行が埋め込まれた 1 つの文字列値を含む表を作成します (INSERT 文に "¥n" で示されます)。次に、# 記号をエスケープ文字として使用して、データを c:¥escape.txt にエクスポートします。

```
CREATE TABLE escape_test( text varchar(10 ) );
INSERT INTO escape_test VALUES( 'one¥ntwo' );
SET OPTION isql_escape_character='#';
SELECT * FROM escape_test;
OUTPUT TO c:¥escape.txt FORMAT TEXT;
```

このコードを実行すると、次のデータが escape.txt に書き込まれます。

```
'one#x0atwo'
```

シャープ記号 (#) はエスケープ文字であり、x0a は ¥n の 16 進表現です。

先頭と末尾の文字 (この場合は一重引用符) は、isql\_quote の設定によって異なります。

## 関連情報

[isql\\_quote オプション \[Interactive SQL\] \[1018 ページ\]](#)

### 1.7.1.6.3.13 isql\_field\_separator オプション [Interactive SQL]

ファイルにエクスポートするデータの値を区切るのに使用される、デフォルトの文字列を制御します。

#### 指定可能な値

文字列

#### デフォルト

ピリオドが小数点として使用されるロケールでは、デフォルトのフィールドセパレータはカンマ (,) です。カンマが小数点として使用されるロケールでは、デフォルトのフィールドセパレータはセミコロン (;) です。

## 備考

OUTPUT 文に DELIMITED BY 句が含まれていない場合は、この設定値が使用されます。

### 例

- 最初の例では、c:¥Employees.txt にエクスポートするデータのフィールドセパレータとしてコロンを指定します。

```
SET OPTION isql_field_separator=':';
SELECT Surname, GivenName FROM Employees WHERE EmployeeID < 150;
OUTPUT TO c:¥Employees.txt FORMAT TEXT;
```

このコードを実行すると、次のデータが Employees.txt に書き込まれます。

```
'Whitney': 'Fran'
'Cobb': 'Matthew'
'Chin': 'Philip'
'Jordan': 'Julie'
```

先頭と末尾の文字 (この場合は一重引用符) は、isql\_quote の設定によって異なります。

- 次の例では、c:¥Employees.txt にエクスポートするデータのフィールドセパレータとしてタブを指定します。

```
SET OPTION isql_field_separator='¥t';
SELECT Surname, GivenName FROM Employees WHERE EmployeeID < 150;
OUTPUT TO c:¥Employees.txt FORMAT TEXT;
```

このコードを実行すると、次のデータが Employees.txt に書き込まれます。

```
'Whitney' 'Fran'
'Cobb' 'Matthew'
'Chin' 'Philip'
'Jordan' 'Julie'
```

先頭と末尾の文字 (この場合は一重引用符) は、isql\_quote の設定によって異なります。エスケープ文字 (この場合はバックスラッシュ) は、isql\_escape\_character の設定によって異なります。

## 関連情報

[isql\\_quote オプション \[Interactive SQL\] \[1018 ページ\]](#)

[isql\\_escape\\_character オプション \[Interactive SQL\] \[1014 ページ\]](#)

### 1.7.1.6.3.14 isql\_maximum\_displayed\_rows オプション [Interactive SQL]

Interactive SQL で結果ウィンドウ枠に表示されるローの最大数を指定します。

#### 指定可能な値

ALL または負でない整数

#### デフォルト

500

#### 備考

このオプションでは、結果ウィンドウ枠に表示されるローの最大数を指定できます。このオプションの値は、Interactive SQL のオプションウィンドウでも設定できます。

#### 警告

表示する結果セットが大きい場合、Interactive SQL でメモリが不足することがあります。その場合、Interactive SQL から問題がレポートされ、結果セットは表示されません。

### 1.7.1.6.3.15 isql\_print\_result\_set オプション [Interactive SQL]

.sql ファイルを実行したときに出力される結果セットを指定します。

#### 指定可能な値

##### LAST

ファイル内の最後の文の結果セットを出力します。

##### ALL

結果セットを返すファイル内のすべての文の結果セットを出力します。

##### NONE

結果セットを出力しません。

## デフォルト

LAST

## 備考

isql\_print\_result\_set オプションは、Interactive SQL をコマンドラインプログラムとして実行する場合 (たとえば、.sql ファイルの実行時) のみ有効です。

このオプションでは、.sql ファイルを実行したときに出力する結果セットを指定できます。

### 1.7.1.6.3.16 isql\_quote オプション [Interactive SQL]

テキストファイルにエクスポートするデータの文字列すべての先頭と最後に付く、デフォルトの文字列を制御します。

## 指定可能な値

文字列

## デフォルト

一重引用符 (')

## 備考

テキストファイルにエクスポートするデータの文字列すべての先頭と最後に付く、デフォルトの文字列を制御します。OUTPUT 文に QUOTE 句が含まれていない場合は、この設定値がデフォルトで使用されます。

### 例

すべての文字列の先頭と最後に付くデフォルト文字列を二重引用符に変更するには、次のように指定します。

```
SET OPTION isql_quote='";  
SELECT Surname, _GivenName FROM Employees WHERE EmployeeID < 150;
```

```
OUTPUT TO c:¥Employees.txt FORMAT TEXT;
```

このコードを実行すると、次のデータが Employees.txt に書き込まれます。

```
"Whitney", "Fran"  
"Cobb", "Matthew"  
"Chin", "Philip"  
"Jordan", "Julie"
```

区切り文字 (この場合はカンマ) は、isql\_field\_separator の設定によって異なります。

## 関連情報

[isql\\_field\\_separator オプション \[Interactive SQL\] \[1015 ページ\]](#)

### 1.7.1.6.3.17 isql\_show\_multiple\_result\_sets オプション [Interactive SQL]

Interactive SQL で**結果** ウィンドウ枠に複数の結果セットを表示可能にするかどうかを指定します。

#### 指定可能な値

On、Off

#### デフォルト

Off

#### 備考

複数の SELECT 文を返すプロシージャを実行したときに複数の結果セットを**結果**ウィンドウ枠に表示するには、このオプションを ON に設定します。

各結果セットは、**結果**ウィンドウ枠の個別のタブに表示されます。デフォルトでは、Interactive SQL では複数の結果セットは表示されません。このオプションの設定は、Interactive SQL をコマンドラインプログラムとして実行する場合にも適用されません。

Interactive SQL をウィンドウベースのアプリケーションとして実行している場合、isql\_show\_multiple\_result\_sets オプションは、**すべての結果セットを表示**オプションとしても表示されます。

Interactive SQL を配備するときに、ユーザによる isql\_show\_multiple\_result\_sets オプションの設定を禁止できます。

#### 例

次の文を実行すると、isql\_show\_multiple\_result\_sets が On に設定されます。

```
SET OPTION isql_show_multiple_result_sets=On
```

プロシージャ ListPeople をテストし、Interactive SQL で複数の結果セットを表示するには、Interactive SQL で次の文を実行します。

```
SET OPTION isql_show_multiple_result_sets=On;  
CALL ListPeople ();
```

## 1.7.1.6.3.18 nulls オプション [Interactive SQL]

Interactive SQL で結果を表示するときにデータベース内の NULL 値をどのように表示するかを指定します。

### 指定可能な値

文字列

デフォルト

(NULL)

### 備考

このオプションは好みに応じて設定します。結果セットをファイルに保存する場合は、このオプションの値は適用されません。結果セットをファイルに保存するときに使用される値は、output\_nulls オプションで指定します。

### 関連情報

[output\\_nulls オプション \[Interactive SQL\] \[1023 ページ\]](#)

## 1.7.1.6.3.19 on\_error オプション [Interactive SQL]

Interactive SQL の文を実行中にエラーが起こった場合の動作を制御します。

### 指定可能な値

#### Stop

Interactive SQL が文の実行を停止します。

#### Prompt

Interactive SQL は、ユーザに続行するかどうかを確認するプロンプトを表示します。

#### Continue

エラーは Interactive SQL の結果エリアにある履歴タブに送信され、Interactive SQL は文の実行を続行します。

#### Exit

Interactive SQL が停止します。

#### Notify\_Continue

エラーがレポートされ、ユーザは続行するために Enter キーを押すか、OK をクリックするよう要求されます。

#### Notify\_Stop

エラーがレポートされ、ユーザは実行を停止するために Enter キーを押すか、OK をクリックするよう要求されます。

#### Notify\_Exit

エラーがレポートされ、ユーザは Interactive SQL を停止するために Enter キーを押すか、OK をクリックするよう要求されます。

### デフォルト

Prompt

### 備考

.sql ファイルを実行している場合は、STOP と EXIT のどちらに設定しても同じ結果になります。これらの値のいずれかを指定すると、Interactive SQL が停止します。

Interactive SQL ユーティリティ (dbisql) の onerror オプションが指定されている場合、このオプションは無視されます。

## 関連情報

[Interactive SQL ユーティリティ \(dbisql\) \[1125 ページ\]](#)

### 1.7.1.6.3.20 output\_format オプション [Interactive SQL]

ファイルにリダイレクトされる SELECT 文で検索したデータや、OUTPUT 文を使用した出力のデフォルトの出力フォーマットを設定します。

#### 指定可能な値

##### TEXT

出力は、ファイルの 1 行に 1 つのローが格納されたテキストフォーマットファイルです。デフォルトでは、値をフィールドセパレータで区切り、文字列をアポストロフィ (一重引用符) で囲みます。フィールドセパレータは、[isql\\_field\\_separator\\_option](#) によって決まります。

##### EXCEL

出力は Microsoft Excel ファイルで、カラム見出しはワークシートの最初の行に書き込まれます。

##### FIXED

出力は、それぞれのカラムが固定幅を持つ固定フォーマットです。このフォーマットでは、カラム見出しは出力されません。

##### HTML

出力は HTML フォーマットです。

##### SQL

出力は、テーブル内の情報を再作成するのに必要な Interactive SQL の INPUT 文です。

##### XML

この出力は、UTF-8 でエンコードされ、DTD が埋め込まれた XML ファイルです。バイナリ値は、2 桁の 16 進数文字列として表されるバイナリデータで CDATA ブロック内にエンコードされます。

## 備考

output\_format option は、OUTPUT 文で FORMAT 句が指定されない場合に使用されます。output\_format option は、OUTPUT 文に FORMAT 句が含まれる場合は無視されます。

---

デフォルト

TEXT

### 1.7.1.6.3.21 output\_length オプション [Interactive SQL]

Interactive SQL が、外部ファイルに情報をエクスポートするときに使用するカラム値の長さを制御します。

指定可能な値

負でない整数

デフォルト

0 (トランケーションなし)

備考

このオプションは、Interactive SQL が外部ファイルにデータをエクスポートするときに使用するカラム値の最大長を制御します (OUTPUT 文で出力リダイレクションを使用)。このオプションは、TEXT、HTML、SQL の出力フォーマットだけに影響します。

### 1.7.1.6.3.22 output\_nulls オプション [Interactive SQL]

NULL 値をどのようにエクスポートするかを制御します。

指定可能な値

文字列

## デフォルト

空の文字列

## 備考

このオプションは、NULL 値をどのように OUTPUT 文で記述するかを制御します。結果セットで NULL 値が見つかった場合、NULL 値の代わりにこのオプションで設定された文字列が返されます。このオプションは、TEXT、HTML、FIXED、SQL の出力フォーマットだけに影響します。

### 例

次の例では、NULL 値として表示される値を (*unknown*) に変更します。

```
SET OPTION output_nulls = '(unknown)';
```

## 1.7.1.6.3.23 truncation\_length オプション [Interactive SQL]

表示内容を画面内に収めるために、幅の広いカラムのトランケーションを制御します。

## 指定可能な値

整数

## デフォルト

256

## 備考

truncation\_length オプションは、表示されるカラムの長さを制限します。単位は文字です。値 0 はカラムがトランケートされないことを意味します。トランケーション長のデフォルト値は 256 です。

## 1.7.1.6.4 キーボードショートカット (Interactive SQL)

Interactive SQL には、以下のキーボードショートカットがあります。

キー	説明
Alt + F4	Interactive SQL を停止します。
Alt + 左矢印 Mac OS X では、Control-Command + 左矢印を使用します。	履歴リストにある直前の SQL 文を表示します。
Alt + 右矢印 Mac OS X では、Control-Command + 右矢印を使用します。	履歴リストにある次の SQL 文を表示します。
Ctrl + Backspace	カーソルの左の 1 ワードを削除します。
Ctrl + Break	実行中の SQL 文を中断します。このショートカットは、Windows オペレーティングシステムでのみサポートされます。
Ctrl + A	アクティブなウィンドウ枠のすべてのテキストを選択します。 結果ウィンドウ枠に表示されている結果が結果セット全体でない場合、残りのローをフェッチするよう要求されます。残りをフェッチしないと、現在フェッチされている結果が選択されます。
Ctrl + C	結果ウィンドウ枠では、選択したローとカラム見出しをクリップボードにコピーします。 SQL 文ウィンドウ枠では、選択したテキストをクリップボードにコピーします。
Ctrl + Shift + C Mac OS X には適用されません。	COMMIT 文を実行します。 <b>i 注記</b> SQL メニューまたはキーボードショートカットを使用して COMMIT を実行した場合、SQL 文ウィンドウ枠の内容は変更されません。ただし、結果ウィンドウ枠の結果タブの内容はクリアされます。
Ctrl + Del	カーソルの右の 1 ワードを削除します。
Ctrl + End	カーソルを現在のウィンドウ枠の一番下に移動します。
Ctrl + F	検索/置換ウィンドウを開きます。
Ctrl + Shift + F10	結果タブの結果セット内にフォーカスがあるセルのポップアップメニューを表示します。 このキーボードショートカットは、テーブルセル内の ... をクリックする代わりに使用します。
Ctrl + G	カーソルを SQL 文ウィンドウ枠内の指定の行に移動します。

キー	説明
Ctrl + H Mac OS X では、Control + Command + H を使用します。	実行した SQL 文の履歴を表示します。
Ctrl + Home	カーソルを現在のウィンドウ枠の先頭に移動します。
Ctrl + K	ロックビューアウィンドウを開きます。
Ctrl + L	現在の行を SQL 文ウィンドウ枠から削除してクリップボードに入れます。
Ctrl + Shift + L	現在の行を削除します。
Ctrl + N	Interactive SQL ウィンドウの内容をクリアし、現在のファイルを閉じます (存在する場合)。
Ctrl + O	ファイルを開きます。
Ctrl + P	SQL 文ウィンドウ枠の内容を印刷します。または、 <b>ツール</b> > <b>オプション</b> > <b>エディタ</b> > <b>印刷</b> をクリックします。
Ctrl + Q Mac OSX では、Command + Q を使用します。	クエリエディタを開きます。クエリエディタを使用すると、SQL クエリを作成できます。クエリの作成が終わったら、OK をクリックして、SQL 文ウィンドウ枠にクエリをエクスポートします。  Mac OSX の場合、このショートカットを使用すると Interactive SQL が終了します。
Ctrl + Shift + R Mac OSX には適用されません。	ROLLBACK 文を実行します。  <b>i 注記</b> SQL メニューまたはキーボードショートカットを使用して ROLLBACK を実行した場合、SQL 文ウィンドウ枠の内容は変更されません。ただし、結果ウィンドウ枠の結果タブの内容はクリアされます。
Ctrl + S	SQL 文ウィンドウ枠の内容を指定ファイルに保存します。
Ctrl + U	選択している内容を小文字に変換します。
Ctrl + T	新しいタブを作成します。
Ctrl + Shift + T	接続ウィンドウを開かずに新しいタブを作成します。
Ctrl + Shift + U	選択している内容を大文字に変換します。
Ctrl + V	選択したテキストを貼り付けます。
Ctrl + X	選択したテキストを切り取ります。
Ctrl + Y	最後の操作を繰り返します。
Ctrl + Z	最後の操作を取り消します。
Ctrl + ]	カーソルを閉じ括弧まで移動します。丸括弧、波括弧、角括弧、山括弧に一致させるときに、このショートカットを使用します。

キー	説明
Ctrl + Shift + ]	選択範囲を閉じ括弧まで拡張します。丸括弧、波括弧、角括弧、山括弧に一致させるときに、このショートカットを使用します。
Ctrl + マイナス記号 (-)	<p>SQL コメントインジケータの二重ハイフン (--) を追加および削除します。</p> <p>既存のテキストをコメントに変換するには、<b>SQL 文</b> ウィンドウ枠でテキストを選択し、Ctrl + マイナス記号キーを押します。SQL コメントインジケータが、選択したテキストの各行の先頭に追加されます。</p> <p>テキストが選択されていない場合、コメントインジケータは現在の行の先頭に追加されます。</p> <p>コメントインジケータを削除するには、テキストを選択して Ctrl + マイナス記号キーを押します。</p>
Ctrl + スラッシュ (/)	<p>SQL コメントインジケータの二重スラッシュ (//) を追加および削除します。</p> <p>既存のテキストをコメントに変換するには、<b>SQL 文</b> ウィンドウ枠でテキストを選択し、Ctrl + スラッシュキーを押します。SQL コメントインジケータが、選択したテキストの各行の先頭に追加されます。</p> <p>テキストが選択されていない場合、コメントインジケータは現在の行の先頭に追加されます。</p> <p>コメントインジケータを削除するには、テキストを選択して Ctrl + スラッシュキーを押します。</p>
Ctrl + 上矢印	<b>SQL 文</b> ウィンドウ枠で、カーソルがある文の前の SQL 文を選択します。
Ctrl + 下矢印	<b>SQL 文</b> ウィンドウ枠で、カーソルがある文の後の SQL 文を選択します。
Ctrl + ペリオド (.)	<b>SQL 文</b> ウィンドウ枠で、カーソルがある SQL 文全体を選択します。または、行をダブルクリックして SQL 文全体を選択します。
Ctrl + カンマ (,)	<b>SQL 文</b> ウィンドウ枠で、カーソルがある行を選択します。
Ctrl + Shift + ペリオド (.)	<p><b>SQL 文</b> ウィンドウ枠で選択されたテキストの行インデントを増加します。</p> <p>テキストが選択されていない場合、インデントは現在の行に適用されます。</p>
Ctrl + Shift + カンマ (,)	<p><b>SQL 文</b> ウィンドウ枠で選択されたテキストの行インデントを減少します。</p> <p>テキストが選択されていない場合、インデントは現在の行に適用されます。</p>

キー	説明
Esc	デフォルトでは、Esc キーを押しても、何も起こりません。 ただし、SQL 文ウィンドウ枠の内容をクリアし、開いている結果セットを閉じるように Esc キーを設定できます。▶ ツール ▶ オプション ▶ 互換性 ▶ をクリックし、Esc キーで SQL 文をクリアし、結果セットを閉じるをクリックします。
F1	オンラインヘルプを表示します。
F2	結果セット内の選択した値を編集します。ローのカラム間でタブ移動できます。
F3	指定したテキストの次の出現箇所を検索します。
Shift + F3	選択したテキストの 1 つ前の出現箇所を検索します。
F5 Mac OSX では Command + R を使用します。	SQL 文ウィンドウ枠内のすべてのテキストを実行します。 この操作は、ツールバーですべての SQL 文の実行をクリックするか、▶ SQL ▶ 実行 ▶ をクリックすることでも実行できます。
Shift + F5	SQL 文ウィンドウ枠で指定された文のプランビューを開きます。指定された文は実行されません。その文をプランビューで実行するには、プランの取得をクリックします。
F6	フォーカスを SQL 文ウィンドウ枠から結果ウィンドウ枠に、またはその逆に移動します。お気に入りウィンドウ枠が開いている場合、フォーカスは SQL 文、結果、お気に入りの各ウィンドウ枠間を移動します。
F7	テーブル名のルックアップウィンドウを表示します。 このウィンドウでテーブルを検索して選択し、Enter キーを押せば、そのテーブル名が SQL 文ウィンドウ枠のカーソル位置に入力されます。また、リストでテーブルを選択し、もう一度 [F7] キーを押すと、そのテーブルのカラムが表示されます。カラムを選択して [Enter] キーを押すと、そのカラム名を SQL 文ウィンドウ枠のカーソル位置に挿入できます。
F8	プロシージャ名のルックアップウィンドウを表示します。 このウィンドウでプロシージャを検索して選択し、Enter キーを押せば、そのプロシージャ名が SQL 文ウィンドウ枠のカーソル位置に入力されます。
F9 Mac OSX では Command + Option + R を使用します。	SQL 文ウィンドウ枠で選択されているテキストを実行します。 テキストが選択されていない場合は、すべての文が実行されます。 この操作は、ツールバーで選択した文の実行をクリックするか、▶ SQL ▶ 実行 ▶ をクリックしても実行できます。

キー	説明
Shift + F9 Mac OSX では Control + R を使用します。	選択された SQL 文を実行し、次の文を選択します。このショートカットを使用すると、一連の SQL 文をステップスルーできます。
Shift + F10	フォーカスのある領域のショートカットメニューを表示します。 このキーボードショートカットは、領域を右クリックする代わりに使用します。
F11	Interactive SQL がデータベースに接続されていない場合に接続ウインドウを開きます。
F12	Interactive SQL と現在のデータベースの接続を切断します。
Home	カーソルを現在の行の行頭、または現在の行の最初の単語に移動します。
Shift + Home	選択範囲を現在の行のテキストの先頭まで拡張します。
Page Down	現在のウインドウ枠を 1 ページ下にスクロールします。
Page Up	現在のウインドウ枠を 1 ページ上にスクロールします。

## 関連情報

[Interactive SQL の SQL 文 \[996 ページ\]](#)

[Interactive SQL 結果セットからロー、カラム、セルをコピーする \[991 ページ\]](#)

[テキスト補完キーボードショートカット \[1030 ページ\]](#)

### 1.7.1.6.5 テキスト補完 (Interactive SQL および SQL Central)

Interactive SQL と SQL Central では、入力時に完全なキーワードとオブジェクト名を供給可能なテキスト補完オプションが提供されます。

たとえば、文字 S を入力すると、文字 S で始まる可能なキーワードまたはオブジェクト名のリストを含むテキスト補完ウインドウが表示されます。テキスト補完設定は、Interactive SQL の [オプションウインドウ](#) または SQL Central の [テキストエディタウインドウ](#) から設定できます。

#### 例

Interactive SQL の [SQL 文ウインドウ](#) 枠で、データベースオブジェクト名の最初の文字を入力します。

デフォルトで、テキスト補完ウインドウは入力すると自動的に表示されます。それ以外では、Ctrl + スペースキー、または Ctrl + Shift + スペースキーを押すと表示されます。テキスト補完ウインドウには、入力した文字で始まるデータベースオブジェクトのリストが含まれます。

使用したいオブジェクト名が見つからない場合は、Ctrl + A キーを押して、データベースオブジェクトの完全なリストを表示します (完全なリストは、ユーザが設定したオプションに基づいて表示されます)。デフォルトでは、すべてのデータベースオブジェクトが表示されます。

リストからオブジェクト名を選択し、Enter キーを押します。

SQL 文ウィンドウ枠にオブジェクト名が表示されます。

このセクションの内容:

[テキスト補完キーボードショートカット \[1030 ページ\]](#)

テキスト補完リストが開いている場合に使用できるキーボードショートカットがあります。

## 1.7.1.6.5.1 テキスト補完キーボードショートカット

テキスト補完リストが開いている場合に使用できるキーボードショートカットがあります。

キー	説明
Ctrl + A	内容に関係なく一致するデータのリストを表示します。
Ctrl + 二重引用符 (") (または Ctrl + Shift + ')	quoted_identifier オプションの設定に関係なく、引用符で囲んで名前の入力を完了します。
Ctrl + アスタリスク(*) (または Ctrl + Shift + 8)	テーブルの場合、カラムのカンマ区切りのリスト (データ型を含む) を挿入します。  ストアドプロシージャの場合、プロシージャ名を挿入し、続けてパラメータ名とデータ型のカンマ区切りのリストを挿入します。
Ctrl + C	テキスト補完リストの内容を変更して、カラムを表示または非表示にします。
Ctrl + F	テキスト補完リストの内容を変更して、SQL 関数を表示または非表示にします。
Ctrl + P	テキスト補完リストの内容を変更して、ストアドプロシージャと関数を表示または非表示にします。
Ctrl + プラス記号 (+) (または Ctrl + Shift + =)	テーブルの場合、カラムのカンマ区切りのリストを挿入します。  ストアドプロシージャの場合、プロシージャ名を挿入し、続けてパラメータ名のカンマ区切りのリストを挿入します。
Ctrl + S	テキスト補完リストの内容を変更して、システムオブジェクトを表示または非表示にします。
Ctrl + Shift + Space または Ctrl + Space	テキスト補完ウィンドウを開きます。Ctrl キーとスペースキーを使用して、テキスト補完ウィンドウを開くこともできます。
Ctrl + T	テキスト補完リストの内容を変更して、テーブルを表示または非表示にします。
Ctrl + V	テキスト補完リストの内容を変更して、ビューを表示または非表示にします。
Esc	テキストを追加しないでテキスト補完ウィンドウを閉じます。
Tab	選択内容を受け入れ、テキスト補完ウィンドウを閉じます。

## 関連情報

[quoted\\_identifier オプション \[766 ページ\]](#)

### 1.7.1.6.6 文のロギング (Interactive SQL)

Interactive SQL で文をロギングします。

#### コンテキスト

文は、ロギングプロセスを停止するか現在のセッションを終了するまで記録されます。

SQL メニュー、または SQL 文のいずれかを使用してロギングを開始または停止できます。

記録された文は、再び使用できるように SQL 文のログファイルに保存されます。

ロギングを開始すると、正常に動作しなかったものも含めて実行した SQL 文がすべて記録されます。

#### 手順

1. Interactive SQL を開きます。
2. ロギングを開始するには、次の手順に従います。

オプション	アクション
SQL メニュー	<ol style="list-style-type: none"><li>1. <b>SQL &gt; ロギングの開始</b> をクリックします。</li><li>2. <b>名前を付けて保存</b> ウィンドウで、ロギングを記録するファイルの場所と名前を指定します。たとえば、そのファイルに <code>mylogs.sql</code> という名前を付けます。</li><li>3. 設定が完了したら、<b>保存</b> をクリックします。</li></ol>
START LOGGING 文	START LOGGING 文を実行します。 たとえば、 <code>c:\mylogs.sql</code> という名前のファイルへのロギングを開始するには、次の文を実行します。 <pre>START LOGGING 'c:\mylogs.sql';</pre>

3. ロギングを停止するには、次の手順に従います。

オプション	アクション
SQL メニュー	<b>SQL &gt; ロギングの停止</b> をクリックします。

オプション	アクション
STOP LOGGING 文	次の文を実行します。  STOP LOGGING;

## 結果

Interactive SQL での文のロギングが開始または停止されます。

## 関連情報

[SQL 文のロギング \[324 ページ\]](#)

[isql\\_command\\_timing オプション \[Interactive SQL\] \[1014 ページ\]](#)

### 1.7.1.6.7 スキーマロックの自動解放 (Interactive SQL)

Interactive SQL を設定して、コミットされていない変更がない場合にデータベーススキーマロックを自動的に解放します。

## コンテキスト

Interactive SQL でテーブルを表示すると、テーブルを変更しなくても、テーブルに対するスキーマロックがデータベースサーバによって作成されます。

Interactive SQL で、**ツール** > **ロックビューア** をクリックすることで、現在の接続に対するロックを表示できます。

## 手順

1. In Interactive SQL で、**ツール** > **オプション** > **SQL Anywhere** をクリックします。
2. スクロール可能なテーブルを選択し、**データベースロックの自動解放** をクリックします。

## 結果

文を実行すると、データベースにコミットされていない変更が接続にあるかどうか<sup>6</sup> Interactive SQL によって確認されます。コミットされていない変更がない場合、Interactive SQL によってスキーマロックが解放されます。

## 関連情報

[IdleTimeout \(IDLE\) 接続パラメータ \[85 ページ\]](#)

[blocking\\_others\\_timeout オプション \[663 ページ\]](#)

### 1.7.1.6.8 ソース制御の統合 (Interactive SQL)

Interactive SQL はサードパーティのソース制御システムと統合でき、Interactive SQL 内からファイルに対する一般的なソース制御操作を実行できます。

Windows の場合、Interactive SQL は、Microsoft Visual SourceSafe などの Microsoft Common Source Code Control API (SCC) をサポートする、ほとんどのソース制御製品と統合できます。Windows やその他のオペレーティングシステムで SCC API をサポートしないソース制御製品を使用するには、ソース制御アクションごとに実行するコマンドラインを指定します。コマンドの出力は、ログウィンドウに表示されます。

Interactive SQL は、次のタスクをサポートします (対象のタスクがソース制御製品でサポートされている場合)。

- ソース制御プロジェクトを開く
- 取得
- チェックイン
- チェックアウト
- チェックアウトの取り消し
- バージョンの比較
- ファイルの履歴の表示
- ファイルのプロパティの表示
- ソース制御マネージャの実行

基本となるソース制御プログラムがアクションをサポートしていない場合は、対応するメニュー項目が無効になります。たとえば、Microsoft Visual SourceSafe は上記のすべてのアクションをサポートしていますが、カスタム (コマンドライン) ソース制御システムを使用すると、ソース制御プロジェクトを開いたり、ソース制御マネージャを実行したりすることはサポートされません。

Interactive SQL からソース制御プログラムを使用する前に、その操作を理解しておく必要があります。

このセクションの内容:

[ソース制御の使用を Interactive SQL に設定する \[1034 ページ\]](#)

ファイルのチェックインやチェックアウト、ファイルの異なるバージョンの比較、ファイルの履歴の表示など、ソース制御操作を行う前に、ソース制御を使用するように Interactive SQL を設定します。

### ソース制御プロジェクトと Interactive SQL [1035 ページ]

ソース制御製品の中には、ソース制御アクションを実行する前にソース制御プロジェクトを開くことが必要なものもあります。

### Interactive SQL の使用によるファイルのチェックアウト [1036 ページ]

SQL 文ウィンドウ枠の内容を変更するか、ファイルメニューのコマンドを使用してファイルをチェックアウトします。

### その他のソース制御アクション [1037 ページ]

ソース制御プロジェクトを開いたり、ファイルのチェックインやチェックアウトを行ったりすることに加えて、Interactive SQL はその他のソース制御アクションをサポートします。

## 1.7.1.6.8.1 ソース制御の使用を Interactive SQL に設定する

ファイルのチェックインやチェックアウト、ファイルの異なるバージョンの比較、ファイルの履歴の表示など、ソース制御操作を行う前に、ソース制御を使用するように Interactive SQL を設定します。

### 前提条件

ソース制御システムを設定する必要があります。

Microsoft SCC API をサポートするソース制御製品を備えた Windows コンピュータ上で Interactive SQL を実行している場合は、その製品またはカスタム (コマンドライン指向の) システムを使用できます。

### 手順

1. Interactive SQL で、**ツール > オプション** をクリックします。
2. 左ウィンドウ枠で、**ソース制御** をクリックします。
3. **ソース制御の統合を有効にする** をクリックします。

オプション	アクション
SCC ソース制御システムの設定	OK をクリックします。
その他のソース制御システムの設定	<ol style="list-style-type: none"><li>1. <b>設定</b> をクリックします。</li><li>2. <b>カスタムソース制御オプション</b> ウィンドウで <b>リセット</b> をクリックします。</li><li>3. リストからソース制御システムを選択し、<b>OK</b> をクリックします。</li><li>4. 必要に応じてリスト内のコマンドを編集します。これを行うには、<b>ソース制御アクション</b> リストからアクションを選択し、<b>コマンドライン</b> ウィンドウ枠に対応するコマンドを入力します。 <b>ソース制御アクション</b> リストで使用システム用のコマンドを定義する場合は、プレースホルダ [FILENAME] を使用して、そのコマンドの実行時に使用するファイル名を表します。たとえば、</li></ol>

オプション	アクション
	<p>Perforce でファイルを送信するためのコマンドは <code>p4 submit [FILENAME]</code> です。このリストに太字で表示される操作だけに、定義されたコマンドがあります。</p> <p>操作に対してコマンドラインを指定しない場合、<b>ファイル</b> &gt; <b>ソース制御</b> メニューの項目は無効になります。</p> <div style="background-color: #ffffcc; padding: 5px; margin: 10px 0;"> <p><b>→ ヒント</b></p> <p>カスタムソース制御オプションウィンドウの<b>エクスポート</b>をクリックすると、ソース制御コマンドラインを外部ファイルにエクスポートできます。このウィンドウにアクセスするには、<b>ツール</b> &gt; <b>オプション</b> をクリックし、<b>ソース制御</b> ウィンドウ枠で<b>設定</b>をクリックします。このウィンドウで<b>インポート</b>をクリックすると、後でこれらのコマンドを読み込むことができます。この機能は、複数のコンピュータ上で Interactive SQL ソース制御コマンドラインを設定する必要がある場合に便利です。</p> </div> <p>5. <b>OK</b> をクリックし、もう一度 <b>OK</b> をクリックします。</p>

## 結果

ソース制御システムが統合されます。

### 1.7.1.6.8.2 ソース制御プロジェクトと Interactive SQL

ソース制御製品の中には、ソース制御アクションを実行する前にソース制御プロジェクトを開くことが必要なものもあります。

プロジェクトの厳密な定義は、使用しているソース制御システムによって異なります。一般には、ソース制御下に置かれているファイルのセットと、それらファイルの作業用コピーが配置されているローカルファイルシステム上のロケーションのことです。プロジェクトを開くには、通常、ユーザ ID とパスワードのようなクレデンシャルをソース制御システムに提供する必要があります。

使用しているソース制御システムでソース制御プロジェクトを開くことがサポートされている場合は、**ファイル** > **ソース制御** > **ソース制御プロジェクトを開く** メニュー項目が有効になります。**ファイル**メニューからこのオプションを選択すると、プロジェクトを開くための、ソース制御に固有のウィンドウが開きます。一度プロジェクトを開くと、以降の Interactive SQL セッションであっても、同じプロジェクトを再び開く必要はありません。プロジェクトは自動的に開きます。

## 1.7.1.6.8.3 Interactive SQL の使用によるファイルのチェックアウト

SQL 文ウィンドウ枠の内容を変更するか、**ファイル**メニューのコマンドを使用してファイルをチェックアウトします。

### コンテキスト

SQL 文ウィンドウ枠でファイルの内容を変更したときに、Interactive SQL によってそのファイルがチェックアウトされるようにするには、Interactive SQL のソース制御オプションを構成するときに**エディタの内容が変更されたら自動的にファイルをチェックアウト**を選択します。

### 手順

1. Interactive SQL で、**ファイル** > **開く** をクリックし、開くファイルを参照します。

#### 警告

ソース制御コード API 準拠のソース制御システムを使用している場合、ステータスは常に正確です。カスタムソース制御システムを使用している場合、ステータスはファイルが読み込み専用かそうでないかに基づきます。読み込み専用画チェックインされることを想定していますが、編集可能ファイルをチェックアウトまたは非制御にすることができます。

Interactive SQL ウィンドウ下部にあるステータスバーに、ファイルのステータスが表示されます。ステータスは、**チェックイン**、**チェックアウト**、*Not Controlled* のいずれかになります。チェックインされているファイルは読み込み専用と見なされ、Interactive SQL のタイトルバーに**読み込み専用**と表示されます。次の例に、チェックインされているファイルを示します。

2. **ファイル** > **ソース制御** > **チェックアウト** をクリックして、ファイルをチェックアウトします。

### 結果

使用しているソース制御製品によっては、チェックアウト手順でコメントの入力やその他のオプション操作を求められることがあります。

### 次のステップ

ファイルの編集が完了したら、Interactive SQL からチェックインできます。**ファイル** > **ソース制御** > **チェックイン** をクリックします。プロンプトが表示されたら、チェックインコメントを入力します。

## 1.7.1.6.8.4 その他のソース制御アクション

ソース制御プロジェクトを開いたり、ファイルのチェックインやチェックアウトを行ったりすることに加えて、Interactive SQL はその他のソース制御アクションをサポートします。

これらのアクションの可用性は、使用しているソース制御システムによって異なります。▶ **ファイル** ▶ **ソース制御** ▶ をクリックしてから、次のいずれかのアクションを選択します。

### 取得

このアクションは、SQL 文ウィンドウ枠で現在開いているファイルの最新のコピーを取得します。

### チェックアウトの取り消し

ファイルをチェックアウトしてから変更を破棄するには、▶ **ファイル** ▶ **ソース制御** ▶ **チェックアウトの取り消し** ▶ をクリックします。この操作によって、ファイルの作業用コピーは破棄され、ソース制御のアーカイブにあるファイルのコピーがダウンロードされます。

### バージョンの比較

このアクションは、開いているファイルの作業用コピーと、ソース制御のアーカイブにあるファイルのバージョンを比較します。

### 履歴

このアクションは、開いているファイルに対して実行されたソース制御アクション (通常はチェックイン) のリストを表示します。

### プロパティ

このアクションは、開いているファイルに関連のあるソース制御プロパティのリストを表示します。

### ソース制御マネージャの実行

このアクションは、使用しているソース制御システムの管理プログラムを起動します。たとえば、Microsoft Visual SourceSafe を使用している場合、Microsoft Visual SourceSafe エクスプローラーを起動します。

## 1.7.1.7 クエリエディタ (Interactive SQL)

クエリエディタは、SELECT 文の構築を支援する Interactive SQL のツールです。

クエリエディタで SQL クエリを作成したり、それらの SQL クエリをインポートして編集したりできます。クエリが完成したら、OK をクリックします。クエリが SQL Central または Interactive SQL にエクスポートされ、処理できるようになります。

### クエリエディタによる SQL 文の使用

クエリエディタでは、クエリを作成するために SQL 文を使用する必要はありません。ただし、次のように、クエリエディタで SQL 文を使用する場合があります。

- Interactive SQL の SQL 文ウィンドウ枠でクエリを作成して、コードを強調表示してからクエリエディタを開くと、クエリエディタにクエリをインポートできます。
- クエリエディタを使用しているときはいつでも、ウィンドウの下部の SQL をクリックすると、構築しているクエリの SQL 文を確認できます。ここでコードを直接編集でき、クエリエディタのフィールドが自動的に更新されます。

## クエリエディタの設定

Interactive SQL または SQL Central でクエリエディタを設定して、SQL が完全なフォーマットで作成される (すべてのテーブルとカラムの名前が完全に修飾され、名前が引用符で囲まれる) ように指定できます。このような特別なフォーマットは通常は必要ありませんが、これによってあらゆる状況において SQL の動作が保証されます。起動時にテーブルのリストを取得することもできます。

Interactive SQL からクエリエディタを設定するには、**ツール** > **オプション** > **SQL Anywhere** をクリックしてから、**クエリエディタ**タブを選択します。

### [クエリエディタ] タブ

クエリエディタには SQL クエリのコンポーネントを設定するための一連のタブがありますが、そのほとんどはオプションです。タブは SQL クエリが通常構築される順序で表示されています。

タブ	説明
テーブルタブ	クエリでテーブルを指定するには、このタブを使用します。
ジョイントタブ	クエリに複数のテーブルを含める場合、テーブルのデータを結合するためのジョイン方式を指定するには、このタブを使用します。ジョイン方式を指定しない場合は、クエリエディタから 1 つの方式が提案されます。テーブルの間に外部キー関係がある場合は、その関係に基づくジョイン条件が生成されます。それ以外の場合は外積 (ベクトル積) が提案されます。クエリを開くとき、クエリエディタはユーザーが指定したジョイン方式をそのまま受け入れます (SQL Anywhere の場合とは異なり、指定のない JOIN が、デフォルトで KEY JOIN になることはありません)。
カラムタブ	結果セットのカラムを指定するには、このタブを使用します。カラムを指定しないと、すべてのカラムが表示されます。
INTO タブ	結果を変数に割り当てるには、このタブを使用します。
WHERE タブ	結果セットのローを制限する条件を指定するには、このタブを使用します。
GROUP BY タブ	結果セットのローをグループ化するには、このタブを使用します。
HAVING タブ	グループの値に基づいて結果セットのローを制限するには、このタブを使用します。
ORDER BY タブ	ローをソートするには、このタブを使用します。

## クエリエディタのツール

クエリエディタには、次のツールもあります。

ウィンドウ	説明
式エディタ	探索条件の構築や、計算カラムの定義には、式エディタを使用します。
派生テーブル	メインのクエリエディタとほとんど同じこのウィンドウを使用して、派生テーブルまたはサブクエリを作成します。

クエリエディタの各コンポーネントには、コンテキスト別のオンラインヘルプがあります。これには、タブの使用方の説明や、関連する概念や使用方法を示す SQL Anywhere マニュアルへのリンクが含まれています。

## クエリエディタの制限事項

クエリエディタは、SELECT 文を構築します。ビューを作成するには設計されていませんが、Interactive SQL でビューを作成してからクエリエディタで参照することはできます。また、SELECT 文以外の、UPDATE 文などの SQL 文の作成にも対応していません。作成されるのは単独の SELECT 文であり、複数の SELECT 文の UNION 演算や INTERSECT 演算は構築されません。さらに、クエリエディタでは Transact-SQL 構文はサポートされません。

このセクションの内容:

[クエリエディタを使用したクエリの作成 \[1039 ページ\]](#)

クエリエディタで SQL クエリの作成、インポート、編集をします。

## 関連情報

[式エディタ \(Interactive SQL\) \[1040 ページ\]](#)

### 1.7.1.7.1 クエリエディタを使用したクエリの作成

クエリエディタで SQL クエリの作成、インポート、編集をします。

#### 手順

1. Interactive SQL からデータベースに接続します。
2. クエリエディタを開きます。

▶ ツール ▶ **クエリの編集** をクリックします。

SQL 文を Interactive SQL で選択していた場合は、選択した文がクエリエディタに自動的にインポートされます。

3. クエリを作成します。

クエリエディタを使用しているときはいつでも、ウィンドウの下部の **SQL** をクリックすると、構築しているクエリの SQL 文を確認できます。ここでコードを直接編集でき、クエリエディタのユーザインタフェースのフィールドが自動的に更新されます。

4. **OK** をクリックして、そのクエリを **SQL 文** ウィンドウ枠に書き込みます。

## 結果

クエリエディタでクエリが作成され、処理用に Interactive SQL にエクスポートされます。

### 1.7.1.8 式エディタ (Interactive SQL)

式エディタによって、探索条件、計算カラム、サブクエリを作成できます。

既存の式を編集する場合は、その式を強調表示してから式エディタを開いてください。強調表示をしないと、**OK** をクリックしたときに、式エディタで作成した式は既存の式の後ろに追加されます。

## 項目

### 式

このボックスで式を構築します。

### カラム

このボックスには、クエリのカラムがリストされます。式にカラムを挿入するには、ここでカラムをダブルクリックするか、**式** ウィンドウ枠にカラムを直接入力します。

### 関数

関数は、データベースの情報を返すために使用される定義済みの式です。関数を式に挿入するには、ドロップダウンリストから式を選択します。

### ストアードプロシージャ

このボックスには、使用できるストアードプロシージャがリストされます。

### 【数字キーパッド】

数字キーパッドは、ウィンドウの左下にあります。クリックした数字が式に挿入されます。キーボードから数字を入力することもできます。

### 比較演算子

= などの算術記号は、ウィンドウの下部中央に表示されます。記号をクリックすると式に挿入されます。ここに表示されるのは一部の一般的な演算子ですが、その他の演算子はキーボードから入力できます。

### 論理演算子

AND などの論理演算子は、ウィンドウの右下に表示されます。演算子をクリックすると式に挿入されます。これらは一般的に使用される論理演算子のサブセットです。キーボードから入力すればあらゆる論理演算子を使用できます。

サブクエリを編集し、エディタに結果を挿入ボタン

このボタンは、論理演算子と同じ場所にあります。このボタンをクリックすると、サブクエリを作成できるウィンドウが表示されます。

## サブクエリの追加

サブクエリを作成するには、式エディタを開き、サブクエリを編集し、エディタに結果を挿入ボタン (*NOT* ボタンの横) をクリックします。これで、サブクエリウィンドウが開きます。このウィンドウは、タイトルバーにサブクエリと表示されること以外はメインのクエリエディタと同じです。サブクエリはメインクエリの場合と同じ方法で作成します。

## 関連情報

[クエリエディタ \(Interactive SQL\) \[1037 ページ\]](#)

### 1.7.1.9 管理ツールのセキュリティ (Interactive SQL)

管理ツールでは、情報を保護するための多くのセキュリティ機能が使用できます。

## Interactive SQL

-nohistory コマンドラインオプション -nohistory コマンドラインオプションを指定すると、プログラムの終了時に SQL 文の履歴が Interactive SQL で保存されないようにすることができます。ただし、プログラムが実行されている間は、SQL 文は履歴タブに表示されます。

### 1.7.2 SQL Central

SQL Central は、データベースサーバ、データベース、およびそれらに含まれているオブジェクトを管理するためのグラフィカルツールです。

## SQL Central 主要機能

### 簡単なコマンドアクセス

オブジェクトを選択すると、SQL Central のファイルメニューが自動的に更新され、そのオブジェクトに直接関連するコマンドが表示されます。オブジェクトを右クリックしてこれらのコマンドにアクセスすることもできます。

### タスクウィザード

新しいオブジェクトを追加するために、SQL Central には手順を追ったタスクを実行するウィザードが用意されています。  
ドラッグアンドドロップ機能

SQL Central では、多くのオペレーションでドラッグアンドドロップ機能を利用できます。たとえば、異なるデータベースにテーブルをコピーしたい場合は、テーブルをクリックしてコピー先にドラッグします。

データベースオブジェクトをコピーするとき、オブジェクトの SQL がクリップボードにコピーされるため、Interactive SQL やテキストエディタなど他のアプリケーションに貼り付けることができます。たとえば、SQL Central でインデックスをコピーし、テキストエディタにペーストすると、そのインデックスの CREATE INDEX 文が表示されます。

キーボードショートカット

一般的に使用される多くのコマンドにはキーボードショートカットがあります。ショートカットは、メニューのコマンド名の横に表示されています。

プラグインサポート

プラグインを使用して、さまざまなデータベース製品やツールを管理できます。SQL Central で、[ヘルプ](#)をクリックしてからプラグイン名を選択すると、そのプラグインの使用や設定についての詳細情報を参照できます。

## 内容の再表示

SQL Central は、データベースに関する情報をキャッシュします。別の接続によるデータベースに変更を加えた場合、SQL Central では、SQL Central でデータベースを再表示するまで変更を認識できません。たとえば、SQL Central によって管理される製品が、同時アクセスをサポートする場合があります。SQL Central がフォルダリストの構造を確認している間に、別のユーザによる変更が発生する場合があります。

常に最新の情報が表示されていることを確認するため、SQL Central の [表示メニュー](#) には次の再表示コマンドがあります。

フォルダの再表示

現在展開されているコンテナのビューを再表示します。F5 キーを押しても、フォルダを再表示できます。

すべて再表示

展開されているすべてのコンテナのビューを再表示します。

これらのコマンドは、SQL Central とそのプラグインに問い合わせる最新の情報を取得します。

## SQL Central の配備

ライセンス契約に従って、SQL Central を含む SQL Anywhere の管理ツールを配備できます。

このセクションの内容:

[SQL Central ナビゲーション \[1043 ページ\]](#)

SQL Central のウィンドウは左右に並んだ 2 つのウィンドウ枠に分割されています。

[データベースオブジェクトの検索 \[1045 ページ\]](#)

SQL Central で、指定したデータベースオブジェクトをデータベース内で検索するか、データベースオブジェクトの SQL 内で文字列を検索します。

[コードエディタ \[1046 ページ\]](#)

コードエディタは、SQL Central の右ウィンドウ枠に **SQL** タブとして表示されます。SQL Central に別のウィンドウとして表示されたり、Interactive SQL に **SQL 文** ウィンドウ枠として表示されることもあります。

#### [ログビューア \[1050 ページ\]](#)

ログビューアは SQL Central のウィンドウで、製品からのメッセージを表示して格納します。

#### [SQL Central プラグインのアーキテクチャ \[1051 ページ\]](#)

SAP の各製品は、SQL Central の別個のプラグインで管理されます。

#### [SQL Anywhere 17 プラグイン \[1053 ページ\]](#)

SQL Anywhere 17 プラグインを使用して、既存データベースのアップグレード、新しいデータベースの作成、データベースの管理を行うことができます。

## 関連情報

### [SQL Central キーポートショートカット \[1046 ページ\]](#)

## 1.7.2.1 SQL Central ナビゲーション

SQL Central のウィンドウは左右に並んだ 2 つのウィンドウ枠に分割されています。

### 左ウィンドウ枠

ビューメニューでは、次のどれを左ウィンドウ枠に表示するか選択できます。

#### フォルダウィンドウ枠

このウィンドウ枠には、データベースオブジェクトの階層ビューが表示されます。

**フォルダ**のオブジェクトツリーに表示されるのはコンテナのみです。他のオブジェクトのコンテナでないオブジェクトは表示されません。たとえば、左ウィンドウ枠には [カラム] フォルダ (コンテナ) は表示されますが、カラムそのものは項目のため表示されず、代わりに右ウィンドウ枠に表示されます。

**フォルダ**ウィンドウ枠に表示するデータベースオブジェクトを設定します。この設定を行うには、データベースを右クリックし、**タイプフィルタの設定**をクリックしてから、左ウィンドウ枠に表示したいオブジェクトタイプを選択します。次に、**OK** をクリックします。必要に応じて、**これを新しいデータベースのデフォルトのタイプフィルタにする**をクリックすると、デフォルトのタイプフィルタを設定したフィルタに設定できます。デフォルトのタイプフィルタは、タイプフィルタが指定されていないデータベースに接続するときに使用されます。

#### タスクウィンドウ枠

このウィンドウ枠には、現在選択されているデータベースオブジェクトのタスクリストが表示されます。

#### 検索ウィンドウ枠

このウィンドウ枠では、プラグイン内のオブジェクトを検索できます。

## 右ウィンドウ枠

右ウィンドウ枠には、現在選択されているコンテナの内容が表示されます。右ウィンドウ枠には、左ウィンドウ枠で選択したコンテナの内容と、選択したコンテナに関するその他の情報を表示するタブがあります。

上部のフィルタフィールドを使用して、オブジェクトをすばやく検索したり、このウィンドウ枠に表示するコンテンツを制限したりします。フィルタフィールドでは、指定テキストを含む項目を検索します。

▶ **表示** ▶ **カラムの選択** ▶ をクリックし、右ウィンドウ枠のタブに表示されるカラムを設定します。

▶ **ツール** ▶ **オプション** ▶ をクリックし、右ウィンドウ枠の表示方法を変更します。

## ツールバー

メインウィンドウのツールバーには、一般的なコマンドのボタンがあります。ツールバーの表示/非表示を切り替えるには、

▶ **表示** ▶ **ツールバー** ▶ **標準ツールバー** ▶ をクリックします。メインツールバーでは、次の作業を実行できます。

- オブジェクトフォルダのナビゲーション
- データベース、データベースサーバ、製品プラグインへの接続または接続の解除
- タスクウィンドウ枠、フォルダウィンドウ枠、または検索ウィンドウ枠の表示
- 接続プロファイルウィンドウにアクセス (ツールメニューからも可能)
- 現在のフォルダのビューの再表示
- オブジェクトの切り取り、コピー、貼り付け、または削除
- 変更の取り消しと再実行
- 選択したオブジェクトのプロパティウィンドウの表示

## コンテキストドロップダウンリスト

コンテキストドロップダウンリストは、ツールバーの下に表示されるリストで、プラグインのオブジェクトフォルダをナビゲーションできます。

## ステータスバー

メニュー間をナビゲーションすると、メインウィンドウの下部に表示されるステータスバーに、メニューコマンドの簡単な説明が表示されます。ステータスバーの表示/非表示を切り替えるには、▶ **表示** ▶ **ステータスバー** ▶ をクリックします。

## 関連情報

[データベースオブジェクトの検索 \[1045 ページ\]](#)

## 1.7.2.2 データベースオブジェクトの検索

SQL Central で、指定したデータベースオブジェクトをデータベース内で検索するか、データベースオブジェクトの SQL 内で文字列を検索します。

### 前提条件

ログインマッピングフォルダ、メンテナンスプラン フォルダ、またはデータベースのページの使用状況タブやロックタブを表示するのに必要な権限がない場合、検索を実行してもこれらのフォルダやタブはスキップされます。

### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. **表示** > **検索ウィンドウ枠** をクリックします。  
左ウィンドウ枠に、**検索ウィンドウ枠**が表示されます。
3. 検索のオプションを設定します。

オプション	アクション
<a href="#">SQL Anywhere17</a> プラグイン	<b>SQL を検索</b> このオプションを選択すると、プロシージャ、イベント、関数、トリガの SQL 内も検索されます。 <b>動的プロパティ (接続、統計、ロック) の検索</b> このオプションを選択すると、接続しているユーザ、SQL Remote 統計情報、テーブルロック、テーブルページの使用状況などの動的プロパティ内も検索されます。
<a href="#">Mobile Link 17</a> プラグイン	<b>スクリプトを検索</b> このオプションを選択すると、同期スクリプト内も検索されます。

4. **検索** をクリックします。

### 結果

左ウィンドウ枠の**結果**に検索結果が表示されます。結果をダブルクリックすると、その結果が右ウィンドウ枠に表示されます。

このセクションの内容:

[SQL Central キーポートショートカット \[1046 ページ\]](#)

SQL Central には、以下のキーボードショートカットがあります。

## 1.7.2.2.1 SQL Central キーボードショートカット

SQL Central には、以下のキーボードショートカットがあります。

ファンクションキー	説明
Alt + Enter	選択した項目のプロパティウィンドウを開きます。
Ctrl + C	選択した内容をクリップボードにコピーします。
Ctrl + F	フィルタをフォーカスします。
Ctrl + V	クリップボードの内容を挿入します。
Ctrl + X	選択した内容を切り取ってクリップボードに移動します。
Delete	選択されている内容を削除します。
F1	SQL Central ヘルプを開きます。
F5	選択したフォルダの内容を再表示します。
F9	接続プロファイルウィンドウを開きます。
F11	複数のプラグインがロードされている場合は、接続メニューを開きます。ロードされているプラグインが1つである場合、F11 キーを押すと、そのプラグインの接続ウィンドウが開きます。
F12	SQL Central の接続が1つだけの場合は切断します。接続が複数の場合は、F12 キーを押すと切断ウィンドウが開き、切断する接続を選択できます。
Shift + F10	選択したオブジェクトのポップアップメニューを開きます。

## 1.7.2.3 コードエディタ

コードエディタは、SQL Central の右ウィンドウ枠に SQL タブとして表示されます。SQL Central に別のウィンドウとして表示されたり、Interactive SQL に SQL 文ウィンドウ枠として表示されることもあります。

SQL Central で別ウィンドウにコードエディタを開くには、左ウィンドウ枠でデータベースオブジェクトを選択し、**ファイル** > **新しいウィンドウで編集** をクリックします。

コードエディタには、標準のテキスト編集機能に加えて次のような機能があります。

- ツールバーとステータスバー
- 構文の自動強調表示
- 言語依存インデント
- テキストの検索と置換機能
- ファイルを開いたり保存したりする機能 (この機能を使用できるかどうかは、使用しているプラグインによって決まります)。
- コード印刷機能
- テキスト補完 (コード入力時)

このセクションの内容:

## コードエディタのキーボードショートカット [1047 ページ]

SQL Central には、コードエディタ用にキーボードショートカットが用意されています。

### 1.7.2.3.1 コードエディタのキーボードショートカット

SQL Central には、コードエディタ用にキーボードショートカットが用意されています。

ファンクションキー	説明
Alt + F4	コードエディタを閉じます (別のウィンドウで編集中の場合)。または、SQL Central を閉じます (SQL Central の右ウィンドウ枠でテキストを編集中の場合)。
Backspace	選択されている内容を削除します。何も選択されていない場合は、Backspace キーを押すとカーソルの左にある 1 文字が削除されます。
Ctrl + ]	カーソルを閉じカッコまで移動します。カッコ、大カッコ、角カッコ、山カッコに一致させるときに、このショートカットを使用します。
Ctrl + A	コードエディタのウィンドウの内容全体を選択します。
Ctrl + Backspace	カーソルの左の 1 ワードを削除します。
Ctrl + C	選択したテキストをクリップボードにコピーします。
Ctrl + Delete	カーソルの右の 1 ワードを削除します。
Ctrl + End	カーソルをコードエディタのウィンドウの一番下に移動します。
Ctrl + F	検索/置換ウィンドウを開きます。現在のウィンドウでテキストを検索していない場合は、指定したテキストを検索して置換できます。それ以外の場合は、指定したテキストの次の出現箇所を検索します。
Ctrl + F3	現在選択されているテキストの次の出現箇所を検索します。
Ctrl + G	移動ウィンドウを開きます。コードエディタのウィンドウ内で移動先の行位置を指定できます。
Ctrl + Home	カーソルをコードエディタのウィンドウの先頭に移動します。
Ctrl + L	現在の行を削除します。
Ctrl + 左矢印	カーソルを 1 ワード分戻します。
Ctrl + N	コードエディタのウィンドウの内容をクリアし、現在のファイルを閉じます (存在する場合)。このショートカットは、SQL Central の右ウィンドウ枠の SQL タブからは使用できません。
Ctrl + O	コードエディタが別ウィンドウとして開かれている場合に、ファイルを開きます。このショートカットは、SQL Central の右ウィンドウ枠の SQL タブからは使用できません。
Ctrl + P	コードエディタのウィンドウの内容を印刷します。▶ ツール ▶ オプション ▶ 印刷 ▶ をクリックすると、印刷テキストの表示を設定できます。
Ctrl + 右矢印	カーソルを 1 ワード分進めます。

ファンクションキー	説明
Ctrl + S	コードエディタのウィンドウの内容を保存します。
Ctrl + Shift + ]	選択範囲を閉じカッコまで拡張します。カッコ、大カッコ、角カッコ、山カッコに一致させるときに、このショートカットを使用します。
Ctrl + Shift + End	選択範囲をコードの終わりまで拡張します。
Ctrl + Shift + F3	現在選択されているテキストより前の出現箇所を検索します。
Ctrl + Shift + Home	選択範囲をコードの先頭まで拡張します。
Ctrl + Shift + L	現在の行を削除します。
Ctrl + Shift + 左矢印	選択範囲を 1 ワード分逆方向に拡張します。
Ctrl + Shift + 右矢印	選択範囲を 1 ワード分先まで拡張します。
Ctrl + Shift + U	選択している内容を大文字に変換します。
Ctrl + Shift + ピリオド(.)	コードエディタのウィンドウで選択されたテキストの行インデントを増やします。テキストが選択されていない場合、インデントは現在の行に適用されます。
Ctrl + Shift + カンマ(,)	コードエディタのウィンドウで選択されたテキストの行インデントを減らします。テキストが選択されていない場合、インデントは現在の行に適用されます。
Ctrl + U	選択している内容を小文字に変換します。
Ctrl + V	クリップボードの内容を現在のカーソル位置に挿入します。
Ctrl + X	選択したテキストを切り取ります。
Ctrl + Y	直前に取り消した操作をやり直します。
Ctrl + Z	最後の操作を取り消します。
Ctrl + マイナス記号 (-)	<p>SQL コメントインジケータの二重ハイフン (--) を追加および削除します。</p> <p>既存のテキストをコメントに変換するには、[コードエディタ] ウィンドウでテキストを選択し、Ctrl + マイナス記号キーを押します。SQL コメントインジケータの二重ハイフンが、選択したテキストを含む各行の先頭に追加されます。</p> <p>テキストが選択されていない場合、コメントインジケータは現在の行の先頭に追加されます。</p> <p>コメントインジケータを削除するには、テキストを選択して Ctrl + マイナス記号キーを押します。</p>

ファンクションキー	説明
Ctrl + スラッシュ (/)	<p>SQL コメントインジケータの二重スラッシュ (//) を追加および削除します。</p> <p>既存のテキストをコメントに変換するには、[コードエディタ] ウィンドウでテキストを選択し、Ctrl + スラッシュキーを押します。SQL コメントインジケータの二重スラッシュが、選択したテキストを含む各行の先頭に追加されます。</p> <p>テキストが選択されていない場合、コメントインジケータは現在の行の先頭に追加されます。</p> <p>コメントインジケータを削除するには、テキストを選択して Ctrl + スラッシュキーを押します。</p>
Delete	選択されている内容を削除します。
下矢印	カーソルを 1 行下に移動します。
End	カーソルを現在の行の末尾に移動します。
F3	検索/置換ウィンドウを開きます。現在のウィンドウでテキストを検索していない場合は、指定したテキストを検索して置換できます。それ以外の場合は、指定したテキストの次の出現箇所を検索します。
Home	カーソルを現在の行の行頭、または現在の行のテキストの先頭に移動します。
左矢印	カーソルを 1 文字左に移動します。
Page Down	カーソルを現在のページの末尾に移動します。
Page Up	カーソルを現在のページの先頭に移動します。
右矢印	カーソルを 1 文字右に移動します。
Shift + 下矢印	選択範囲を 1 行下まで拡張します。
Shift + End	現在の行を選択します。
Shift + F3	検索/置換ウィンドウを開きます。テキストが選択されていない場合は、指定したテキストを検索して置換できます。テキストが選択されている場合は、そのテキストの前の出現箇所を検索します。
Shift + F10	<p>フォーカスのある領域のポップアップメニューを表示します。</p> <p>このキーボードショートカットは、領域を右クリックする代わりに使用します。</p>
Shift + Home	選択範囲を現在の行のテキストの先頭まで拡張します。
Shift + 左矢印	選択範囲を、現在選択している文字の左隣の文字まで拡張します。
Shift + Page Down	選択範囲を 1 ページ分下へ拡張します。
Shift + Page Up	選択範囲を 1 ページ分上へ拡張します。
Shift + 右矢印	選択範囲を、現在選択している文字の右隣の文字まで拡張します。
Shift + 上矢印	カーソルを 1 行分上へ拡張します。
上矢印	カーソルを 1 行上に移動します。

## 1.7.2.4 ログビューア

ログビューアは SQL Central のウィンドウで、製品からのメッセージを表示して格納します。

次のようなタイプのメッセージを表示します。

情報

現在のセッションについての基本的な情報

警告

発生したアクションについての警告メッセージ

エラー

失敗したアクションについてのエラーメッセージ

ビューメニューを使用することで、これらのメッセージをフィルタリングして特定のタイプや番号のメッセージだけを表示したり、特定のプラグインからのメッセージだけが表示されるように選択したりできます。また、メッセージをファイルに保存したり、リストからすべてのメッセージをクリアしたりすることもできます。

ログビューアを開き、現在のメッセージがある場合にそれを表示するには、**ツール > ログビューア** をクリックします。

### ログビューアの個別メッセージの詳細

ログビューアのショートメッセージカラムに特定メッセージの十分な情報が表示されていない場合は、ログビューアでメッセージを選択し、**ビュー > 詳細** をクリックすれば追加の詳細を表示できます。

### ログビューアでのログエントリの検索

ログビューアで検索機能を使用し、特定のメッセージを検索します。ログビューアで、**ビュー > 検索** を選択します。

このセクションの内容:

#### [ログビューアのメッセージリストの設定 \[1051 ページ\]](#)

ログビューア内のメッセージの順序を設定したり、表示されるメッセージの数を制限 (またはフィルタ処理) したり、すべてのメッセージをクリアしたりします。

## 1.7.2.4.1 ログビューアのメッセージリストの設定

ログビューア内のメッセージの順序を設定したり、表示されるメッセージの数を制限 (またはフィルタ処理) したり、すべてのメッセージをクリアしたりします。

### 手順

SQL Central で、**ツール > ログビューア** をクリックし、ログビューアを開きます。

オプション	アクション
すべてのメッセージの表示	ログビューアで、 <b>ビュー &gt; すべてのイベント</b> を選択します。
すべてのメッセージのクリア	ログビューアで、 <b>ログ &gt; すべてのイベントをクリア</b> を選択します。
メッセージの順序の設定	ログビューアで、 <b>ビューメニュー</b> から次のいずれかのオプションを選択します。 <ul style="list-style-type: none"><li>最新のメッセージから並べるには、<b>新しい順</b> を選択します。</li><li>一番古いメッセージから並べるには、<b>古い順</b> を選択します。</li></ul>
メッセージのフィルタリング	<ol style="list-style-type: none"><li>ログビューアで、<b>ビュー &gt; イベントをフィルタする</b> を選択します。 フィルタウィンドウが表示されます。</li><li>オプションを設定します。</li></ol>

### 結果

ログビューアのメッセージリストがカスタマイズされます。

## 1.7.2.5 SQL Central プラグインのアーキテクチャ

SAP の各製品は、SQL Central の別個のプラグインで管理されます。

次の SQL Central プラグインが提供されています。

- SQL Anywhere データベース
- Mobile Link 同期
- Ultra Light データベース

SQL Central で製品を使用するには、それらの製品のプラグインを登録してロードします。多くの場合、製品をインストールするときに、プラグインは自動的に登録およびロードされます。特殊な状況の場合、SQL Central に表示されるプラグインを手動で登録することもできます。

## → ヒント

プラグインのバージョン番号とは別に、SQL Central 自体のバージョン番号があります。SQL Central のバージョン番号またはプラグインのバージョン番号を表示するには、[▶ ヘルプ ▶ SQL Central について ▶](#)をクリックします。

プラグインファイルは、SQL Anywhere インストール環境の以下のロケーションに格納されます。

プラグイン	ファイル名とロケーション
<a href="#">SQL Anywhere 17</a>	%SQLANY17%\java\sapplugin.jar
<a href="#">Mobile Link 17</a>	%SQLANY17%\java\mlplugin.jar
<a href="#">Ultra Light 17</a>	%SQLANY17%\java\ulplugin.jar

このセクションの内容:

### [プラグインの登録 \[1052 ページ\]](#)

プラグイン登録ファイルまたはプラグイン JAR ファイルを指定して、追加のプラグインを登録します。

## 1.7.2.5.1 プラグインの登録

プラグイン登録ファイルまたはプラグイン JAR ファイルを指定して、追加のプラグインを登録します。

### 手順

1. SQL Central で、[▶ ツール ▶ プラグイン ▶](#)を選択します。  
[SQL Central プラグイン](#)ウィンドウが表示されます。
2. 登録するプラグインを選択し、[登録](#)をクリックします。  
[プラグインの登録ウィザード](#)が表示されます。
3. ウィザードの指示に従います。

### 結果

プラグインが登録されます。

## 1.7.2.6 SQL Anywhere 17 プラグイン

SQL Anywhere 17 プラグインを使用して、既存データベースのアップグレード、新しいデータベースの作成、データベースの管理を行うことができます。

モードを選択するには、**モード**メニュー、またはモードに対応するツールバーのボタンを使用します。

SQL Anywhere 17 プラグインは、次のいずれかのモードで操作します。

### 設計モード

設計モードでは、テーブル、ユーザ、トリガ、インデックス、リモートデータベースサーバなどのデータベースオブジェクトを作成したり、変更したりできます。また、テーブルへのデータの追加、新しいデータベースの作成、既存データベースのアップグレードを行うことも可能です。

### デバッグモード

デバッグモードでは、SQL Anywhere のデバッグを使用して、SQL のストアドプロシージャ、トリガ、イベントハンドラの開発に役立てることができます。

## ウィザードによって生成された SQL 文とユーティリティコマンドの表示

ほとんどの SQL Anywhere プラグインウィザードの最後のページでは、指定された選択に基づいてウィザードで実行される SQL 文とユーティリティコマンドが表示されます。

完了をクリックすると、SQL 文かユーティリティコマンドまたはその両方が実行されます。または、SQL 文をクリップボードにコピーし、キャンセルをクリックしてウィザードを終了した後、Interactive SQL を使用して SQL 文を実行することもできます。

この機能を使用すると、データベースを変更することなくウィザードを使用して SQL スクリプトを生成できます。

このセクションの内容:

### [データベーススキーマの比較 \[1054 ページ\]](#)

SQL Central を使用して 2 つのデータベースを比較します。

### [もう一方と一致させるためのデータベーススキーマの変換 \[1055 ページ\]](#)

SQL Central を使用して、データベーススキーマを変換してもう一方と一致させます。

### [ER \(実体関連\) 図の表示 \[1057 ページ\]](#)

SQL Central を使用して、データベースのテーブルの ER 図を表示します。

### [データベースの正常性と統計情報 \[1058 ページ\]](#)

設計モードでは、概要タブにデータベースサーバの概要、正常性、統計、機能が表示されます。

### [データベースのドキュメント化 \(SQL Central\) \[1059 ページ\]](#)

SQL Central のデータベースドキュメントウィザードを使用して、データベースの依存性と参照を表示する HTML ファイルを生成します。

### [ユーザ定義フォルダの作成 \(SQL Central\) \[1060 ページ\]](#)

ユーザ定義フォルダを作成し、データベースオブジェクトを整理します。手動でフォルダを管理するか、データベースでの変更に応じて内容を更新するかを選択できます。

## 関連情報

[SQL Anywhere プロファイラ \[1411 ページ\]](#)

### 1.7.2.6.1 データベーススキーマの比較

SQL Central を使用して 2 つのデータベースを比較します。

#### 前提条件

両方のデータベースで SELECT ANY TABLE システム権限が必要です。

また、パスワードを比較するために、両方のデータベースで ACCESS USER PASSWORD システム権限も必要です。

#### コンテキスト

データベースを比較する際、オブジェクトはオブジェクト ID ではなく、名前、名前と所有者、または名前と所有者の組み合わせで照合されます。

#### 手順

1. SQL Central の *SQL Anywhere17* プラグインを使用すると、比較する 2 つのデータベースに接続できます。
2. **ツール** > *SQL Anywhere17* > **データベーススキーマの比較** をクリックします。  
データベーススキーマの比較ウィンドウが表示されます。
3. 比較する 2 つのデータベースを、**データベース 1** フィールドと **データベース 2** フィールドに指定します。
4. **比較** をクリックします。  
*Objects* タブに、データベース間の違いの概要がリスト化されます。
5. *Objects* タブの項目をクリックすると、そのオブジェクトの SQL 文を参照できます。背景が赤の場合、オブジェクトの SQL は **データベース 1** のみにあり、背景が緑の場合、オブジェクトの SQL は **データベース 2** のみにあり、背景が青の場合、オブジェクトの SQL は両方のデータベースにあることを示します。

#### 結果

データベースが比較されます。

## 次のステップ

データベース 1 のスキーマをデータベース 2 のスキーマと一致させるため、変換に必要な SQL 文が *SQL Script* タブに表示されます。

## 関連情報

[Interactive SQL \[980 ページ\]](#)

### 1.7.2.6.2 もう一方と一致させるためのデータベーススキーマの変換

SQL Central を使用して、データベーススキーマを変換してもう一方と一致させます。

## 前提条件

両方のデータベースで SELECT ANY TABLE システム権限が必要です。

データベース 1 のスキーマがデータベース 2 のスキーマと一致するように変換するために実行する必要がある各 SQL 文に必要な権限を持っている必要があります。

生成された SQL スクリプトには、データベース 1 のスキーマをデータベース 2 のスキーマと同じにするために必要な文が含まれます。スクリプトを実行する前に、データベース 1 はバックアップすることをお奨めします。また、スクリプトの実行中は、データベースに対する他の接続がないことを確認します。

## コンテキスト

次に示す操作は、この機能を使用してデータベース 1 には実行することはできないため、比較中に生成される SQL 文のリストには含まれていません。これらの操作のいずれかが必要な場合は、生成される SQL 文のリストの一番上に、操作を説明する通知が表示されます。

- DB 領域のファイル名の変更
- ドメインの変更
- リモートサーバの読み込み専用設定の変更
- テーブルの DB 領域の変更
- グローバルテンポラリテーブルの COMMIT アクションか SHARE BY ALL 設定またはその両方の変更
- テーブルまたはグローバルテンポラリテーブルでのカラムの順序の変更
- パブリケーションの同期タイプの変更

データベースを比較する際、オブジェクトはオブジェクト ID ではなく、名前、名前と所有者、または名前と所有者の組み合わせで照合されます。データベース 2 にはなく、データベース 1 にはある名前のオブジェクト (名前が変更されたテーブルとカラムを含む) は、削除され再作成されるので、その際にデータが失われる場合があります。

## 手順

1. SQL Central の *SQL Anywhere17* プラグインを使用すると、比較する 2 つのデータベースに接続できます。
2. **ツール** > *SQL Anywhere17* > **データベースの比較** をクリックします。

データベースの比較ウィンドウが表示されます。

3. スキーマを変換するデータベースがデータベース 1 フィールドに指定され、スキーマの変換元データベースがデータベース 2 フィールドに指定されていることを確認します。
4. **比較** をクリックします。

**Objects** タブに、データベース内のオブジェクトがリスト化されます。**SQL スクリプト** タブに生成された SQL 文が表示されます。

5. **Objects** タブで項目をクリックすると、そのオブジェクトの SQL 文の違いを参照できます。背景が赤の場合、オブジェクトの SQL はデータベース 1 のみにあり、背景が緑の場合、オブジェクトの SQL はデータベース 2 のみにあり、背景が青の場合、オブジェクトの SQL は両方のデータベースにあることを示します。
6. **SQL スクリプト** タブの **Interactive SQL** で **開く** をクリックすると、**SQL 文** ウィンドウ枠に文を貼り付けた状態で Interactive SQL を開くことができます。文は必要に応じて編集、実行できます。SQL 文を実行すると、データベース 1 をデータベース 2 と同じにできます。

## 結果

データベース 1 のスキーマがデータベース 2 のスキーマと同じになるよう変換されます。

## 関連情報

[Interactive SQL \[980 ページ\]](#)

### 1.7.2.6.3 ER (実体関連) 図の表示

SQL Central を使用して、データベースのテーブルの ER 図を表示します。

#### 前提条件

PUBLIC システムロールが必要です。

#### コンテキスト

また、ER 図のレイアウトをインポートおよびエクスポートすることもできます。

#### 手順

1. SQL Central で、[SQL Anywhere17 プラグイン](#)を使用してデータベースに接続します。
2. データベースを選択し、右ウィンドウ枠の **ER 図** タブをクリックして ER 図を表示します。
3. **ファイル > ER 図のテーブルを選択** をクリックし、ER 図に表示するテーブルを指定します。
4. **OK** をクリックします。
5. オプションで、データベースのフィルタリングセットを調整して、図に表示するテーブルを変更します。  
図のテーブルは、データベースのフィルタリング設定に従って表示されます。テーブルは所有者に基づいてフィルタリングされます。**ファイル > 所有者フィルタの設定** をクリックします。
6. 図でオブジェクトを整列させます。オブジェクトをドラッグしてレイアウトを変更します。レイアウトの変更は、SQL Central セッション間で保持されます。また、**ファイル > レイアウトのインポート** をクリックして、レイアウトをインポートすることもできます。
7. (任意) テーブルをダブルクリックし、そのテーブルのカラム定義を表示します。

#### 結果

ER 図が変更されます。

#### 次のステップ

**ファイル > レイアウトのエクスポート** をクリックし、ER 図をエクスポートします。

## 関連情報

[データベースの作成 \[265 ページ\]](#)

### 1.7.2.6.4 データベースの正常性と統計情報

設計モードでは、**概要**タブにデータベースサーバの概要、正常性、統計、機能が表示されます。

このタブには、次の項目があります。

#### データベース

このウィンドウ枠は左上隅にあり、データベースサーバに関する一般的な情報が表示されます。

データベースサーバソフトウェアを更新するには、**更新のチェック**をクリックします。

#### 機能

このウィンドウ枠は左下隅にあり、データベースと、その製品および機能を視覚的に表示します。図のノードをクリックすると、右側の**正常性と統計情報**ウィンドウ枠で対応するセクションが展開し、もう一度ノードをクリックするとセクションが折りたたまれます。

#### **i** 注記

Mobile Link の情報の取り出しを開始する必要があります。そうしないと、これらのノードは不明なノードとしてグレー表示されます。

#### 正常性と統計情報

このウィンドウ枠は右側にあり、データベースの全体的なステータスに関する統計と情報を表示します。次の折りたたみ可能なウィンドウ枠があります。

##### 統計情報

ディスクとの間で読み書きされたページ数など、一般的な統計情報を表示します。スケジュールされていない要求がある場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

##### DB 領域

すべての DB 領域を表に示します。DB 領域のサイズがディスクの残り空き領域の 10 % に満たない場合や、DB 領域ファイルが見つからない場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

##### トランザクションログ

必要に応じてトランザクションログとトランザクションログミラーの情報を表示します。このウィンドウ枠は、データベースにトランザクションログがある場合にのみ表示されます。トランザクションログファイルのサイズがディスクの残り空き領域の 10 % に満たない場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

##### 接続しているユーザ

接続しているユーザとトランザクションの統計情報を表示します。トランザクションがあれば、上位 5 つのトランザクション時間を表に示します。ブロックされた接続があれば、すべてのブロックされた接続を表に示します。ブロックされた接続がある場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

##### データベースのミラーリング

システムのプライマリサーバに接続しているときに、プライマリサーバ、監視サーバ、ミラーサーバ、コピーサーバの情報とデータベースミラーリングシステムまたは読み込み専用スケールアウトシステムの情報を表示します。このウ

ウィンドウ枠は、ミラーリングまたはスケールアウトを使用している場合にのみ表示されます。監視サーバまたはミラーサーバが切断されている場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

#### リモートサーバ

データベースが使用するリモートサーバを表に示します。このウィンドウ枠は、リモートサーバが存在する場合にのみ表示されます。リモートサーバが切断されている場合や、リモートサーバが接続を確立できない場合は、警告を表示します。詳細を確認するには警告をクリックしてください。

#### Mobile Link および Notifier

Mobile Link および Notifier の統計値を表示します。このウィンドウ枠は、Mobile Link のテーブルとビューがデータベース内にある場合のみ表示されます。

#### i 注記

このウィンドウ枠の情報を再表示するには、**再表示**ボタンをクリックする必要があります。他のウィンドウ枠の情報と異なり、このウィンドウ枠の情報は、**表示** **再表示** をクリック (またはツールバーの**再表示**アイコンをクリック) しても再表示されません。再表示はデータベースのパフォーマンスに影響を与えることがあるため、この情報はユーザが個別に再表示する必要があります。

#### SQL Remote ユーザ

すべての SQL Remote ユーザと、それぞれの最後の送信時刻および受信時刻を表に示します。このウィンドウ枠は、データベースに SQL Remote ユーザが存在する場合にのみ表示されます。

## 関連情報

[ソフトウェアの更新 \[1062 ページ\]](#)

### 1.7.2.6.5 データベースのドキュメント化 (SQL Central)

SQL Central のデータベースドキュメントウィザードを使用して、データベースの依存性と参照を表示する HTML ファイルを生成します。

#### 手順

1. SQL Central で、**ツール** **SQL Anywhere17** **データベースドキュメントの生成** をクリックします。
2. ウィザードの指示に従います。

#### 結果

生成されるドキュメントは、ナビゲーションや確認がしやすいように HTML ファイルに保存されます。

## 1.7.2.6.6 ユーザ定義フォルダの作成 (SQL Central)

ユーザ定義フォルダを作成し、データベースオブジェクトを整理します。手動でフォルダを管理するか、データベースでの変更に応じて内容を更新するかを選択できます。

### コンテキスト

フォルダには1つのタイプのデータベースオブジェクトを保存します。次のタイプのフォルダを作成できます。

#### フォルダ

データベース内のオブジェクトを選択することで、フォルダに含まれる一連のオブジェクトを指定します。フォルダを作成すると、オブジェクトをコピーアンドペーストまたはドラッグアンドドロップすることでオブジェクトを追加できます。プレーンフォルダ内のオブジェクトは、フォルダから明示的に削除されるか、データベースから削除されるまで保持されます。

スマートフォルダ データベース内のゼロ個以上のオブジェクトと一致する式を定義することで、スマートフォルダに含まれる一連のオブジェクトを指定します。データベースオブジェクトを作成、変更、削除すると、フォルダの式と一致するオブジェクトのみがフォルダに含まれるようにスマートフォルダの内容が変更されます。オブジェクトのスマートフォルダへの明示的な追加や、スマートフォルダからの削除はできません。

ユーザ定義フォルダの定義はデータベースの外部に保存され、フォルダの作成で使用したデータベース名、データベースサーバ、コンピュータ名に固有のものです。フォルダ定義は、ファイルにエクスポートしてから別のデータベースやデータベースサーバで使用するためにインポートすることで共有することができます。

### 手順

1. SQL Central で、データベースに接続します。
2. 左ウィンドウ枠でデータベースを右クリックして、**新規** > **フォルダ** をクリックします。
3. フォルダ名を入力します。次へをクリックします。
4. フォルダタイプを選択します。次へをクリックします。
5. オブジェクトタイプを選択します。次へをクリックします。
6. 次のオプションのうちの1つを選択してください。

オプション	アクション
フォルダ	フォルダの内容を指定します。
スマートフォルダ	スマートフォルダ式エディタで、フォルダに含めるデータベースオブジェクトと一致する式を定義します。

7. **完了**をクリックします。

フォルダが作成されました。フォルダに含まれるオブジェクトタイプは変更できません。そのオブジェクトタイプ用の新しいフォルダを作成してください。

8. フォルダ定義をファイルにエクスポートすることで、他のユーザや別のデータベースとフォルダを共有します。

- a. 左ウィンドウ枠でデータベースを右クリックして、**フォルダのエクスポート**をクリックします。
- b. ウィザード内の指示に従って、エクスポートするフォルダを選択します。フォルダは、別のデータベースシステムでの使用やバックアップとしての保存の目的でインポートすることができます。

## 結果

他のユーザと共有できるフォルダの定義ファイルが作成されました。

フォルダ定義をインポートするには、SQL Central で別のデータベースシステムに接続し、データベースを右クリックして、**フォルダのエクスポート**を選択します。

### 例

スマートフォルダのサンプル式を以下に示します。

- 所有者が 'DBA' のテーブルを表示します

```
Owner = 'DBA'
```

- 'APP1\_' で始まる名前のテーブルを表示します

```
Name STARTS WITH 'APP1_'
```

- 所有者が 'DBA' で、'APP1\_' で始まる名前のテーブルを表示します

```
Owner = 'DBA' AND Name STARTS WITH 'APP1_'
```

- 所有者が 'DBA' で、'APP1\_' で始まる名前または '\_test' を含む名前のテーブルを表示します

```
Owner = 'DBA' AND (Name STARTS WITH 'APP1_' OR Name CONTAINS '_test')
```

- 所有者が 'DBA' で、タイプが 'Proxy' のテーブルを表示します

```
Owner = 'DBA' AND Type = 'Proxy'
```

- 失敗ログインの試行回数が 5 回を超えるユーザを表示します

```
FailedLoginAttempts > 5
```

- 最終ログイン時刻が '2013-01-01 00:00:00' より前のユーザを表示します

```
LastLoginTime < '2013-01-01 00:00:00'
```

- 最終ログイン時刻が 1 時間以内のユーザを表示します

```
LastLoginTime IN THE PREVIOUS '1h'
```

- システムにログインしたことのあるユーザを表示します

```
LastLoginTime IS NOT NULL
```

以下の式は、**ユーザ**を含むスマートフォルダ用の式です。システムにログインしたことがあり、文字 u で始まる名前で、失敗ログインの試行回数が 5 回以上で、パスワードが 2015 年 1 月 1 日正午よりも前に作成されており、最終ログインが過去 15 日以内である、ロックを解除されるユーザアカウントを表示します。

```
LastLoginTime IS NOT NULL
```

```
AND Name STARTS WITH 'u'  
AND FailedLoginAttempts > 5  
AND Locked = 'No'  
AND Type = 'User'  
AND PasswordCreationTime < '2015-01-01 12:00:00'  
AND LastLoginTime IN THE PREVIOUS '15d'
```

## 1.7.3 ソフトウェアの更新

複数の方法でソフトウェアの更新を確認できます。

スタートメニュー

▶ スタート ▶ プログラム ▶ *SQL Anywhere17* ▶ **更新のチェック** をクリックします。

SQL Central

▶ ヘルプ ▶ *SQL Anywhere17* ▶ **更新のチェック** をクリックします。

Interactive SQL

▶ ヘルプ ▶ **更新のチェック** をクリックします。

SQL Anywhere サポートユーティリティ (dbsupport)

次のコマンドを実行します。

```
dbsupport -iu
```

このセクションの内容:

[更新チェッカーの設定 \(SQL Central の場合\) \[1063 ページ\]](#)

サポートパッケージやマイナーリリースなどの更新が利用できるようになったときにそのことを通知するように更新チェッカーを設定します。

[更新チェッカーの設定 \(Interactive SQL の場合\) \[1063 ページ\]](#)

サポートパッケージやマイナーリリースなどの更新が利用できるようになったときにそのことを通知するように更新チェッカーを設定します。

## 関連情報

[トラブルシューティング: エラーのレポート \[973 ページ\]](#)

[A Guide to Downloading SAP SQL Anywhere Support Packages \(SAP SQL Anywhere サポートパッケージのダウンロードガイド\)](#)

### 1.7.3.1 更新チェッカーの設定 (SQL Central の場合)

サポートパッケージやマイナーリリースなどの更新が利用できるようになったときにそのことを通知するように更新チェッカーを設定します。

#### コンテキスト

デフォルトでは、SQL Anywhere はソフトウェア更新を毎日チェックします。

#### 手順

1. SQL Central で、**ヘルプ** > **SQL Anywhere17** > **更新チェッカーの設定** をクリックします。
2. **ソフトウェアの更新と通知**の設定を編集します。
3. **OK** をクリックします。

#### 結果

更新チェッカーが設定されます。

#### 関連情報

[サポートユーティリティ \(dbsupport\) \[1165 ページ\]](#)

[Performance data collection details \(パフォーマンスデータ収集詳細\)](#)

### 1.7.3.2 更新チェッカーの設定 (Interactive SQL の場合)

サポートパッケージやマイナーリリースなどの更新が利用できるようになったときにそのことを通知するように更新チェッカーを設定します。

#### コンテキスト

デフォルトでは、SQL Anywhere はソフトウェア更新を毎日チェックします。

## 手順

1. Interactive SQL で、**ツール** > **オプション** > **サポート** をクリックします。
2. **ソフトウェアの更新と通知**の設定を編集します。
3. **OK** をクリックします。

## 結果

更新チェッカーが設定されます。

## 関連情報

[サポートユーティリティ \(dbsupport\) \[1165 ページ\]](#)

[Performance data collection details \(パフォーマンスデータ収集詳細\)](#)

## 1.7.4 データベース管理ユーティリティ

データベース管理タスクを実行するための複数のユーティリティプログラムが提供されています。

管理ユーティリティは、さまざまな管理およびメンテナンス機能を実行するために使用されます。

このセクションの内容:

### [設定ファイル \[1066 ページ\]](#)

多くのユーティリティでは、設定ファイルにコマンドラインオプションを保存できます。

### [バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

実行中のデータベースのデータベースファイルやトランザクションログについて、クライアント側またはサーバ側のバックアップを作成します。

### [Broadcast Repeater ユーティリティ \(dbns17\) \[1077 ページ\]](#)

他のサブネット上で実行されているデータベースサーバや、ファイアウォールの反対側において UDP ブロードキャストが通常は届かない SQL Anywhere データベースサーバを、クライアントが検索できるようにします。

### [証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

X.509 証明書を作成します。

### [証明書ビューアユーティリティ \(viewcert\) \[1087 ページ\]](#)

パブリックキーインフラストラクチャ (PKI) オブジェクト内の値の表示、PKI オブジェクトのエンコーディングの変換、プライベートキーの暗号化と復号化を行います。

### [データソースユーティリティ \(dbdsn\) \[1089 ページ\]](#)

SQL Anywhere ODBC データソースの作成、削除、説明、およびリスト作成を行います。

#### [dbisqlc ユーティリティ \(廃止予定\) \[1101 ページ\]](#)

データベースで SQL 文を実行します。

#### [消去ユーティリティ \(dberase\) \[1103 ページ\]](#)

データベースファイルと関連するトランザクションログを消去します。または、個々のトランザクションログファイルを消去します。

#### [イベントトレースデータ \(ETD\) ファイル管理ユーティリティ \(dbmanageetd\) \[1104 ページ\]](#)

ユーザ定義のイベントとシステムイベントの情報を確認できる診断ログを生成します。イベントの説明を参照するには、sp\_trace\_events システムプロシージャを使用します。

#### [ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

難読化または単純暗号化を使用して、設定ファイルと初期化ファイルの内容を非表示にします。

#### [ヒストグラムユーティリティ \(dbhist\) \[1110 ページ\]](#)

ヒストグラムを Microsoft Excel チャートに変換します。これには、述部の選択性に関する情報が含まれます。

#### [情報ユーティリティ \(dbinfo\) \[1111 ページ\]](#)

指定したデータベースに関する情報を表示します。

#### [初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

新しいデータベースを作成します。

#### [Interactive SQL ユーティリティ \(dbisql\) \[1125 ページ\]](#)

SQL 文を実行し、データベースに対してスクリプトファイルを実行します。

#### [キーペアジェネレータユーティリティ \(createkey\) \[1130 ページ\]](#)

Mobile Link エンドツーエンド暗号化で使用する RSA のキーペアを作成します。

#### [言語選択ユーティリティ \(dblang\) \[1131 ページ\]](#)

データベースサーバ、SQL Central、Interactive SQL、およびその他のツールで使用される言語を制御するレジストリ設定をレポートし、変更します。

#### [ログ変換ユーティリティ \(dbtran\) \[1133 ページ\]](#)

トランザクションログを SQL スクリプトファイルに変換します。

#### [パフォーマンス統計値ユーティリティ \(dbstats\) \(UNIX\) \[1138 ページ\]](#)

UNIX でデータベースサーバのパフォーマンス統計値を返します。

#### [Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

データベースサーバとデータベースへの接続をテストします。

#### [プロファイラユーティリティ \(dbprof\) \[1144 ページ\]](#)

プロファイラを起動し、データベースのプロファイリングまたは以前のプロファイリングセッション情報の検証を行えるようにします。

#### [サーバ列挙ユーティリティ \(dblocate\) \[1145 ページ\]](#)

TCP/IP ネットワーク上のデータベースサーバを検索します。

#### [サーバライセンス取得ユーティリティ \(dblic\) \[1148 ページ\]](#)

データベースサーバまたは Mobile Link サーバに、ソフトウェアライセンスを適用します。

#### [Linux 用サービスユーティリティ \(dbsvc\) \[1152 ページ\]](#)

SQL Anywhere サービスの作成、変更、削除を行います。

#### [Windows 用サービスユーティリティ \(dbsvc\) \[1156 ページ\]](#)

SQL Anywhere サービスの作成、変更、削除を行います。

[サーババックグラウンド起動ユーティリティ \(dbspawn\) \[1161 ページ\]](#)

バックグラウンドでデータベースサーバを起動します。

[サーバ停止ユーティリティ \(dbstop\) \[1163 ページ\]](#)

データベースまたはデータベースサーバを停止します。

[サポートユーティリティ \(dbsupport\) \[1165 ページ\]](#)

エラーや診断情報に関するパフォーマンスデータをテクニカルサポートに送信します。

[トランザクションログユーティリティ \(dblog\) \[1173 ページ\]](#)

データベースのトランザクションログを管理します。

[アンロードユーティリティ \(dbunload\) \[1176 ページ\]](#)

指定したディレクトリにデータベースをアンロードします。

[アップグレードユーティリティ \(dbupgrad\) \[1197 ページ\]](#)

システムテーブルとビューの更新、新しいデータベースオプションの追加、すべてのシステムストアプロセスの再作成、jConnect サポートのインストール、トランザクションログのアーカイブ、新しいトランザクションログの作成を行います。

[検証ユーティリティ \(dbvalid\) \[1200 ページ\]](#)

テーブルとマテリアライズドビューのインデックスとキーを検証します。

[バージョン診断ユーティリティ \(dbversion\) \[1204 ページ\]](#)

指定された実行プログラムに関する情報を返します。

## 関連情報

[レジストリと INI ファイル \[565 ページ\]](#)

[SQL Central \[1041 ページ\]](#)

[Interactive SQL \[980 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

### 1.7.4.1 設定ファイル

多くのユーティリティでは、設定ファイルにコマンドラインオプションを保存できます。

オプションの拡張セットを使用する場合は、それらを設定ファイルに保存すると便利です。

@data オプションを使用すると、コマンドラインで環境変数または設定ファイルを指定できます。

環境変数を指定するには、@data を環境変数の名前に置き換えます。

設定ファイルを指定するには、次のように、data を設定ファイルのパスと名前に置き換えます。

同じ名前の環境変数と設定ファイルが存在する場合は、環境変数が使用されます。

設定ファイルには、改行を含めたり、@data オプションを含むあらゆるオプションの設定を格納したりできます。行の先頭にシャープ記号 (#) を使用すると、その行をコメントとして指定することができます。行の最後に単独で存在するアンパサンド (&)

文字は、前のトークンが次の行に続くことを示します。たとえば、次の設定ファイルを使用して、強力な暗号化を許可するサーバを起動できます。

```
-ec TLS (FIPS=Y;IDENTITY=rsaserver.id; &
    IDENTITY_PASSWORD=test)
-x TCPIP c:¥mydemo.db
```

@data パラメータはコマンドラインの任意の位置に指定でき、ファイルに含まれるパラメータがその位置に挿入されます。さらに @data は、1つのコマンドラインで複数回使用して複数の設定ファイルまたは環境変数を指定できます。

ユーティリティは、指定の設定ファイルを展開し、コマンドライン全体を左から右へ読み取ります。コマンドライン内のその他のオプションで上書きされるオプションを指定した場合、行の終端に近い方のオプションが有効になります。競合するオプションのためにエラーが発生することがあります。

### i 注記

サーババックグラウンド起動ユーティリティ (dbspawn) では、@data オプションで指定された設定ファイルは展開されません。

設定ファイル内のパスワードなどの情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容を暗号化します。

## 設定ファイルのエスケープ文字

設定ファイルでは、オプションの値にスペースを含む値が含まれている場合は、その値を二重引用符で囲む必要があります。さらに、引用符で囲んだ値にスペースを含む別の値が含まれている場合は、その別の値を囲む二重引用符をエスケープする必要があります。SQL Anywhere では、¥のエスケープシーケンスとして ¥¥、"のエスケープシーケンスとして ¥" がサポートされています。

たとえば、次に示す dblsn 設定ファイルの抜粋には、エスケープされた二重引用符が含まれています。

```
-l "subject=$remote_id;
content=sync cardealer;
action='run dbmlsync.exe -c ¥"filedsn=c:¥my fdsns¥CarDealer.dsn¥"
-ot dbmlsync.log -k -e sa=on';"
```

### 例

次の設定ファイルには、検証ユーティリティ (dbvalid) の一連のオプションが含まれています。

```
#Connect to the sample database as the user DBA with password sql
-c "UID=DBA;PWD=sql;DBF=%SQLANYSAMP17%¥demo.db"
#Perform an express check on each table
-fx
#Log output messages to the specified file
-o "c:¥validationlog.txt"
```

この設定ファイルを c:¥config.txt として保存すると、コマンドで次のように使用できます。

```
dbvalid @c:¥config.txt
```

このセクションの内容:

[設定ファイルでの条件付き解析 \[1068 ページ\]](#)

設定ファイルで条件付き解析を使用することによって、そのファイルを使用できるユーティリティを指定できます。

## 関連情報

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

### 1.7.4.1.1 設定ファイルでの条件付き解析

設定ファイルで条件付き解析を使用することによって、そのファイルを使用できるユーティリティを指定できます。

#### 構文

```
configuration-file= text...
```

```
text : comment | conditional | command-line-option
```

```
comment : line starting with # that is not a conditional
```

```
conditional :
```

```
#if condition  
text  
  [ #elif condition  
text  
  ] ...  
  [ #else  
text  
  ] ...  
#endif
```

```
condition : { tool=utility-name[,utility-name]... | utility-name }
```

utility-name では、次の値がサポートされています。

dbbackup	dbinfo	dbmlsync	dbstop	dbxtract
dbdsn	dbinit	dbping	dbsupport	mlsrv
dbeng	dblic	dbremote	dbsvc	mluser
dberase	dblocate	dbspawn	dbunload	qaagent
dbfhide	dblog	dbsrv	dbupgrad	mlstop
dbhist	dblsn	dbstats	dbvalid	

## 備考

条件ディレクティブを使用すると、設定ファイルを使用するユーティリティに応じてコマンドパラメータを適用したり除外したりすることができます。設定ファイルで条件付き解析を使用する場合でも、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容を暗号化することができます。

ディレクティブと認識されるようにするには、1 つの行で空白以外の最初の文字を # とします。#if または #elif ディレクティブ内にユーティリティが含まれている場合、そのディレクティブから次の条件付きディレクティブまでの行が処理の対象となります。#else ディレクティブは、それより前のブロックにユーティリティがない場合に実行する処理を示します。条件付きディレクティブ構造は、#endif ディレクティブで終了します。

tool= で指定するツール名のリスト内に空白を入れることはできません。条件付きディレクティブはネストすることができます。設定ファイルの解析中にエラーが発生すると、ユーティリティは設定ファイルを開くことができない旨をレポートします。

dbspawn ユーティリティでは、作成するコマンドで設定ファイル参照を指定できますが、dbspawn ユーティリティのオプションを使用して設定ファイルを指定することはできません。たとえば、次の最初のコマンドはサポートされていますが、2 つ目のコマンドはサポートされていません。

```
dbspawn dbeng17 @myconfig.ini
dbspawn @spawnopts.ini dbeng17 demo.db
```

## 例

次の設定ファイルを使用できるユーティリティは、dbping、dbstop、dbvalid です。

```
#if tool=dbping,dbstop,dbvalid
#always make tools quiet
-q
-c "UID=DBA;PWD=passwd;Host=myhost;DBN=mydb"
#if dbping
#make a database connection
-d
#elif tool=dbstop
#don't ask
-y
#else
#must be dbvalid
#use WITH EXPRESS CHECK
-fx
#endif
#endif
```

## 関連情報

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

[データソースユーティリティ \(dbdsn\) \[1089 ページ\]](#)

[@data データベースサーバオプション \[382 ページ\]](#)

[消去ユーティリティ \(dberase\) \[1103 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

[ヒストグラムユーティリティ \(dbhist\) \[1110 ページ\]](#)

[情報ユーティリティ \(dbinfo\) \[1111 ページ\]](#)

- 初期化ユーティリティ (dbinit) [1113 ページ]
- サーバライセンス取得ユーティリティ (dblic) [1148 ページ]
- サーバ列挙ユーティリティ (dblocate) [1145 ページ]
- トランザクションログユーティリティ (dblog) [1173 ページ]
- Ping ユーティリティ (dbping) [1140 ページ]
- サーババックグラウンド起動ユーティリティ (dbspawn) [1161 ページ]
- サーバ停止ユーティリティ (dbstop) [1163 ページ]
- Linux 用サービスユーティリティ (dbsvc) [1152 ページ]
- Windows 用サービスユーティリティ (dbsvc) [1156 ページ]
- アンロードユーティリティ (dbunload) [1176 ページ]
- アップグレードユーティリティ (dbupgrad) [1197 ページ]
- 検証ユーティリティ (dbvalid) [1200 ページ]

## 1.7.4.2 バックアップユーティリティ (dbbackup)

実行中のデータベースのデータベースファイルやトランザクションログについて、クライアント側またはサーバ側のバックアップを作成します。

### 構文

```
dbbackup [ options ] target-directory
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-aw[ - ]	ライタの自動チューニングを有効にします。こうすることでライタの数を増やし、全体のバックアップパフォーマンスを改善できます。-aw- を指定するとライタの自動チューニングが無効になり、データベースサーバがそれ以上ライタを作成しなくなります。-aw オプションと -aw- オプションは必ず -s オプションと一緒に使用してください。
-bblock-size	データベースサーバから dbbackup へのページ転送に使用するブロックの最大サイズ (ページ数) を指定します。dbbackup ユーティリティは、指定された数のページを割り付けようとします。割り付けが失敗した場合は、この値を半分にして再試行し、成功するまでこれを繰り返します。デフォルトのサイズは 128 です。

オプション	説明
<code>-bccomment</code>	バックアップ履歴ファイルにコメントを記録します。アーカイブバックアップの場合、コメントはアーカイブファイルにも記録されます。 <code>-bc</code> オプションは必ず <code>-s</code> オプションと一緒に使用してください。
<code>-c"keyword=value;..."</code>	接続パラメータを指定します。
<code>-d</code>	トランザクションログファイルがあっても、それをバックアップしないで、メインデータベースファイルだけをバックアップします。
<code>-h[ - ]</code>	バックアップ履歴を有効にし、 <code>backup.syb</code> ファイルに行を追加します。 <code>-h-</code> を指定するとバックアップ履歴が無効になり、 <code>backup.syb</code> ファイルが更新されなくなります。 <code>-h</code> オプションと <code>-h-</code> オプションは必ず <code>-s</code> オプションと一緒に使用してください。

オプション	説明
-kcheckpoint-log-copy-option	<p>バックアップ先のディレクトリに書き込む前に dbbackup でデータベースファイルに対してどのような処理を実行するかを指定します。バックアップ中に更新前イメージを適用するか、チェックポイントログをバックアップとしてコピーするかを選択します。どちらを選択するかによって、パフォーマンスに違いが生じます。-s オプションを指定してサーバ側でバックアップを実行する場合、-k のデフォルト設定は auto です。それ以外の場合は、copy がデフォルト設定となります。</p> <p><b>auto</b></p> <p>データベースサーバはバックアップディレクトリがあるボリュームの空きディスク領域サイズをチェックします。バックアップを開始する時点でデータベースサイズの 2 倍以上の空きディスク領域がある場合は、copy を指定した場合と同じ方法でバックアップが実行されます。それ以外の場合は、nocopy を指定した場合と同じ方法でバックアップが実行されます。この設定は、-s を指定した場合のみ使用できます。</p> <p><b>copy</b></p> <p>バックアップでは、変更されたページの更新前イメージを適用せずにデータベースファイルを読み込みます。チェックポイントログの全体とシステム DB 領域がバックアップディレクトリにコピーされます。このデータベースを次に起動すると、データベースは自動的にバックアップ開始時のチェックポイントの状態にリカバリされます。</p> <p>このオプションを使用すると、ページの更新前イメージをテンポラリファイルに書き込む必要がないため、バックアップパフォーマンスが向上し、バックアップ中に動作中の他の接続との内部サーバ競合が減少します。ただし、データベースファイルのバックアップコピーにはチェックポイントログが含まれ、このログにはバックアップの開始以降に変更されたページの更新前イメージが保存されているため、バックアップコピーのサイズがバックアップを開始した時点のデータベースファイルより大きくなる場合があります。copy オプションは、バックアップ先ディレクトリのディスク領域が十分である場合に使用してください。</p> <p><b>nocopy</b></p> <p>チェックポイントログがバックアップとしてコピーされません。この場合、変更されたページの更新前イメージはテンポラリファイルに保存され、バックアップの実行中はバックアップに適用されます。データベースファイルのバックアップコピーは、バックアップを開始した時点のデータベースと同じサイズになります。バックアップコピーは、チェックポイントログが含まれていないため、実際には多少小さくなる場合があります。このオプションを指定すると、データベースファイルのバックアップは小</p>

オプション	説明
	<p>さくなりますが、バックアップの速度が遅くなり、データベースサーバで実行される他の操作のパフォーマンスが低下することがあります。バックアップ先のドライブの空き領域が少ない場合に、このオプションが役に立ちます。</p> <p><b>recover</b></p> <p>copy オプションを指定した場合と同じようにチェックポイントログがコピーされますが、このチェックポイントログはバックアップの完了時にデータベースに適用されます。したがって、データベースファイルのバックアップは、バックアップ操作を開始した時点と同じ状態（および同じサイズ）となります。このオプションが役に立つのは、バックアップドライブの空き領域が少ない場合です（チェックポイントログをバックアップする copy オプションの場合と同じ量の空き領域が必要ですが、バックアップファイルのサイズは copy オプションを指定した場合より小さくなります）。この設定は、-s を指定した場合のみ使用できます。</p>
-filename	<p>サーバがクラッシュした場合に第 2 のシステムをすばやく起動できるようにします。サーバが実行されている間、ライブバックアップは停止しないで実行を続けます。ライブバックアップは、プライマリサーバが使用できなくなるまで実行されます。クラッシュが発生した時点でライブバックアップは停止しますが、バックアップされたログファイルはそのまま残り、第 2 のシステムを即座に起動するために使用できます。</p> <p>-l を指定する場合は、-s を使用してサーバ上にバックアップを作成できません。</p>
-n	<p>バックアップトランザクションログファイルの命名規則を <code>yyymmddxx.log</code> に変更します。xx は AA から ZZ までの連続した英字を表し、yyymmdd は現在の年月日を表します。このオプションは -r と一緒に使用します。</p> <p>トランザクションログファイルのバックアップコピーは、<code>yyymmddxx.log</code> の命名規則に基づいて、コマンドで指定したディレクトリに保存されます。これにより、トランザクションログファイルの複数のバージョンのバックアップを、同じバックアップディレクトリに保存することができます。</p> <p>また、-x オプションと -n オプションの両方を使用して、ログコピーの名前を変更することもできます。次に例を示します。</p> <pre>dbbackup -c "UID=DBA;PWD=passwd" -x -n mybackupdir</pre>
-Ofilename	指定したファイルに、出力メッセージを追加します。

オプション	説明
-p	<p>データベースサーバからクライアントに、フォーマットされた進行メッセージを送信します。</p> <p>-s オプションを使用してサーバ側のバックアップを実行しないかぎり、-p オプションは無視されます。</p>
-q	<p>出力メッセージを表示しません。このオプションは、このユーティリティをコマンドプロンプトで実行する場合のみ使用できます。</p>
-r	<p>トランザクションログの名前を変更して新しいログを開始します。このオプションを使用すると、チェックポイントが発生し、次の 3 つの手順が発生します。</p> <ol style="list-style-type: none"> <li>1. 現在作業しているトランザクションログファイルのコピーが作成され、コマンドで指定したディレクトリに保存されます。</li> <li>2. 現在のトランザクションログは現在のディレクトリ内に残りますが、yyymmddxx.log のフォーマットに従って名前が変更されます。xx は AA から ZZ までの連続した英字を表し、yyymmdd は現在の年月日を表します。このファイルは、現在のトランザクションログではなくなります。</li> <li>3. トランザクションを含まない新しいトランザクションログファイルが作成されます。このファイルに、直前まで現在のトランザクションログとして使用されていたファイルの名前が付けられ、データベースサーバによって現在のトランザクションログとして使用されます。</li> </ol>
-s	<p>BACKUP DATABASE 文を使用して、サーバ上にイメージのバックアップを作成します。-s オプションを指定する場合、-l オプション (トランザクションログのライブバックアップを作成) は使用できません。指定されたディレクトリは、サーバの現在のディレクトリに対する相対パスなので、完全パス名を指定する必要があります。さらに、サーバには指定されたディレクトリへの書き込みパーミッションが必要です。-s を指定すると、バックアップユーティリティは進行状況のメッセージを表示せず、既存のファイルを上書きするときにプロンプトを表示しません。既存のファイルを上書きしようとするときにプロンプトを表示させる場合は、-s または -y を指定しないでください。-k リカバリオプションを指定する場合は、-s オプションも必ず指定してください。</p>
-t	<p>トランザクションログをデータベースファイルの最新のバックアップコピーに対して適用できるので、インクリメンタルバックアップとして使用できるバックアップを作成します。</p>
-wa	<p>トランザクションが完了してトランザクションログの名前変更またはトランケートが行われるまで待機します。-s オプションも使用しないかぎり、-wa オプションは SQL Anywhere 16 以前のサーバではサポートされません。</p>

オプション	説明
-wb	アクティブなトランザクションがなくなるまで、データベースのバックアップを遅らせます。-wb オプションは必ず -s オプションと一緒に使用してください。
-X	既存のトランザクションログをバックアップし、元のログを削除して、新しいトランザクションログを開始します。データベースミラーリングを使用している場合は、このオプションを使用しないでください。
-XO	<p>現在のトランザクションログを削除して、新しいトランザクションログを開始します。この操作では、バックアップは実行されません。この操作の目的は、レプリケーション環境以外の環境でディスク領域を解放することです。データベースミラーリングを使用している場合は、このオプションを使用しないでください。</p> <p><b>警告</b></p> <p>このオプションを使用すると、データベースがメディア障害からリカバリできなくなる可能性があります。このオプションは、データの損失を許容できる場合にのみ使用してください。</p> <p>レプリケートされるデータベースまたは同期されるデータベースでは、-xo オプションを使わないでください。SQL Remote と Mobile Link はトランザクションログ情報に依存します。</p>
-y	確認メッセージを表示することなく、バックアップディレクトリを作成するか、ディレクトリ内の既存のバックアップファイルを置き換えます。既存のファイルを上書きしようとするときにプロンプトを表示させる場合は、-s または -y を指定しないでください。
target-directory	バックアップファイルのコピー先ディレクトリを指定します。このディレクトリが存在しない場合は作成されます。ただし、親ディレクトリが存在していなければなりません。デフォルトでは、バックアップユーティリティは、データベースファイルのクライアント側のバックアップを作成します。-s オプションを指定すると、BACKUP DATABASE 文を使用してサーバ上にバックアップを作成できます。

## 権限

BACKUP DATABASE システム権限が必要です。

## 備考

バックアップユーティリティを使うと、すべてのバックアップコピーを単一データベースに作成することができます。単純なデータベースは、メインデータベースファイルとトランザクションログの 2 つのファイルで構成されます。より複雑なデータベースは、複数のファイルにテーブルを格納できます。各ファイルは個別の DB 領域となります。すべてのバックアップデータベースファイル名はデータベースファイル名と同じです。バックアップユーティリティで作成したイメージのバックアップは、バックアップされた各ファイルの別ファイルで構成されます。

実行中のデータベースに対してバックアップユーティリティを使用するのは、データベースが実行されていないときにデータベースファイルをコピーするのと同じです。バックアップユーティリティを使用すると、他のアプリケーションやユーザがデータベースを使用しているときでも、そのデータベースをバックアップできます。

-d または -t のいずれのオプションも使用されていない場合は、すべてのデータベースファイルがバックアップされます。

デフォルトでは、バックアップユーティリティは、データベースファイルのクライアント側のバックアップを作成します。-s オプションを指定すると、BACKUP DATABASE 文を使用してサーバ上にバックアップを作成できます。

### 警告

データベースとトランザクションログのバックアップコピーには、どのような変更でも加えるべきではありません。バックアップ中に処理中のトランザクションがなかった場合、または BACKUP DATABASE WITH CHECKPOINT LOG RECOVER か WITH CHECKPOINT LOG NO COPY を指定した場合は、読み込み専用モードを使用するか、バックアップデータベースのコピーを検証することによって、バックアップデータベースの妥当性をチェックできます。

一方、トランザクションの処理中だった場合、または BACKUP DATABASE WITH CHECKPOINT LOG COPY を指定した場合は、検証の開始時にデータベースサーバがデータベースのリカバリを実行する必要があります。リカバリを実行するとバックアップコピーに変更が加えられるため、元のデータベースからの以降のトランザクションログファイルが適用されなくなります。この場合は、インメモリ検証を使用してデータベースとトランザクションが変更されないようにするか、データベースのコピーを検証してください。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

### 例

たとえば、次のコマンドは、DBA ユーザとして sample\_host という名前のコンピュータに接続し、このコンピュータで実行中のサンプルデータベースを SQLAnybackup ディレクトリにバックアップします。

```
dbbackup -c "Host=sample_host;DBN=demo;UID=DBA;PWD=sql" SQLAnybackup
```

## 関連情報

[データベースミラーリングシステムでのトランザクションログの管理 \[1678 ページ\]](#)

[データベースのバックアップとリカバリ \[894 ページ\]](#)

[バックアップの種類 \[896 ページ\]](#)

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[設定ファイル \[1066 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) を使用したデータベースのバックアップ \[902 ページ\]](#)

### 1.7.4.3 Broadcast Repeater ユーティリティ (dbns17)

他のサブネット上で実行されているデータベースサーバや、ファイアウォールの反対側において UDP ブロードキャストが通常は届かない SQL Anywhere データベースサーバを、クライアントが検索できるようにします。

#### 構文

```
dbns17 [ options ] [ address ... ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-apport	データベースサーバで使用されるポート番号を指定します。デフォルトのポート番号は 2638 です。
-mip	DBNS プロセスが実行されているコンピュータの IP アドレスを指定します。このパラメータは、複数の IP アドレスがあるコンピュータでは必須です。このアドレスは IPv4 アドレスであることが必要です。
-ofilename	Broadcast Repeater のメッセージウィンドウに表示される出力を指定されたファイルに書き込みます。
-pport	DBNS Broadcast Repeater で使用されるポート番号を指定します。デフォルトは 3968 です。サブネット間にファイアウォールがある場合は、標準のクライアント/サーバ通信用のポート 2638 に加え、Broadcast Repeater ユーティリティが DBNS プロセス間の TCP 接続に使用するポートも開く必要があります。
-pfilename	指定されたファイルにプロセス ID を書き込みます。
-q	クワイエットモードで実行します (メッセージは表示されません)。
-s	新しい DBNS プロセスは別の DBNS プロセスが同じサブネット上で実行されていないかどうかをチェックし、既存の DBNS プロセスが見つかった場合は、エラーを返してから停止します。

オプション	説明
<code>-ud</code>	(UNIX のみ) UNIX オペレーティングシステムで、DBNS Broadcast Repeater をデーモンとして実行します。DBNS Broadcast Repeater をデーモンとして実行する場合は、 <code>-o</code> オプションを指定して、出力情報のログを取ってください。  DBNS Broadcast Repeater をデーモンとして起動すると、現在のユーザの <code>umask</code> 設定によってパーミッションが制御されます。DBNS Broadcast Repeater に適切なパーミッションを付与するため、DBNS Broadcast Repeater を起動する前に <code>umask</code> 値を設定してください。
<code>-ui</code>	(UNIX のみ) X Window サーバがサポートされている Linux で、使用可能な表示がない場合にシェルモードで DBNS Broadcast Repeater を起動します。
<code>-ux</code>	(UNIX のみ) Linux で <i>SQL Anywhere Broadcast Repeater</i> ウィンドウを開きます (X Window Server を使用)。
<code>-xhost</code>	指定するホストで実行中の DBNS プロセスを停止します。IP アドレスとホスト名のどちらでも指定できます。
<code>-z</code>	DBNS Broadcast Repeater をデバッグモードで起動します。デバッグモードで実行すると、受信または送信される各 SQL Anywhere ブロードキャストパケットに対応する行が Broadcast Repeater のメッセージウィンドウに表示されます。デバッグ出力が冗長であるため、接続上の問題がある場合のみデバッグモードを使用してください。
<code>address</code>	DBNS プロセスを実行中または将来実行する他のコンピュータの IP アドレスまたはホスト名を指定します。これによって、DBNS プロセスが互いを検出し、既知のデータベースサーバや他の DBNS プロセスの情報を交換できます。

## 権限

なし。

## 備考

Broadcast Repeater を使用すると、他のサブネット上で実行されている SQL Anywhere データベースサーバや、ファイアウォールの外側にあつて UDP ブロードキャストが通常は届かない SQL Anywhere データベースサーバを、クライアントは Host 接続パラメータや LDAP を使用することなく検索できます。

`address` には、IP アドレスとコンピュータ名のどちらでも指定できます。複数のアドレスを指定する場合は、スペースで区切ります。

このユーティリティは、サポートされている UNIX プラットフォームと、32 ビットおよび 64 ビットのすべての Windows プラットフォームで使用できます。

Broadcast Repeater を使用するためには、クライアントとデータベースサーバで SQL Anywhere 9.0.2 以降が実行されている必要があります。

#### 警告

dbns17 ユーティリティを SQL Anywhere データベースサーバと同じコンピュータ上で実行しないでください。dbns17 またはデータベースサーバが UDP ブロードキャストを受信しなくなる可能性があります。

#### 例

サブネット 10.50.83.255 と 10.50.125.255 のコンピュータがブロードキャストを使用して接続できるようにする場合を想定します。10.50.83.255 サブネット上のコンピュータ (10.50.83.114 のコンピュータ A) と 10.50.125.255 サブネット上のコンピュータ (10.50.125.103 のコンピュータ B) を使用する必要があります。

これら 2 つの各コンピュータで dbns17 を実行し、他方のコンピュータの IP アドレスを渡します。コンピュータ A では次のコマンドを実行します。

```
dbns17 10.50.125.103
```

コンピュータ B では次のコマンドを実行します。

```
dbns17 10.50.83.114
```

いずれかのコンピュータに複数の IP アドレスがある場合は、-m オプションを使用してローカル IP アドレスを指定する必要があります。

コンピュータ A では次のコマンドを実行します。

```
dbns17 -m 10.50.83.114 10.50.125.103
```

## 関連情報

[トラブルシューティング: Broadcast Repeater ユーティリティによるデータベースサーバの検索方法 \[254 ページ\]](#)

[-pf データベースサーバオプション \[458 ページ\]](#)

## 1.7.4.4 証明書作成ユーティリティ (createcert)

X.509 証明書を作成します。

#### 構文

```
createcert [ -r | -s ]
```

オプション	プロンプト名	説明
@data	該当なし	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-bvalue	RSA のキー長を入力してください (512 ~ 16384)	RSA キー長を指定します。512 ~ 16384 ビットの間で長さを選択できます。  このオプションは、-s オプションが指定されている場合は無視されます。
-cfilename	署名者の証明書のファイルパスを入力してください	署名者の証明書のロケーションとファイル名を指定します。  この情報を指定した場合、生成された証明書は署名済み証明書です。この情報を指定しなかった場合、生成された証明書は自己署名ルート証明書です。  このオプションは、-r オプションが指定されている場合は無視されます。
-ca {0   1}	証明書認証局 (Y/N) [N]	他の証明書への署名に使用できる認証局証明書を作成します。デフォルトでは、証明書は認証局ではありません (0)。  このオプションは、-r オプションが指定されている場合は無視されます。
-ckfilename	署名者のプライベートキーのファイルパスを入力してください	証明書要求に関連付けられたプライベートキーを保存するロケーションとファイル名。このオプションのプロンプトは、-c オプションでファイル名を指定した場合のみ表示されます。
-COfilename	証明書を保存するファイル名を入力してください	指定されたファイルにパブリック証明書を書き込みます。  証明書を保存するロケーションとファイル名を指定します。指定しない場合、証明書は保存されません。  このオプションは、-r オプションが指定されている場合は無視されます。

オプション	プロンプト名	説明
<code>-cppassword</code>	署名者のプライベートキーのパスワードを入力してください	署名者のプライベートキーの暗号化に使用されたパスワードを指定します。  このパスワードは、プライベートキーが暗号化されている場合にのみ指定します。  このオプションは、 <code>-r</code> オプションが指定されている場合は無視されます。
<code>-iofilename</code>	ID を保存するファイル名を入力してください	生成された ID を保存するロケーションとファイル名を指定します。  このオプションは、 <code>-r</code> オプションが指定されている場合は無視されます。
<code>-kofilename</code>	プライベートキーを保存するファイル名を入力してください	プライベートキーを保存するロケーションとファイル名を指定します。
<code>-kppassword</code>	プライベートキーを保護するためのパスワードを入力してください	プライベートキーを暗号化するために使用するパスワードを指定します。  パスワードを指定しない場合、プライベートキーは暗号化されません。このオプションは、 <code>-ko</code> オプションを使用してファイル名を指定した場合にのみ適用されます。
<code>-mvalue</code>	シリアル番号 [生成された GUID]	シリアル番号を指定します。シリアル番号は 40 桁以下の 16 進数文字列である必要があります。この番号は、現在の署名者が署名したすべての証明書でユニークである必要があります。シリアル番号を指定しなかった場合、シリアル番号として GUID が生成されます。  このオプションは、 <code>-r</code> オプションが指定されている場合は無視されます。
<code>-p1</code>	該当なし	PKCS #1 を使用して、デフォルトの PKCS #8 の代わりに、暗号化されていない RSA プライベートキーをエンコードします。

オプション	プロンプト名	説明
-r	該当なし	PKCS #10 証明書要求を生成します。 このオプションを指定すると、次のオプションは適用されません。 <ul style="list-style-type: none"> <li>● -c</li> <li>● -ca</li> <li>● -ck</li> <li>● -co</li> <li>● -cp</li> <li>● -m</li> <li>● -io</li> <li>● -u</li> <li>● -x</li> <li>● -v</li> </ul>
-rOfilename	要求を保存するファイルパスを入力してください	PKCS #10 証明書要求の保存に使用するロケーションとファイル名を指定します。 このオプションは、-r オプションを指定した場合にかぎり適用されます。
-sfilename	該当なし	署名される PKCS10 証明書要求のファイル名と場所を指定します。証明書要求は、DER または PEM でエンコードできます。このオプションを指定すると、次のオプションは適用されません。 <ul style="list-style-type: none"> <li>● -b</li> <li>● -ec</li> <li>● -sc</li> <li>● -scn</li> <li>● -sl</li> <li>● -so</li> <li>● -sou</li> <li>● -sst</li> <li>● -t</li> <li>● -x</li> </ul>
-svalue	該当なし	sha1、sha256、sha384、sha512 のうち、どのシグニチャアルゴリズムを使用するかを指定します。 デフォルトのアルゴリズムは sha256 です。
-scvalue	国コード	サブジェクト国コードを指定します。 このオプションは、-s オプションが指定されている場合は無視されます。

オプション	プロンプト名	説明
<code>-scnvalue</code>	通称	サブジェクト通称を指定します。 このオプションは、 <code>-s</code> オプションが指定されている場合は無視されます。
<code>-s/value</code>	地方	サブジェクト地方を指定します。 このオプションは、 <code>-s</code> オプションが指定されている場合は無視されます。
<code>-sovalue</code>	組織	サブジェクト組織を指定します。 このオプションは、 <code>-s</code> オプションが指定されている場合は無視されます。
<code>-souvalue</code>	組織単位	サブジェクト組織単位を指定します。 このオプションは、 <code>-s</code> オプションが指定されている場合は無視されます。
<code>-ssfvalue</code>	都道府県	サブジェクト都道府県を指定します。 このオプションは、 <code>-s</code> オプションが指定されている場合は無視されます。
<code>-ftype</code>	該当なし	暗号化タイプを指定します。RSA を指定します。 このオプションは、 <code>-s</code> オプションが指定されている場合は無視されます。

オプション	プロンプト名	説明
-Uvalue, ...	キーの使用法	<p>証明書の使用方法を指定します。</p> <p>証明書のプライベートキーをどのように使用できるかを示す番号をカンマ区切りのリストで指定します。これは詳細オプションであり、ほとんどの状況ではデフォルト設定で十分です。デフォルトは、証明書が認証局であるかどうかによって異なります。証明書認証局のある証明書のデフォルトは、6,7です。証明書認証局のない証明書のデフォルトは、3,4,5です。</p> <p>このオプションは、-r オプションが指定されている場合は無視されます。</p> <p>次のような値があります。</p> <ul style="list-style-type: none"> <li>• 1. デジタル署名</li> <li>• 2. 否認防止</li> <li>• 3. キーの暗号化</li> <li>• 4. データの暗号化</li> <li>• 5. キーの承諾</li> <li>• 6. 証明書の署名</li> <li>• 7. CRL の署名</li> <li>• 8. 暗号化のみ</li> <li>• 9. 復号化のみ</li> </ul>
-Vyears	証明書の有効期間 (年数) (1-100)	<p>証明書が有効である年数 (1 ~ 100) を指定します。この期間を過ぎると、証明書と、その証明書で署名されたすべての証明書の有効期限が切れます。</p> <p>このオプションは、-r オプションが指定されている場合は無視されます。</p>
-X	該当なし	<p>自己署名証明書を生成します。</p> <p>このオプションは、-r オプションが指定されている場合は無視されます。</p>
-3des	該当なし	<p>3DES 暗号化を使用してプライベートキーを暗号化します。このオプションを使用して、Certicom 暗号化モジュールを実行する旧バージョンのデータベースサーバのほか、OpenSSL 暗号化モジュールを使用するデータベースサーバで使用可能なサーバ ID ファイルを作成します。FIPS モードで実行している場合はこのオプションを使用しないでください。</p>

## 権限

なし。

## 備考

ユーザは、一般的にサードパーティから証明書を購入します。そのような認証局には、証明書を作成するために独自のツールが用意されています。次のツールは、開発用やテスト用の証明書を作成するときに特に便利ですが、稼働時の証明書に使用することもできます。

オプションを指定せず、`createcert` から必要な値を要求されるようにすることで、署名証明書を作成できます。または、証明書に必要なオプションを指定することもできます。

あるユーザが要求を作成して別のユーザがそれに署名するというように、処理を 2 つのステップに分ける場合、1 番目のユーザは `-r` を指定した `createcert` を実行して要求を作成し、2 番目のユーザは `-s` を指定した `createcert` を実行して要求に署名することができます。

ルート証明書 (他の証明書に署名する証明書) を作成する場合は、認証局を使用して自己署名証明書を作成し、デフォルトの署名オプションを使用してください。たとえば、`-ca` オプションに `y` を指定し、`-u` オプションに `6,7` を指定します。

`-3des` オプションに関しては、ソフトウェアバージョン 16.0 のビルド 1704 まで、ソフトウェアバージョン 12.0.1 のビルド 3993 まで、および 12.0.1 以前のすべてのバージョンには、FIPS、TLS および HTTPS で使用される強力な暗号化のための Certicom 暗号化モジュールが含まれていました。このモジュールを使用して暗号化されたプライベートキーは、AES ではなく 3DES 暗号化を使用して暗号化されています。

SQL Anywhere では、PKCS #1、PKCS #8 (デフォルト)、および PKCS #10 がサポートされています。PKCS #12 などの他の形式を使用する場合、OpenSSL ユーティリティを使用して、サポートされる形式に証明書を変換してください。

### 例

次の例では、次の特性を使用して自己署名ルート証明書を作成します。

- 1024 ビット RSA キー
- 生成されたシリアル番号
- 5 年間有効
- 証明書認証局ではない
- デジタル署名、キーの暗号化、データの暗号化、キーの承諾で使用される
- パスワードが `sqlkey` のプライベートキー

```
createcert -t rsa -b 1024 -sc CA -sst ON -sl Waterloo -so MyCompany -sou MyUnit -scn test -x -m 0 -v 5 -ca 0 -u 1,3,4,5 -co root-cert.pem -ko root-key.pem -io root-id.pem -kp sqlkey
```

次の例では、次の特性を使用して `ca` キーで署名された証明書を作成します。

- 1024 ビット RSA キー
- 生成されたシリアル番号
- 5 年間有効
- 証明書認証局ではない

- デジタル署名、キーの暗号化、データの暗号化、キーの承諾で使用される
- パスワードが sqlkey のプライベートキー:

```
createcert -t rsa -b 1024 -sc CA -sst ON -sl Waterloo -so MyCompany -sou MyUnit -scn test -c ca-cert.pem -ck ca-key.pem -cp cakeypass -m 0 -v 5 -ca 0 -u 1,3,4,5 -co root-cert.pem -ko root-key.pem -io root-id.pem -kp sqlkey
```

次の例では、署名済み証明書を作成します。この例では、署名者の証明書にファイル名を指定しないため、自己署名ルート証明書になります。

```
createcert
SQL Anywhere X.509 証明書ジェネレータバージョン 17.0.4.1691
警告: 証明書には、ソフトウェアバージョン 12.0.1 のビルド 3994 以前
およびバージョン 16.0 のビルド 1691 以前の旧バージョンとの
互換性はありません。互換性が必要な場合、-3des スイッチを使用してください。
RSA のキー長を入力してください (512 ~ 16384): 1024
キーペア作成中...
国コード: CA
都道府県: Ontario
地方: Waterloo
組織: MyCompany
組織単位: Engineering
通称: Test Certificate
署名者の証明書のファイルパスを入力してください:
証明書は自己署名ルート
シリアル番号 [生成された GUID]:
生成されたシリアル番号: 3ccd9e6b0c8c45d98f45f8a3f51d6ba0
証明書の有効期間 (年数) (1-100): 10
証明書認証局 (Y/N) [N]: n
1. デジタル署名
2. 否認防止
3. キーの暗号化
4. データの暗号化
5. キーの承諾
6. 証明書の署名
7. CRL の署名
8. 暗号化のみ
9. 復号化のみ
キーの使用法 [1、3、4、5]: 1,3,4,5
証明書を保存するファイル名を入力してください: cert.pem
プライベートキーを保存するファイル名を入力してください: key.pem
プライベートキーを保護するためのパスワードを入力してください: pwd
ID を保存するファイル名を入力してください: id.pem
```

次の例では、デフォルトの署名を使用する認証局による自己署名ルート証明書であるルート証明書 (他の証明書に署名する証明書) を作成します。証明書認証局プロンプトへの応答または -ca オプションは y、[キーの使用法] プロンプトへの応答または -u オプションは 6,7 (デフォルト) とします。

```
証明書認証局 (Y/N) [N]: y
1. デジタル署名
2. 否認防止
3. キーの暗号化
4. データの暗号化
5. キーの承諾
6. 証明書の署名
7. CRL の署名
8. 暗号化のみ
9. 復号化のみ
```

## 関連情報

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

[FIPS 認定の暗号化テクノロジー \[1642 ページ\]](#)

[証明書ビューアユーティリティ \(viewcert\) \[1087 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

### 1.7.4.5 証明書ビューアユーティリティ (viewcert)

パブリックキーインフラストラクチャ (PKI) オブジェクト内の値の表示、PKI オブジェクトのエンコーディングの変換、プライベートキーの暗号化と復号化を行います。

#### 構文

```
viewcert [ options ] input-file
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-d	出力を DER でエンコードします。このオプションは、-o オプションと一緒に指定した場合のみ役立ちます。-p とは一緒に使用できません。デフォルトでは、判読可能なテキスト形式で PKI オブジェクトが出力されます。
-ipinput-password	input-file に暗号化されたプライベートキーが含まれる場合、プライベートキーの復号化に必要なパスワードを指定します。
-ofilename	出力が書き込まれるファイルを指定します。デフォルトでは、viewcert を実行しているコマンドプロンプトウィンドウに出力が書き込まれます。

オプション	説明
<code>-opoutput-password</code>	プライベートキーを暗号化するために使用されるパスワードを指定します。このオプションは、 <code>-d</code> または <code>-p</code> と一緒に指定した場合のみ役立ちます。デフォルトでは、プライベートキーは暗号化されません。
<code>-p</code>	出力を PEM でエンコードします。このオプションは、 <code>-o</code> オプションと一緒に指定した場合のみ役立ちます。 <code>-d</code> とは一緒に使用できません。デフォルトでは、判読可能なテキスト形式で PKI オブジェクトが出力されます。
<code>-p1</code>	PKCS #1 を使用して、デフォルトの PKCS #8 の代わりに、暗号化されていない RSA プライベートキーをエンコードします。
<code>-q</code>	クワイエットモードで実行します (メッセージを表示しません)。
<code>input-file</code>	DER または PEM でエンコードされた PKI オブジェクトである必要があるファイルを指定します。
<code>-3des</code>	<code>-p</code> 、 <code>-ip</code> 、および <code>-op</code> オプションを指定する場合に、AES ではなく 3DES 暗号化を使用することを指定します。FIPS モードで実行している場合はこのオプションを使用しないでください。

## 権限

なし。

## 備考

viewcert ユーティリティを使用して表示できる PKI オブジェクトのタイプは、次のとおりです。

- X.509 証明書
- 証明書要求
- プライベートキー
- 証明書失効リスト (CRL)

viewcert は、DER と PEM の間でエンコードタイプを変換したり、プライベートキーを暗号化または復号化したりするために使用することもできます。

viewcert ユーティリティは RSA オブジェクトをサポートします。

## 例

次の例では、SQL Anywhere に付属するサンプル RSA 証明書を表示できます。`%SQLANY%SAMP17%`  
`¥Certificates` ディレクトリから次のコマンドを実行します。

```
viewcert rsaroot.crt
```

この例は、次の出力を生成します。

```
SQL Anywhere X.509 証明書ビューアバージョン 17.0.0.1245
通称: サンプル RSA ルート証明書
国コード: CA
都道府県: ON
地方: Waterloo
組織: test
組織単位: SQL Anywhere
発行者: サンプル RSA ルート証明書
シリアル番号: 101
発行日時: 2015 年 4 月 19 日 13:49:00
失効: 2025 年 4 月 20 日 13:49:00
シグネチャアルゴリズム: RSA、SHA256
キータイプ: RSA
キーサイズ: 2048 ビット
基本的な制約: 認証局であり、パス長制限は次のとおり: 10
キーの使用法: 証明書の署名、CRL の署名
```

## 関連情報

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

### 1.7.4.6 データソースユーティリティ (dbdsn)

SQL Anywhere ODBC データソースの作成、削除、説明、およびリスト作成を行います。

#### 構文

```
dbdsn [ modifier-options ]
{ -l[ s | u ]
| -d[ s | u ] dsn
| -g[ s | u ] dsn
| -w[ s | u ] dsn [details-options;...]
| -cl }
```

主要オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-[s u]	利用できる ODBC データソースをリストします。リストの形式は -b オプションまたは -v オプションを使用して変更できます。Windows では、u (ユーザ) または s (システム) 指定子を使用して、オプションを変更できます。デフォルトの指定子は u です。
-d[s u] dsn	指定されたデータソースを削除します。-y を指定すると、確認メッセージを表示せずに既存のデータソースが削除されます。Windows では、u (ユーザ) または s (システム) 指定子を使用して、オプションを変更できます。デフォルトの指定子は u です。
-g[s u] dsn	指定されたデータソースの定義を表示します。出力のフォーマットは、-b オプションまたは -v オプションを使用して変更できます。Windows では、u (ユーザ) または s (システム) 指定子を使用して、オプションを変更できます。デフォルトの指定子は u です。
-w[s u] dsn [ details-options ]	新しいデータソースを作成します。同じ名前のデータソースが存在する場合は上書きします。-w オプションと一緒に -c オプションを指定する必要があります。-y を指定すると、確認メッセージを表示せずに既存のデータソースを上書きします。Windows では、u (ユーザ) または s (システム) 指定子を使用して、オプションを変更できます。デフォルトの指定子は u です。
-cl	dbdsn ユーティリティがサポートしている接続パラメータをリストします。

変更オプション	説明
-b	リストの出力を 1 行の接続文字列にフォーマットします。
-cm	データソースの作成に使用するコマンドを表示します。このオプションを使用すると、作成コマンドをファイルに出力できます。作成コマンドは別のコンピュータにデータソースを追加したり、変更が加えられたデータソースを元の状態にリストアするために使用できます。-cm とともに -g オプションまたは -l オプションを指定しないとコマンドは失敗します。-g を指定すると、指定のデータソースの作成コマンドが表示されるのに対し、-l を指定するとすべてのデータソースの作成コマンドが表示されます。  指定のデータソースが存在しない場合は、データソースを削除するコマンドが生成されます。たとえば、コンピュータに mydsn データソースが存在しない場合、dbdsn -cm -g mydsn は mydsn データソースを削除する次のコマンドを返します。  <pre>dbdsn -y -du "mydsn"</pre>

変更オプション	説明
-dr	<p>データソースを表示するときに Driver パラメータを含めます。これにより、-cm オプションを使用してデータソースを再作成するときに、dbdsn の現バージョンで異なるバージョンの ODBC ドライバを参照するデータソースを作成することができます。</p> <p>たとえば、次のようなコマンドを使用してバージョン 11 のデータソースを作成します。</p> <pre>dbdsn -y -wu "11.0 Student Sample" -c "UID=DBA;PWD=sql;...;Driver=SQL Anywhere 11"</pre> <p>dbdsn -cm -l を実行すると、同じコマンドが Driver= パラメータなしで表示されます。これによって、SQL Anywhere バージョン 17 の ODBC ドライバを使用して ODBC データソースが再作成されます。</p> <p>それに対し、dbdsn -dr -cm -l を実行すると、コマンドに "Driver=parameter" が含まれるため、バージョン 11 の ODBC ドライバを使用した場合とまったく同じデータソースを再作成できます。</p>
-f	<p>使用されているシステムファイルの名前を表示します。このオプションは UNIX でのみ使用できます。</p>
-ns	<p>環境変数の設定を使用してシステム情報ファイル (デフォルトのファイル名は .odbc.ini) のロケーションを特定するように指定します。このオプションは、使用される可能性のあるシステム情報ファイルが複数ある場合に、dbdsn が使用するファイルを特定する目的にも利用できます。このオプションは UNIX でのみ使用できます。</p> <p>データソースを作成するときに -ns オプションを指定しなかった場合、システム情報ファイルはユーザのホームフォルダとそのパス内で検索されます。</p>
-Ofilename	<p>指定したファイルに、出力メッセージを追加します。</p>
-or	<p>-c オプションとともに指定すると、SQL Anywhere17 - Oracle ドライバのデータソースが作成されます。例:</p> <pre>dbdsn -w MyOracleDSN -or -c Userid=DBA;Password=passwd;ServiceName=abcd;ArraySize=500;ProcResults=y</pre> <p>-cl オプションを -or オプションとともに指定すると、SQL Anywhere17 - Oracle ドライバの接続パラメータのリストを取得できます。</p>

変更オプション	説明
-pe	<p>Password (PWD) 接続パラメータで指定されたパスワードを難読化し、EncodedPassword (ENP) 接続パラメータを使用して、難読化されたパスワードをデータソースに格納します。難読化されたパスワードはどのコンピュータでも使用可能で、セキュリティは確保されません。このオプションは廃止予定で、推奨されません。</p>
-pet a   c   u	<p>Password (PWD) 接続パラメータで指定されたパスワードをエンコードし、ENP (EncodedPassword) 接続パラメータを使用して、エンコードされたパスワードをデータソースに格納します。</p> <p>-pet a 難読化パスワードはすべてのコンピュータで使用できます。エンコードパスワード文字列とそれに対応するユーザ ID は、このデータベースへの接続を開始するために、どのユーザでも、どのコンピュータ上でも使用できます。</p> <p>-pet c パスワードは、エンコードが行われたのと同じコンピュータ上でのみデコードできます。エンコードパスワード文字列とそれに対応するユーザ ID は、エンコードパスワードが作成されたコンピュータからデータベースへの認証を行う場合のみ使用できます。このコンピュータにログインできるユーザであれば誰でも、エンコードパスワードとそれに対応するユーザ ID を使用してデータベースへの認証を実行できます。他のコンピュータでは使用できません。</p> <p>-pet u パスワードは、エンコードが行われたのと同じコンピュータ上で、同じユーザによってのみデコードできます。エンコードパスワード文字列とそれに対応するユーザ ID は、エンコードパスワードが作成されたコンピュータから、作成したユーザがデータベースへの認証を行う場合のみ使用できます。同じコンピュータの他のユーザはデコードすることができません。同じユーザであっても他のユーザであっても、他のコンピュータ上ではデコードすることができません。</p>
-q	<p>データベースサーバメッセージウィンドウに出力を表示しないようにします。データソースの削除時または変更時に -q を指定する場合は、-y も指定してください。</p>
-v	<p>数行のリスト出力を表としてフォーマットします。</p>
-y	<p>確認を要求するプロンプトを表示しないで、各データソースを削除または上書きします。データソースの削除時または変更時に -q を指定する場合は、-y も指定してください。</p>
詳細オプション	説明
-c"keyword=value:..."	<p>接続パラメータを接続文字列として指定します。</p>
-cw	<p>(-c で指定された) DBF パラメータが確実に絶対ファイル名になるようにします。DBF の値が絶対ファイル名でない場合、dbdsn は現在の作業ディレクトリ (CWD) を付加します。オペレーティングシステムの中には、バッチファイルですぐに利用できる CWD 情報がないものもあるので、このオプションは便利です。</p>

## 権限

なし。

## 備考

変更オプションは、主要オプションの前または後に指定できます。

接続文字列または SQLCONNECT 環境変数で同じパラメータが直接指定されている場合、DSN に保存された接続パラメータは使用されません。

データソースユーティリティは、プラットフォームを問わないユーティリティで、ODBC データソースアドミニストレータに代わり、ODBC データソースの作成、削除、記述、リストを実行します。このユーティリティは、バッチ処理に役立ちます。

### 警告

パスワードはデータソースに保存しないでください。ODBC データソースにプレーンテキストのパスワードを保存すると、ODBC データソースが保存されているコンピュータのロケーションに対する不正なアクセスを防ぐための手順が及ばず、セキュリティが確保されません。パスワードやデータベースキーは、入力を求めることをおすすめします。

-pet オプションを使用すると、プレーンテキストでないように、エンコードされたパスワードがデータソースに保存されます。データソースの ENP (EncodedPassword) フィールドには、16 進文字列としてエンコードされたパスワードが含まれます。エンコードされた文字列は、接続の要求時にクライアントソフトウェアによってプレーンテキストに複合化されます。これは、接続が試行される前にクライアントコンピュータで実行されることに注意してください。

-pet c および -pet u オプションは、エンコードされたパスワードの使用を目的とするコンピュータまたはコンピュータ/ユーザでのみ使用できます。

Microsoft Windows System データソースでは -pet u を使用できますが、-pet c の方が適している場合もあります。-pet c のエンコードパスワードは、コンピュータのどのユーザでも使用できるためです。システムデータソースが Microsoft Windows ユーザ 1 人によってのみ作成および使用される場合は、-pet u を使用することができます。

### 注記

Microsoft Windows オペレーティングシステムでは、データソースはレジストリまたはディスクに保持されます。データソースユーティリティでは、FILEDSN を使用したファイルベースのデータソースの使用をサポートしていません。

32 文字より長いデータソース名は作成できません。

### 注記

UNIX オペレーティングシステムでは、データソースはシステム情報ファイル (デフォルトのファイル名は .odbc.ini) に保持されます。データソースユーティリティを使用して ODBC データソースを UNIX 上で作成または削除すると、システム情報ファイルの [ODBC Data Sources] セクションが自動的に更新されます。UNIX で -c オプションを使用して Driver 接続パラメータを指定しない場合、データソースユーティリティによって SQLANY17 環境変数の設定に基づき SQL Anywhere ODBC ドライバのフルパスを使って Driver エントリが自動的に追加されます。

SQL Anywhere データソースのみを使用する場合以外は、UNIX でファイル非表示ユーティリティ (dbfhide) を使ってシステム情報ファイル (.odbc.ini) をエンコードしないでください。他のデータソース (Mobile Link 同期など) を使用する予定の場合、システム情報ファイルをエンコードすると、他のドライバが正しく機能しなくなることがあります。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 例

ユーザデータソース newdsn の定義を書き込みます。データソースがすでに存在する場合、上書きするかどうかを確認するプロンプトが表示されます。

```
dbdsn -w newdsn -c "Host=myhost;Server=myserver;UID=DBA"
```

ユーザデータソース newdsn の定義を書き込みます。データソースがすでに存在する場合、上書きするかどうかを確認するプロンプトが表示されません。

```
dbdsn -y -w newdsn -c "Host=myhost;Server=myserver;UID=DBA"
```

エンコードされたパスワードを含むユーザデータソース newdsn の定義を書き込みます。データソースは、そのデータソースが作成されたコンピュータ上のユーザによってのみ使用できます。

```
dbdsn -pet u -w newdsn -c "Host=myhost;Server=myserver;UID=DBA;PWD=sql"
```

既知のすべてのユーザデータソースをリストします (1 行に 1 つのデータソース名)。

```
dbdsn -l
```

既知のすべてのシステムデータソースをリストします (1 行に 1 つのデータソース名)。

```
dbdsn -ls
```

すべてのユーザデータソースを関連する接続文字列とともにリストします。

```
dbdsn -l -b
```

ユーザデータソース MyDSN 用の接続文字列をレポートします。

```
dbdsn -g MyDSN
```

システムデータソース MyDSN 用の接続文字列をレポートします。

```
dbdsn -gs MyDSN
```

最初に BadDSN の接続パラメータをリストし、確認メッセージを表示してから、ユーザデータソース BadDSN を削除します。

```
dbdsn -d BadDSN -v
```

確認メッセージを表示せずに、ユーザデータソース BadDSN を削除します。

```
dbdsn -d BadDSN -y
```

dsninfo.txt への出力のログを取り、すべてのユーザデータソースをリストします。

```
dbdsn -l -o dsninfo.txt
```

すべての接続パラメータ名とそのエイリアスをリストします。

```
dbdsn -cl
```

dsninfo.txt への出力のログを取り、すべての接続パラメータ名をリストします。

```
dbdsn -cl -o dsninfo.txt
```

相対ファイル名を指定します。ODBC データソースが作成される場合、DatabaseFile 接続パラメータには、DBF=c:¥SQLAnywhere17 (現在のディレクトリの場合) のような絶対パスが含まれます。例:

```
c:¥SQLAnywhere17> dbdsn -w testdsn -cw -c UID=DBA;Server=SQLAny;DBF=my.db
```

次のコマンドを実行して、SQL Anywhere 17 Demo データソース定義を読み込み、その定義を restoredsn.bat というファイルに出力します。

```
dbdsn -cm -gs "SQL Anywhere 17 Demo" > restoredsn.bat
```

restoredsn.bat ファイルには以下が含まれています。

```
dbdsn -y -ws "SQL Anywhere 17 Demo" -c "UID=DBA;DBF='C:¥Users¥Public¥Documents
¥SQL Anywhere 17¥Samples¥demo.db';
ServerName=demo17;START='C:¥Program Files¥SQL Anywhere 17¥Bin64¥dbeng17.exe';
ASTOP=YES;Description='SQL Anywhere 17 Sample Database'"
```

次のコマンドは、UNIX のシステム情報ファイルのロケーションを返します。

```
dbdsn -f
```

このコマンドは次の出力を返します。

```
dbdsn using /home/user/.odbc.ini
```

システム情報ファイルのロケーションを変更します。

```
export ODBCINI=./myodbc.ini
```

dbdsn -f を再実行し、システム情報ファイルの新しいロケーションをレポートします。

```
dbdsn using ./myodbc.ini
```

データソースを作成し、環境変数の設定を使用してシステム情報ファイルのロケーションを特定する場合、-ns オプションを使用してください。

```
dbdsn -w NewDSN -c "UID=DBA;PWD=passwd" -ns
```

このコマンドは次の出力を返します。

```
Configuration "newdsn" written to file ./myodbc.ini
```

このセクションの内容:

[ODBC 接続パラメータ \[1096 ページ\]](#)

データソースユーティリティ (dbdsn) は次の ODBC 接続パラメータをサポートしています。

## 関連情報

[設定ファイル \[1066 ページ\]](#)

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[ODBC データソース \[111 ページ\]](#)

[UNIX での ODBC データソース \[121 ページ\]](#)

[ODBC データソースの作成 \(ODBC データソースアドミニストレータの場合\) \[112 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

### 1.7.4.6.1 ODBC 接続パラメータ

データソースユーティリティ (dbdsn) は次の ODBC 接続パラメータをサポートしています。

ブール (true または false) 引数は、true の場合は YES または 1、false の場合は NO または 0 のいずれかです。

名前	説明
Delphi	Delphi では、1 ローにつき複数のブックマーク値を処理できません。この値を NO に設定すると、各ローに 1 つ (YES の場合は 2 つ) のブックマーク値が割り当てられます。このオプションを YES に設定すると、スクロール可能なカーソルのパフォーマンスが向上します。

名前	説明
DescribeCursor	<p>このパラメータでは、プロシージャが呼び出されたときにカーソルを再記述する頻度を指定できます。デフォルト設定は [任意] です。</p> <p><b>Never</b></p> <p>カーソルを再記述する必要がないことがわかっている場合は、0、N、または NO を指定します。カーソルの再記述は負荷がかかり、パフォーマンスが低下することがあります。</p> <p><b>If Required</b></p> <p>カーソルを再記述する必要があるかどうかを ODBC ドライバに判断させる場合は、1、Y、または YES を指定します。プロシージャに RESULT 句がある場合、ODBC アプリケーションはカーソルが開いた後に結果セットを再記述しなくなります。これはデフォルト設定です。</p> <p><b>Always</b></p> <p>2、A、または ALWAYS を指定した場合、カーソルは開くたびに再記述されます。Transact-SQL プロシージャ、または複数の結果セットを返すプロシージャを使用する場合は、カーソルが開くたびにカーソルを再記述する必要があります。</p>
Description	このパラメータにより、ODBC データソースの説明を入力できます。
Driver	<p>このパラメータでは、接続の ODBC ドライバを次の形式で指定できます。Driver=<b>driver-name</b> デフォルトで使用されるドライバは SQL Anywhere17 です。<b>driver-name</b> は、SQL Anywhere<i>x</i> にしてください。その場合、<i>x</i> にはソフトウェアのメジャーバージョン番号を指定します。<b>driver-name</b> が <i>SQL Anywhere</i> で始まらない場合、データソースユーティリティ (dbdsn) によって読み取ることができません。</p> <p>UNIX では、このパラメータによって共有オブジェクトへの完全に修飾されたパスが指定されます。UNIX で Driver 接続パラメータを指定しない場合、データソースユーティリティによって SQLANY17 環境変数の設定に基づき SQL Anywhere ODBC ドライバのフルパスを使って Driver エントリが自動的に追加されます。</p>

名前	説明
Escape	<p>このパラメータは、テーブルまたはカラムのリストを返すときに ODBC ドライバで生成される、SQL 文の LIKE 句で使用されるエスケープ文字を指定します。</p> <p>デフォルトでは、ODBC ドライバでチルダ文字 (~) が使用されますが、一部のアプリケーションではエスケープ文字に円記号 (¥) が使用されることもあります。</p> <p>次の接続文字列フラグメントは、エスケープ文字に円記号 (¥) を指定します。</p> <pre data-bbox="762 689 1355 792">"DSN=SQL Anywhere 17 Demo;UID=DBA;PWD=sql;ESCAPE=¥ ¥;Host=myhost"</pre>
GetTypeInfoChar	<p>このオプションを YES に設定すると、CHAR カラムは SQL_VARCHAR ではなく SQL_CHAR として返されます。デフォルトでは、CHAR カラムは SQL_VARCHAR として返されます。</p>
InitString	<p>InitString により、接続確立後すぐに実行されるコマンドを指定できます。たとえば、データベースオプションを設定したり、ストアプロシージャを呼び出ししたりできます。</p>

名前	説明
IsolationLevel	<p>以下の値の1つを指定して、データソースの初期独立性レベルを設定できます。</p> <p><b>0</b></p> <p>これはコミットされない読み出し独立性レベルとも呼ばれます。これはデフォルトの独立性レベルです。これは最大レベルの同時実行性を提供しますが、結果セットにダーティリード、繰り返し不可能読み出し、幻ローが発生する場合があります。</p> <p><b>1</b></p> <p>これはコミットされた読み出しレベルとも呼ばれます。レベル0よりも低い同時実行性になりますが、レベル0の結果セットに見られる不整合性が一部解消されます。繰り返し不可能ローや幻ローが発生することはありませんが、ダーティリードは発生しません。</p> <p><b>2</b></p> <p>これは繰り返し可能読み出しレベルとも呼ばれます。幻ローが発生することがあります。ダーティリードと繰り返し不可能ローは発生しません。</p> <p><b>3</b></p> <p>これは、直列化可能レベルとも呼ばれます。これは最低レベルの同時実行性を提供する、最も厳しい独立性レベルです。ダーティリード、繰り返し不可能読み出し、幻ローは発生しません。</p> <p><b>Snapshot</b></p> <p>この独立性レベルを使用するには、データベースのスナップショットアイソレーションを有効にする必要があります。スナップショットアイソレーションのレベルは、読み込みと書き込み間の干渉を防ぎます。書き込みは相互に干渉する可能性があります。競合の場合、一貫性のない動作が多少生じる可能性があります。パフォーマンスは独立性レベルを0に設定した場合と同じです。</p> <p><b>Statement-snapshot</b></p> <p>この独立性レベルを使用するには、データベースのスナップショットアイソレーションを有効にする必要があります。スナップショットアイソレーションのレベルは、読み込みと書き込み間の干渉を防ぎます。書き込みは相互に干渉する可能性があります。競合の場合、一貫性のない動作が多少生じる可能性があります。パフォーマンスは独立性レベルを0に設定した場合と同じです。</p> <p><b>Readonly-statement-snapshot</b></p> <p>この独立性レベルを使用するには、データベースのスナップショットアイソレーションを有効にする必要があります。スナップショットアイソレーションのレベルは、読み込みと書き込み間の干渉を防ぎます。書き込みは相互に干渉する可能性があります。</p>

名前	説明
	競合の場合、一貫性のない動作が多少生じる可能性があります。パフォーマンスは独立性レベルを 0 に設定した場合と同じです。
KeysInSQLStatistics、MSApplications、MSApp	SQLStatistics 関数によって外部キーが返されるようにするには、このパラメータを YES に設定します。ODBC 仕様では、SQLStatistics によってプライマリーキーと外部キーが戻されないように指定しています。ただし、一部の Microsoft アプリケーション (Microsoft Visual Basic や Microsoft Access など) では、SQLStatistics によってプライマリーキーと外部キーが戻されることを前提にしています。MSApplications と MSApp は KeysInSQLStatistics の同意語です。
LazyAutocommit	文が完了するまでコミット操作を遅延させるには、このパラメータを YES に設定します。
MatView	マテリアライズドビューは通常、SQLTables などのスキーマ関数によって VIEW としてレポートされます。マテリアライズドビューとビューを区別したい場合は、いくつかの文字列値に対してこのパラメータを設定してください。たとえば、MatView="MAT VIEW" と設定すると、ODBC ドライバはマテリアライズドビューを MAT VIEW としてレポートします。
PrefetchOnOpen	PrefetchOnOpen が YES に設定されていると、カーソルを開く要求を含むプリフェッチ要求が送信されます。プリフェッチにより、カーソルを開くたびにネットワーク要求が行われることはなくなります。カーソルを開くときにプリフェッチを実行するには、カラムをバインドしておきます。この接続パラメータにより、クライアント/サーバ要求の数が削減され、LAN や WAN を介したパフォーマンスが向上します。
PreventNotCapable	SQL Anywhere ODBC ドライバは、修飾子をサポートしていないためエラーを返します。ODBC アプリケーションの中には、このエラーを適切に処理しないものもあります。このようなアプリケーションでも作業できるように、このエラーコードが返されないようにするには、このパラメータを YES に設定します。
SuppressInfoForDataTypes	<p>このパラメータにデータ型名のカンマ区切りリストを設定すると、指定したデータ型についての情報が ODBC ドライバから返されなくなります。指定したデータ型は SQLGetTypeInfo 関数または DatabaseMetaData.getTypeInfo 関数からは返されません。ただし、アプリケーションではカラム宣言と結果セットクエリでデータ型を使用できます。たとえば、次の例は接続文字列の一部ですが、NVARCHAR、LONG VARCHAR、VARBIT の各データ型についての情報が ODBC ドライバから返されなくなります。</p> <pre>SuppressInfoForDataTypes=nvarchar, long nvarchar, varbit</pre>

名前	説明
SuppressWarnings	このパラメータを YES に設定すると、フェッチ時にデータベースサーバから返される警告メッセージが表示されなくなります。バージョン 8.0.0 以降のデータベースサーバでは、それよりも前のバージョンのソフトウェアに比べて多様なフェッチ警告が返されます。以前のバージョンのソフトウェアを使用して配備されたアプリケーションに対して、フェッチの警告を適切に処理するためにこのオプションを選択できます。
TranslationDLL	このオプションは下位互換性のために提供されています。トランスレータの使用はお奨めできません。
TranslationName	このオプションは下位互換性のために提供されています。トランスレータの使用はお奨めできません。
TranslationOption	このオプションは下位互換性のために提供されています。トランスレータの使用はお奨めできません。

## 1.7.4.7 dbisqlc ユーティリティ (廃止予定)

データベースで SQL 文を実行します。

### i 注記

dbisqlc ユーティリティでは、SQL でサポートされるすべての SQL 文と機能がサポートされているわけではなく、現バージョンのデータベースサーバで使用できるすべての機能がサポートされない場合もあります。Interactive SQL ユーティリティを使用します。

### 構文

```
dbisqlc [ options ] [ dbisqlc-statement | dbisql-script-file ]
```

オプション	説明
<code>-C"keyword=value;..."</code>	接続パラメータを指定します。Interactive SQL が接続できない場合は、接続パラメータを入力するウィンドウが表示されます。
<code>-ddelimiter</code>	コマンドデリミタを指定します。デリミタを囲む引用符は省略可能ですが、コマンドシェル自体がデリミタを特別な方法で解釈するときは必ず指定します。  指定したコマンドデリミタは、現在の dbisqlc セッションのすべての接続で使用できます。

オプション	説明
-q	出力メッセージを表示しません。これは、文またはスクリプトファイルを使って Interactive SQL を起動したときのみ便利です。このオプションを指定した場合、エラーメッセージは表示されますが、次の情報は表示されません。 <ul style="list-style-type: none"> <li>警告およびその他の致命的でないメッセージ</li> <li>結果セットの出力</li> </ul>
-x	文をスキャンしますが、実行しません。長いスクリプトファイルの構文エラーをチェックする場合に有用です。

## 権限

なし。

## 備考

dbisqlc ユーティリティを使用すると、SQL 文を入力したり、Interactive SQL スクリプトファイルを実行したりすることができます。

dbisqlc-statement を指定すると、dbisqlc がその文を実行します。スクリプトファイル名も指定できます。dbisqlc-statement または dbisql-script-file 引数を指定しなかった場合、dbisqlc は対話型モードになります。このモードでは、文をコマンドウィンドウに入力できます。

### 例

次のコマンドを実行すると、ユーザ ID DBA とパスワード passwd で、現在のデフォルトサーバに対してスクリプトファイル mycom.sql が実行されます。スクリプトファイルにエラーがあった場合は、処理は停止します。

```
dbisqlc -c "UID=DBA;PWD=passwd" mycom.sql
```

次のコマンドを実行すると、現在のデフォルトデータベースにユーザが追加されます。

```
dbisqlc -c "UID=DBA;PWD=passwd" CREATE USER joe IDENTIFIED BY joespasswd
```

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[Interactive SQL ユーティリティ \(dbisql\) \[1125 ページ\]](#)

## 1.7.4.8 消去ユーティリティ (dberase)

データベースファイルと関連するトランザクションログを消去します。または、個々のトランザクションログファイルを消去します。

### 構文

```
dberase [ options ] database-file
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-ekkey	強力に暗号化されているデータベースの暗号化キーをコマンドに直接指定します。データベースが強力に暗号化されている場合、データベースまたはトランザクションログを使用するには必ず暗号化キーを指定します。強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。強力に暗号化されたデータベースでは、正しいキーを指定しないとコマンドが失敗します。
-ep	暗号化キーの入力を求めるプロンプトを表示するよう指定します。このオプションを指定すると、暗号化キーを入力するためのウィンドウが表示されます。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。強力に暗号化されたデータベースでは、正しいキーを指定しないとコマンドが失敗します。
-ofilename	指定したファイルに、出力メッセージを追加します。
-q	クワイエットモードで実行します (メッセージを表示しません)。このオプションを指定する場合、-y も指定しないと操作は失敗します。
-y	確認メッセージを表示することなく、各ファイルを削除します。-q を指定する場合、-y も指定しないと操作は失敗します。

### 権限

なし。

## 備考

消去ユーティリティを使って、データベースファイルと、それに関連するトランザクションログを消去できます。または、トランザクションログファイルやトランザクションログミラーファイルを消去できます。すべてのデータベースファイルとトランザクションログファイルに読み込み専用のマークを付けて、データベースが突然損傷を受けたり、データベースファイルが不用意に削除されないようにします。

`database-file` は、データベースファイルまたはトランザクションログファイルです。ファイル名は、拡張子も含めてすべて指定してください。データベースファイルを指定すると、関連するトランザクションログファイルが (ミラーもある場合はそれも含めて) 消去されます。

### i 注記

消去ユーティリティは DB 領域を消去しません。DB 領域を消去するには、DROP DBSPACE 文を使用します。

他の DB 領域を参照するデータベースファイルを削除しても、DB 領域ファイルが自動的に削除されることはありません。DB 領域ファイルを自分で消去するには、ファイルを読み込み専用から書き込み可能に変更した後、ファイルを個別に削除します。または、DROP DATABASE 文か **データベース消去ウィザード** を使用して、データベースと関連する DB 領域ファイルを消去することもできます。

データベースファイルを消去すると、関連するトランザクションログとトランザクションミラーも削除されます。トランザクションログミラーも保有しているデータベースのトランザクションログを消去しても、ミラーは削除されません。

このユーティリティの使用中に、消去するデータベースを起動しないでください。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 関連情報

[設定ファイル \[1066 ページ\]](#)

[データベースの削除 \(dberase ユーティリティの場合\) \[301 ページ\]](#)

[データベースの削除 \(SQL Central の場合\) \[300 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

### 1.7.4.9 イベントトレースデータ (ETD) ファイル管理ユーティリティ (dbmanageetd)

ユーザ定義のイベントとシステムイベントの情報を確認できる診断ログを生成します。イベントの説明を参照するには、`sp_trace_events` システムプロシージャを使用します。

### 構文

```
dbmanageetd [ options ] [ filtering-options ] [ information-options ] [ output-format-options ] [ output-redirectation-options ] filename [ filename2 ... ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-m	複数の診断ファイルをマージします。
-s	診断の概要を表示します。-s オプションは -etd 出力形式オプションと一緒に使用できません。
-h	使用方法のヘルプを表示します。

フィルタリングオプション	説明
-fevent-name, ...	イベント名でフィルタリングします。
-fhost, ...	ホスト名でフィルタリングします。
-flseverity-level	重要度レベルでフィルタリングします。有効な重要度レベルは、ALWAYS、CRITICAL、ERROR、WARNING、INFORMATION または DEBUG です。たとえば、WARNING には ALWAYS、CRITICAL、ERROR が含まれますが、INFORMATION と DEBUG は含まれません。
-froffset, count	レコード範囲でフィルタリングします。offset は、0 から始まるオフセットです。count は、返されるレコードの数です。
-ftfrom, to	タイムスタンプの範囲でフィルタリングします。  タイムスタンプには、YYYY-MM-DDThh:mm:ss.sssTZD の形式を使用します。ここで、TZD はタイムゾーン指定子 (Z または +hh:mm または -hh:mm) です。  Z を使用して UTC オフセットを指定するか、+hh:mm または -hh:mm を使用してタイムゾーンの調整を指定します。
-fregexregex	イベントデータを正規表現でフィルタリングします (Windows と Linux のみ)。

情報オプション	説明
-all	すべての情報 (プロセス ID、スレッド ID、ホスト名、ホスト情報、ファイル名) を表示します。
-epoch	時間をミリ秒の Epoch 形式で表示します。
-filename	ファイル名を表示します。
-hostinfo	ホスト情報を表示します。この情報には、オペレーティングシステムのバージョンとプロセッサタイプが含まれます。
-hostname	ホスト名のフィールドを表示します。
-pid	プロセス ID のフィールドを表示します。

情報オプション	説明
<code>-tid</code>	スレッド ID のフィールドを表示します。
<code>-tzadjminutes</code>	タイムゾーンの調整値を分で指定します。デフォルトは、ユーティリティを実行しているコンピュータのローカルの時間です。
<code>-utc</code>	時間を UTC 形式で表示します。

出力形式のオプション	説明
<code>-text</code>	出力をオペレーティングシステムの文字セットで表示します。この値はデフォルトです。 <code>-te</code> オプションでエンコードを指定しないかぎり、オペレーティングシステムの文字セットが使われます。
<code>-etd</code>	出力を ETD ファイル形式で表示します。このオプションと一緒に <code>-o</code> オプションを指定してください。 <code>-etd</code> オプションは、 <code>-s</code> オプションとは一緒に使用できません。
<code>-xml</code>	出力を、UTF-8 を使用して XML 形式で表示します。
<code>-le</code>	利用できる文字セットエンコードをリストします。拡張リストを表示するには <code>+</code> を指定します。
<code>-teencoding</code>	<code>-text</code> 出力形式の文字セットエンコードを指定します。デフォルトでは、オペレーティングシステムの文字セットが使われます。

出力リダイレクションのオプション	説明
<code>-ofilename</code>	出力の保存先のファイル名を指定します。

## 前提条件

情報をイベントトレースデータファイルに記録したイベントトレースセッションは設定済である必要があります。

## 権限

なし。

## 備考

`-fe` と `-fh` のフィルタリグオプションでは、カンマ区切りのリスト値が許可されます。

`-o` オプションを指定しないかぎり、出力は画面上に表示されます。

## 例

診断情報の概要を表示する - 次のコマンドを実行して、診断情報の概要を表示します。

```
dbmanageetd -s diagnostic-log.etc
```

エラー以上の重要度レベルのイベント一覧を表示する - -fl オプションを使用します。例:

```
dbmanageetd -fl error diagnostic-log
```

.etc ファイルを .xml ファイルに変換する - -xml オプションと -o オプションを使用します。例:

```
dbmanageetd -xml -o merged.xml diagnostic-log.etc
```

特定のホストのイベントのみを表示する - ホスト MyHost のイベントのみを表示するには、-fh オプションを使用します。例:

```
dbmanageetd -fh MyHost diagnostic-log.etc
```

さまざまなホストから複数の .etc ファイルをマージする - -m オプションを使用します。例:

```
dbmanageetd -m -etc -o merged.etc machine1.etc machine2.etc machine3.etc  
machine4.etc
```

タイムスタンプの範囲でフィルタリングしたイベントを表示する - EST の 2012 年 3 月 26 日の午前 1:30 から午前 2:00 の間のイベントを表示するには、-ft オプションを使用します。例:

```
dbmanageetd -ft 2012-03-26T01:30:00.000-04:00,2012-03-26T02:00:00.000-04:00  
diagnostic-log.etc
```

## 関連情報

[イベントのトレース \[964 ページ\]](#)

[イベントトレースデータ \(ETD\) 診断ログファイルの内容の表示 \[969 ページ\]](#)

### 1.7.4.10 ファイル非表示ユーティリティ (dbfhide)

難読化または単純暗号化を使用して、設定ファイルと初期化ファイルの内容を非表示にします。

#### 構文

```
dbfhide [ options ] original-configuration-file encrypted-configuration-file
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。

オプション	説明
<code>-q</code>	クワイエットモードで実行します (メッセージは表示されません)。
<code>-w</code>	このコンピュータで現在のユーザによってのみ使用できるようにファイルをエンコードします。エンコードされたファイルは、このコンピュータの別のユーザが使用することも、別のコンピュータで使用することもできません。
<code>-wm</code>	このコンピュータでのみ、すべてのユーザによって使用できるようにファイルをエンコードします。エンコードされたファイルは、他のコンピュータでは使用できません。
<code>original-configuration-file</code>	エンコードされていない元のファイルの名前を指定します。
<code>encrypted-configuration-file</code>	エンコードされた新しいファイルの名前を指定します。

## 権限

なし。

## 備考

一部のユーティリティでは、設定ファイルを使用してコマンドラインオプションを保持します。

デフォルトでは、dbfhide は指定されたファイルで単純な難読化を使用します。単純な難読化は、設定ファイルが不用意に直接アクセスされた場合にデータが表示されないようにすることだけを目的としています。これにより第三者は、設定ファイルの内容を解読することが不可能とまではいきませんが、困難になります。

Microsoft Windows では、`-w` および `-wm` オプションは Microsoft Cryptography API に加えて、統合された Microsoft キーストアを使用して強力な暗号化を実現します。この API は、暗号化キーを特定のユーザおよびコンピュータ (`-w`) または特定のコンピュータ (`-wm`) と関連付けます。

UNIX では、`-w` および `-wm` オプションは安全でない難読化アルゴリズムを使用します。

`-w` または `-wm` オプションを指定する場合、ターゲットとなるコンピュータでファイルをエンコードする必要があります (たとえば、クライアントソフトウェアのインストール時など)。これらのオプションのいずれかを使用してコンピュータ A でエンコードされたファイルは、コンピュータ B では使用できません。`-w` および `-wm` オプションのどちらも使用しない場合、難読化されたファイルを他のコンピュータで使用できます。

### 警告

パスワードを設定ファイルに保存しないことをおすすめします。設定ファイルにパスワードを保存すると、セキュリティに関するリスクが生じます。ユーザ ID とパスワードの両方が発見されると、そのデータベースサーバへのアクセス権を持つユーザは、そのユーザ ID とパスワードを使用して認証することが可能です。エンコードされた設定ファイルの内容は、Microsoft Windows で `-w` または `-wm` オプションが使用される場合にのみ安全です。エンコードされた設定ファイルの内容は、Linux、Mac OS X およびその他のシステムでは安全ではありません。したがって、設定ファイルは適切なオペレーティングシステムの権限によって保護される必要があります。

エンコードされたファイルの内容はテキストエディタでは変更できず、また、エンコードされたファイルを単純にデコードすることはできません。したがって、後で参照できるように、元のプレーンテキストファイルを保存しておく必要があります。エンコードされたファイルを変更するには、元のプレーンテキストファイルを編集し、そのファイルを使用して、エンコードされたファイルを再作成する必要があります。

## 例

データベースサーバに **Elora** という名前を付けて、キャッシュを 10 MB に設定し、サンプルデータベースを起動する設定ファイルを作成します。次のように設定ファイルを作成します。

```
# Configuration file for server Elora
-n Elora
-c 10M
"C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

# で始まる行は、コメントとして処理されます。

ファイルに `sample.txt` という名前を付けます。この設定ファイルを使用してサンプルデータベースを起動するには、次のように入力します。

```
dbsrv17 @sample.txt
```

設定ファイルをエンコードできるようになりました。

```
dbfhide sample.txt sample.enc
```

`sample.enc` ファイルを使用して、サンプルデータベースを起動します。

```
dbsrv17 @sample.enc
```

Windows 環境変数で指定されたオプションを使用して、設定ファイルをエンコードします。

```
set hideopts=-q -wm
dbfhide @hideopts sample.txt sample.enc
```

## 関連情報

[設定ファイルとデータベースサーバの起動オプション \[322 ページ\]](#)

[データのセキュリティ \[1594 ページ\]](#)

[設定ファイル \[1066 ページ\]](#)

[.ini ファイルの内容を非表示にする \[566 ページ\]](#)

[設定ファイルでの条件付き解析 \[1068 ページ\]](#)

## 1.7.4.11 ヒストグラムユーティリティ (dbhist)

ヒストグラムを Microsoft Excel チャートに変換します。これには、述部の選択性に関する情報が含まれます。

### 構文

```
dbhist [ options ] -t table-name [ excel-output-filename ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-Coptions	接続パラメータを指定します。
-ncolname	ヒストグラムを関連付けるカラムの名前を指定します。カラムを指定しない場合は、テーブル内のヒストグラムを持っているすべてのカラムが返されます。
-ftable-name	チャートを生成するテーブルまたはマテリアライズドビューの名前を指定します。
-Uowner	テーブルまたはマテリアライズドビューの所有者を指定します。
excel-output-name	生成される Microsoft Excel ファイルの名前を指定します。名前を指定しない場合は、Microsoft Excel から名前の入力を求める名前を付けて保存ダイアログが表示されます。

### 権限

MANAGE STATISTICS システム権限が必要です。

### 備考

ヒストグラムは ISYSCOLSTAT システムテーブルに格納され、sa\_get\_histogram ストアドプロシージャを使用して取り出せます。ヒストグラムユーティリティは、ヒストグラムを Microsoft Excel チャートに変換します。これには、述部の選択性に関する情報が含まれます。ヒストグラムユーティリティ (dbhist) は Windows でのみ使用できます。このユーティリティを使用するためには、コンピュータに Microsoft Excel 97 以降がインストールされている必要があります。

たとえば、統計が最近削除された場合は、テーブルまたはマテリアライズドビューに対する統計 (ヒストグラムを含む) が存在しないことがあります。この場合、ヒストグラムユーティリティによって、"Histogram contains no data, aborting" というメッセージが表示されます。このような場合は、統計情報を作成してから、再度ヒストグラムユーティリティを実行する必要があります。テーブルまたはマテリアライズドビューの統計を作成するには、CREATE STATISTICS 文を実行します。

文字列カラムに対する述部の選択性を決定するには、ESTIMATE 関数または ESTIMATE\_SOURCE 関数を使用します。文字列カラムからヒストグラムを取り出そうとすると、sa\_get\_histogram とヒストグラムユーティリティがエラーを生成します。

シート名にはカラム名が使用されます。カラム名は、25 文字目以降がトランケートされ、¥、/、?、\*、[、]、: (Microsoft Excel では使用できない文字) はアンダースコア (\_) に置き換えられます。チャート名は chart で始まり、上記と同じ命名規則が適用されます。名前が重複した場合は (文字の置換、トランケート、カラム名が chart で始まることなどによって)、重複した名前は使用できないことを示す Microsoft Excel エラーが発生します。ただし、スプレッドシートは作成され、以前のバージョンで作成された名前 (Sheet1、Chart1 など) が付けられます。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

sa\_get\_histogram スタアドプロシージャを使用して、ヒストグラムを取得することもできます。

### 例

次のコマンド (全体を 1 つの行に入力) は、データベース *demo.db* にあるテーブル SalesOrderItems のカラム ProductID の Microsoft Excel チャートを生成し、それを histogram.xls として保存します。

```
dbhist -c "UID=DBA;PWD=sql;DBF=%SQLANYSAMPl7%¥demo.db" -n ProductID -t
SalesOrderItems histogram.xls
```

次の文は、テーブル SalesOrders 内にヒストグラムがあるすべてのカラムを対象とするチャートを生成します。サンプルデータベースがすでに起動されていることを前提とします。この文では、UID=DBA と PWD=sql を使用して接続も行います。出力ファイル名を指定していないので、Microsoft Excel から入力するように要求されます。

```
dbhist -t SalesOrders -c "UID=DBA;PWD=sql"
```

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[設定ファイル \[1066 ページ\]](#)

## 1.7.4.12 情報ユーティリティ (dbinfo)

指定したデータベースに関する情報を表示します。

### 構文

```
dbinfo [ options ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-c"keyword=value;..."	接続パラメータを指定します。
-ofilename	指定したファイルに、出力メッセージを追加します。
-q	クワイエットモードで実行します (メッセージを表示しません)。
-u	システムテーブルやユーザ定義のテーブルを含むすべてのテーブルとマテリアライズドビューの使用状況とサイズに関する情報を表示します。  ページ使用統計を要求できるのは、データベースに接続している他のユーザがいない場合のみです。ページの使用状況情報は sa_table_page_usage システムプロシージャを使用して取得します。この実行には MANAGE ANY STATISTICS 権限が必要です。

## 権限

どのユーザも dbinfo を実行できます。ただし、ページ使用統計を取得するには MANAGE ANY DBSPACE 権限が必要です。

## 備考

dbinfo ユーティリティを使用すると、データベースに関する情報が表示されます。データベースの名前、トランザクションログファイルまたはログミラーの名前、ページサイズ、照合名とラベル、テーブル暗号化が有効であるかどうかなどの情報がレポートされます。必要に応じて、テーブルの使用状況に関する統計とその詳細を含めることもできます。

dbinfo ユーティリティを使用して、ディスク上のテーブルのサイズを判断できます。そのためには、次のようなコマンドを実行します。

```
dbinfo -u -c "UID=DBA;PWD=sql;DBF=%SQLANY%SAMP17%¥demo.db"
```

結果は、データベース内の各テーブルに含まれるデータを保持するために使用されているページ数 (Pages) と、それらのページの使用率 (%used) を示します。すべてのテーブルについて、ページ数にデータベースページのサイズを掛け、その結果に %used を掛けて、そのテーブル用に使用されている領域を判断できます。

また、次の操作を行うことで、データベースに関する情報をさらに取得できます。

- DB\_PROPERTY システム関数を使用した個々のデータベースプロパティの問い合わせ。
- sa\_db\_properties システムプロシージャを使用したすべてのデータベースプロパティの問い合わせ。
- ping ユーティリティ (dbping) での -pd オプションの使用。-pd を指定すると、指定された各データベースプロパティの値が dbping から返されます。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

### 1.7.4.13 初期化ユーティリティ (dbinit)

新しいデータベースを作成します。

 **構文**

```
dbinit [ options ] new-database-file
```

オプション	説明
@data	<p>指定された環境変数または設定ファイルからオプションを読み込みます。</p> <p>設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。</p>
-a	<p>CHAR データ型または NCHAR データ型で Unicode 照合アルゴリズム (UCA) を使用している場合 (-z と -zn を参照)、文字列比較の際に文字のアクセント記号の違いを考慮します (たとえば、e は é より小さいと見なします)。UCA 照合を使用して作成された日本語のデータベースを除き、デフォルトではアクセント記号は無視されます (e は é と等しいと見なされます)。すべての基本文字 (アクセントと大文字/小文字の区別を取り除いた文字) が等しい場合は、アクセントが左から右へ比較されます。</p> <p>日本語のデータベースを作成する場合の UCA 照合では、デフォルトでアクセント記号が区別されます。つまり、アクセント記号が考慮されます。</p>

オプション	説明
-af	<p>CHAR データ型または NCHAR データ型で UCA を使用している場合 (-z と -zn を参照)、文字列比較の際に文字のアクセント記号の違いを考慮します (たとえば、e は é より小さいと見なします)。デフォルトでは、アクセント記号は無視されます (e は é と等しいと見なされます)。すべての基本文字 (アクセント記号を取り除いた文字) が等しい場合は、フランス語の規則に従ってアクセント記号が右から左へ比較されます。</p>
-b	<p>データベースにブランクを埋め込みます。</p> <p>データベースサーバは、すべての文字列を、可変長であり VARCHAR ドメインを使用して格納されている文字列として比較します。これには、固定長の CHAR カラムまたは NCHAR カラムの文字列比較も含まれます。また、データベースへの値の格納時に、データベースサーバは値の末尾への空白のパディングやこの削除は行いません。</p> <p>デフォルトでは、データベースサーバは空白を意味のある文字として扱います。したがって、値 'a' (文字 'a' と、後続の 1 つのブランク) は、単一文字の文字列 'a' と等しくありません。不等号比較の照合でも、ブランクは、ブランク以外の文字として扱われます。</p> <p>ブランク埋め込みが有効である場合 (dbinit -b オプション)、文字列比較のセマンティックは ANSI/ISO SQL 標準への追従性が高まります。ブランク埋め込みが有効であると、データベースサーバはすべての比較で末尾の空白を無視します。</p> <p>上に挙げた例では、ブランクを埋め込まれたデータベースで 'a' を 'a' に対して等号比較すると、TRUE が返されます。ブランクを埋め込まれたデータベースでは、固定長文字列の値は、アプリケーションによってフェッチされたときにブランクで埋め込まれます。このような文字の割り当てが行われたときにアプリケーションが文字列のトランケーション警告を受け取るかどうかは、ansi_blanks 接続オプションによって制御されます。</p>
-c	<p>比較や文字列操作をするときに、すべての値で大文字と小文字を区別します。大文字と小文字を区別するデータベースであっても、データベースの識別子については大文字と小文字は区別されません。</p> <p>UCA 照合を使用して作成された日本語のデータベースを除き、デフォルトではすべての比較において大文字と小文字が区別されません。日本語のデータベースを作成する場合の UCA 照合では、デフォルトで大文字と小文字が区別されます。</p> <p>このオプションは、ISO/ANSI SQL 標準との互換性を保つために用意されています。</p>

オプション	説明
<p><code>-dba</code>DBA-user,pwd</p>	<p>必須。作成後のデータベースへのログインに使用できるユーザ ID とパスワードを指定します (DBA ユーザとも呼ばれます)。このユーザはデータベースに対する完全な管理権限を持つことになるため、権限を必要とするすべての操作をデータベースで実行でき、セキュリティモデルを設定できます。</p> <p>デフォルトでは、<code>-mpl</code> オプションで別の値が指定されないかぎり、パスワードは 6 文字以上である必要があります。パスワードは 7 ビットの ASCII 文字で構成します。それ以外の文字を使用すると、データベースサーバがクライアントの文字セットを UTF-8 に変換できない場合、パスワードが正しく機能しないことがあります。</p> <p>次のコマンドは、パスワードが <code>passwd</code> である <code>dba</code> という DBA ユーザを使用して、<code>mydb</code> というデータベースを作成します。</p> <pre>dbinit -dba dba,passwd mydb.db</pre> <p>次のコマンドは、DBA ユーザをパスワードが <code>passwd</code> である <code>user1</code> に変更します。</p> <pre>dbinit -dba user1,passwd mydb.db</pre>
<p><code>-dbssize</code>[<i>k</i>   <i>m</i>   <i>g</i>   <i>p</i>]</p>	<p>データベース用の領域を事前に割り付けます。データベース用の領域を事前に割り付けると、データベースがあるドライブの空き領域が不足する危険性を小さくすることができます。また、データベースサイズを拡大する操作は時間を要するため、それが必要となる前にデータベースに保存できるデータの量を増やすことがパフォーマンスの向上につながります。</p> <p>デフォルトでは、<code>size</code> の値はバイト単位となります。単位をキロバイト、メガバイト、ギガバイト、またはページで指定するには、それぞれ <code>k</code>、<code>m</code>、<code>g</code>、<code>p</code> を使用します。</p>

オプション	説明
-eaalgorithm	<p>データベースエンコーディングまたはテーブルスエンコーディング (-et) で使用するアルゴリズムを指定します。単純な難読化の場合は、(-ek や -ep を指定するのではなく) -ea simple を指定します。単純な難読化は、データベースファイルが不用意に直接アクセスされた場合にデータが表示されないようにすることだけを目的としています。これにより第三者は、ディスクユーティリティを使用してファイルを表示し、データベースのデータを解読することが困難になります。</p> <p>セキュリティを強化するには、128 ビットの場合は AES、256 ビットの場合は AES256 の強力な暗号化を指定します。FIPS 認定の暗号化を使用するには、128 ビットの場合は AES_FIPS、256 ビットの場合は AES256_FIPS をそれぞれ指定してください。強力な暗号化を使用するためには、-ek または -ep オプションを使用してキーを指定する必要があります。</p> <p>暗号化されていないデータベースを作成するには、-ea none を指定するか、-ea オプションを指定しません (-et、-ep、-ek オプションも指定しません)。</p> <p>-ea オプションを指定しない場合、デフォルトの動作は次のようになります。</p> <ul style="list-style-type: none"> <li>• -ea NONE (-ek、-ep、-et が指定されない場合)</li> <li>• -ea AES (-ek または -ep が指定される場合) (-et の指定とは無関係)</li> <li>• -ea SIMPLE (-ek または -ep を指定せずに -et を指定する場合)</li> </ul> <p>アルゴリズム名の大文字と小文字は区別されません。</p> <p>次のコマンドは、強力的に暗号化されたデータベースを作成し、暗号化キーとアルゴリズムを指定します。</p> <pre>dbinit -dba DBA,passwd -ek "0kZ2o56AK#" -ea AES_FIPS "myencrypteddb.db"</pre> <p>ただし、ファイル圧縮ユーティリティを使用する場合、暗号化を実行したデータベースは、暗号化されていないデータベースほどには圧縮できません。</p>

オプション	説明
<p><code>-ekkey</code></p>	<p>コマンドに暗号化キーを直接指定することで、強力に暗号化されたデータベースを作成するように指定します。<code>-ek</code> オプションは、<code>-ea</code> オプションによって任意で指定された AES アルゴリズムとともに使用します。<code>-ea</code> オプションを指定しないで <code>-ek</code> オプションを指定すると、デフォルトで AES が使用されます。</p> <p>このオプションを <code>-et</code> とともに指定すると、データベースは暗号化されません。この場合、テーブル暗号化が有効になります。</p> <p><b>警告</b></p> <p>データベースに強力な暗号化を適用した場合、暗号化キーのコピーを必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、テクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。</p>
<p><code>-ep</code></p>	<p>ウィンドウに暗号化キーを入力することで、強力に暗号化されたデータベースを作成するように指定します。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。</p> <p>暗号化キーは、正確に入力されたことを確認するために 2 回入力してください。キーが一致しない場合は、初期化は失敗します。</p> <p>このオプションを <code>-et</code> とともに指定すると、データベースは暗号化されません。この場合、テーブル暗号化が有効になります。</p>
<p><code>-et</code></p>	<p><code>-ea</code> オプションで指定した暗号化アルゴリズム (およびキー) を使用したテーブル暗号化を有効にします。このオプションは、データベース全体を暗号化しないで、暗号化されたテーブルを作成する場合に使用します。<code>-et</code> を <code>-ek</code> または <code>-ep</code> とともに指定し、<code>-ea</code> を指定しないと、デフォルトで AES アルゴリズムが使用されます。<code>-et</code> だけを指定した場合は、単純難読化が使用されます。</p> <p>テーブル暗号化を有効にただで、テーブルが暗号化されるわけではありません。データベースを作成した後で、テーブルを個別に暗号化する必要があります。</p> <p>テーブルの暗号化が有効な場合、暗号化されたテーブルのテーブルページ、関連するインデックスページ、テンポラリファイルのページ、暗号化されたテーブルのトランザクションを含むトランザクションログページが暗号化されます。</p> <p>次の例は、キー <code>abc</code> と暗号化アルゴリズム <code>AES_FIPS</code> を使用する強力なテーブル暗号化を有効にしてデータベース <code>new.db</code> を作成します。</p> <pre>dbinit -dba DBA,passwd -et -ek abc -ea AES_FIPS new.db</pre>

オプション	説明
<code>-i</code>	<p>jConnect システムオブジェクトをデータベースから除外します。</p> <p>jConnect JDBC ドライバを使用してシステムカタログ情報にアクセスするには、jConnect カタログサポートをインストールします (デフォルトでインストールされます)。このオプションを指定した場合でも、システム情報にアクセスしないかぎり、JDBC を使用できます。</p> <p>SQL Central または ALTER DATABASE 文を使用して、jConnect のサポートを後から追加することもできます。</p>
<code>-k</code>	<p>SYSCOLUMNS ビューと SYSINDEXES ビューを作成しません。</p> <p>デフォルトでは、データベース作成機能は、Watcom SQL (このソフトウェアのバージョン 4 以前) で使用可能なシステムテーブルとの互換性を保つために、ビュー SYS.SYSCOLUMNS と SYS.SYSINDEXES を生成します。これらのビューは、SAP Adaptive Server Enterprise の互換ビュー dbo.syscolumns と dbo.sysindexes と競合します。</p>
<code>-kdi</code> number	<p>暗号化キーのハッシュ回数を指定します。1 から 1000 の間の整数を指定してください。デフォルト値は 2 で、暗号化キーが 2000 回ハッシュされます。回数が多いほど、セキュリティは高くなります。速度の遅いコンピュータ上でデータベースを稼働させている場合、回数が多すぎるとデータベースの起動が遅くなる可能性があります。この設定が起動時間に与える影響を評価するには、<code>-li</code> オプションを使用します (データベースがいったん動作すれば、パフォーマンスへの影響はありません)。<code>-kdi</code> オプションは、<code>-ek</code> または <code>-ep</code> オプションとともに指定する必要があります。</p>
<code>-l</code>	<p>使用可能な照合順を列挙した後、停止します。データベースは作成されません。照合順を指定するには、<code>-z</code> オプションを使用します。</p> <p>古いバージョンの SQL Anywhere との互換性の維持や、その他の特殊な用途で使用できる代替照合を含む、古い照合のリストを表示するには、<code>-l+</code> オプションを指定します。</p>
<code>-li</code> number	<p>暗号化キーがハッシュされる回数に基づいて、暗号化キーの作成にかかる時間をテストします。この時間は、総当たり攻撃でパスワードの候補を試すのにかかる時間と同じです。データベースの起動時にソフトウェアで暗号化キーの検証にかかる時間も指定します。これらの結果に基づいて、<code>-kdi</code> オプションの値を選択できます。1 から 1000 の間の整数を指定してください。<code>-li</code> オプションを使用するときには、他のオプションを指定しないでください。</p>

オプション	説明
<p><code>-le</code></p>	<p>使用可能な文字セットエンコードを列挙した後、停止します。データベースは作成されません。文字セットエンコードは、1つ以上のラベルによって識別されます。エンコードの識別に使用できる文字列があります。表示される各テキスト行には、エンコードラベルと、エンコードの識別に使用できる代替ラベルがリストされます。これらのラベルは、いくつかの共通カテゴリのいずれかに該当します。カテゴリには、SA (SQL Anywhere ラベル)、IANA (Internet Assigned Numbers Authority)、MIME (Multipurpose Internet Mail Extensions)、ICU (International Components for Unicode)、JAVA、ASE (Adaptive Server Enterprise) があります。</p> <p>代替ラベルが含まれる文字セットエンコードのリストを表示するには、<code>-le+</code> オプションを指定します。</p> <p>dbinit ユーティリティが文字セットエンコードをレポートするときは、必ず、ラベルの SQL Anywhere バージョンもレポートします。たとえば、次のコマンドを使用すると、CHAR データ型の文字セットエンコード 1250 のレポートが行われます。</p> <pre>dbinit -dba DBA,passwd -ze cp1250 -z uca test.db</pre>
<p><code>-mfilename</code></p>	<p>トランザクションログミラーを作成します。トランザクションログミラーはトランザクションログと同一のコピーで、通常は別のデバイスで管理され、データを確実に保護しています。デフォルトでは、トランザクションログミラーは使用されません。</p>
<p><code>-mplpositive-integer</code></p>	<p>最小パスワード長を設定します。オプションを指定しない場合、新しいデータベースに対する最小パスワード長は 6 です。</p>
<p><code>-n</code></p>	<p>トランザクションログのないデータベースを作成します。トランザクションログを使用しないデータベースを作成すると、ディスク領域を節約できますが、コミットごとにチェックポイントが発生するので、パフォーマンスが低下する可能性があります。また、トランザクションログを使用しないで実行していた場合、データベースが壊れたときにデータを回復できません。ただし、トランザクションログはデータレプリケーションに必要です。メディア障害またはシステム障害の発生時に、データベース情報のセキュリティを強化します。</p>
<p><code>-Ofilename</code></p>	<p>指定したファイルに、出力メッセージを追加します。</p>

オプション	説明
<p><code>-ppage-size [k]</code></p>	<p>データベースのページサイズを指定します。データベースのページサイズには、2048 (非推奨)、4096、8192、16384、32768 バイトのいずれかを指定できます。デフォルトは 4096 バイトです。<i>k</i> を使用して、キロバイトの単位を指定します (-p 4k など)。</p> <p>大規模なデータベースでは、より大きなページサイズの方が有利です。たとえば、テーブルのスキャンでは一度にページ全体が読み込まれるため、通常、必要な I/O 操作の回数は少なくなります。ただし、大きなページサイズには、より多くのメモリが必要です。ページサイズを選択するときは、パフォーマンステスト (およびテスト全般) を実行することを強くおすすめします。そして、満足できる結果を得られた最小のページサイズを選択します。ほとんどのアプリケーションでは、16 KB または 32 KB のページサイズはおすすめしません。常に十分なデータベースサーバキャッシュの確保が可能であり、メモリとディスク領域のパフォーマンス特性に対するトレードオフが調査済みである場合以外は、運用システムで 16 KB または 32 KB のページサイズは使用しないでください。多数のデータベースを同じサーバで起動する場合は、適切なページサイズを選択してください。</p>
<p><code>-pd</code></p>	<p>16.0 より前のシステムプロシージャのうち、権限を必要とする操作を実行するものを、呼び出し側の権限ではなく定義側 (所有者) の権限で実行することを指定します。<code>-pd</code> を指定した場合、ユーザはプロシージャに対する EXECUTE 権限以外の権限がなくても、これらのシステムプロシージャを実行できます。この設定は、バージョン 16.0 以降で追加されたシステムプロシージャのうち権限を必要とする操作を実行するものには影響を与えず、ユーザ定義のプロシージャ (これらは引き続き定義側となります) のデフォルトの動作にも影響を与えません。</p>
<p><code>-q</code></p>	<p>クワイエットモードで実行します (メッセージを表示しません)。</p>

オプション	説明
-s[ - ]	<p>グローバルチェックサムを追加します (チェックサムは各データベースページに追加されます)。デフォルトでは、このオプションはオンになります。このオプションをオフにするには、-s- を指定します。</p> <p>チェックサムは、データベースページがディスク上で変更されたかどうかを判断するために使用します。グローバルチェックサムを有効にしてデータベースを作成した場合、チェックサムはページがディスクに書き込まれる直前に計算されます。そのページが次にディスクから読み出されるときに、ページのチェックサムが再計算されて、ページに保存されているチェックサムと比較されます。チェックサムが異なる場合は、ディスク上のページが変更されているか、破損しており、エラーが発生します。-s オプションの値を指定したかどうかにかかわらず、重要なデータベースページにはデータベースサーバによって必ずチェックサムが追加されます。</p> <p>リムーバブルドライブなどのストレージデバイスで実行されるデータベースでは、データベースの破損を速やかに検出できるように、自動的にチェックサムが有効になります。</p> <p>グローバルチェックサムを無効にして (つまり、-s- を指定して) データベースが作成された場合でも、-wc オプションまたは START DATABASE 文を使用すると、ページの書き込み時にチェックサムをページに追加できます。</p>
-ttransaction-log-name	<p>トランザクションログファイルの名前を指定します。トランザクションログは、使用しているアプリケーションにかかわらず、すべてのユーザが行った変更をデータベースサーバがその中に記録するファイルです。トランザクションログはバックアップとリカバリ、データレプリケーションで重要な役割を果たします。トランザクションログのファイル名にパスがない場合、トランザクションログはデータベースファイルと同じディレクトリに保存されます。-t または -n を指定しないで dbinit を実行した場合は、データベースファイルと同じ名前で拡張子が .log のトランザクションログが作成されます。</p>

オプション	説明
<p><code>-zcoll [collation-tailoring-string]</code></p>	<p>データベースの照合順を指定します。照合順は、文字データ型 (CHAR、VARCHAR、LONG VARCHAR) のソートと比較に使用されます。照合は、使用されるエンコード (文字セット) に文字の比較と順序付けに関する情報をもたらすものです。照合は慎重に選択してください。データベースの作成が完了した後で照合を変更するためには、データベースをアンロードし、再ロードする必要があります。照合が指定されていない場合、データベースサーバは、オペレーティングシステムの言語と文字セットに基づいて照合を選択します。使用可能な照合順を表示する場合は、<code>-l</code> オプションを参照してください。</p> <p>必要に応じて、文字列のソートや比較を詳細に制御することために、照合の調整オプション (<code>collation-tailoring-string</code>) を指定することもできます。これらのオプションは、"キーワード=値" のペアの形式で、括弧で囲んで指定して、その後ろに照合名を記述します。例:</p> <pre>dbinit -dba DBA,passwd -c -z uca(locale=es;case=LowerFirst) spanish2.db</pre> <p>大文字小文字の設定とアクセント記号の区別の設定の両方を <code>collation-tailoring-string</code> で指定した場合は、<code>dbinit</code> に対する大文字小文字やアクセント記号の区別のオプション (<code>-c</code>、<code>-a</code>、および <code>-af</code>) が上書きされます。</p> <p><b>i 注記</b></p> <p>照合の適合理化オプションを使用して初期化したデータベースは、10.0.1 より前のデータベースサーバでは起動できません。</p>
<p><code>-zeencoding</code></p>	<p>照合のエンコードを指定します。<code>-z</code> で指定するほとんどの照合には、エンコード (文字セット) と順序付けの両方が定義されています。そのような照合については、<code>-ze</code> を指定する必要はありません。</p> <p><code>-z</code> で指定した照合が Unicode 照合アルゴリズム (UCA) である場合、<code>-ze</code> では CHAR データ型のエンコードとして UTF-8 または シングルバイトエンコードを指定できます。デフォルトでは、データベースサーバは UTF-8 を使用します。<code>-ze</code> を使用すると、ロケール固有のエンコードを指定し、比較と順序付けに UCA を利用することができます。</p>

オプション	説明
<code>-zn coll [collation-tailoring-string]</code>	<p>各国の文字データ型 (NCHAR、NVARCHAR、LONG NVARCHAR) のソートと比較に使用する照合順を指定します。照合は、使用される UTF-8 エンコード (文字セット) に文字の順序付けに関する情報をもたらすものです。このオプションの値は、UCA (デフォルト) と UTF8BIN です。UTF8BIN は、エンコードが 0x7E を超えるすべての文字のバイナリ順を規定します。dbicu17 と dbicudt17 の DLL がインストールされていない場合、デフォルトの NCHAR 照合は UTF8BIN になります。</p> <p>必要に応じて、文字列のソートや比較を詳細に制御するために、照合の調整オプション (<code>collation-tailoring-string</code>) を指定することもできます。これらのオプションは、"キーワード=値" のペアの形式で、括弧で囲んで指定して、その後ろに照合名を記述します。例:</p> <pre>dbinit -dba DBA,passwd -c -zn UCA(case=LowerFirst) sens.db</pre> <p>大文字小文字の設定とアクセント記号の区別の設定の両方を <code>collation-tailoring-string</code> で指定した場合は、dbinit に対する大文字小文字やアクセント記号の区別のオプション (-c、-a、および -af) が上書きされます。</p> <p><b>i 注記</b></p> <p>照合の適合化オプションを使用して初期化したデータベースは、10.0.1 より前のデータベースサーバでは起動できません。</p>

## 権限

なし。

## 備考

初期化するときいくつかのデータベース属性が指定されます。これらの属性は、データベース全体のアンロード、再初期化、再構築以外の方法では、後から変更することはできません。データベース属性には次のものがあります。

- 大文字と小文字の区別の有無
- アクセント記号の区別
- 句読表記の区別
- 比較における後続ブランクの処理

- ページサイズ
- 文字セットエンコードと照合順
- データベースの暗号化
- テーブル暗号化

たとえば、次のようにして 8192 バイトのページを持つデータベース `test.db` を作成できます。

```
dbinit -dba DBA,passwd -p 8192 test.db
```

クライアントに送信されるメッセージは、どのタイプのデータベース暗号化がデータベースに使用されるのかを示します。暗号化を使用している場合は、使用されているアルゴリズムも表示されます。

データベースに `utility_db` という名前を付けることはできません。この名前は、ユーティリティデータベースのために予約されています。

データベースに大文字小文字またはアクセント記号の区別がない場合は、初期化コマンドに照合の適合化オプションを指定するときに、句読表記の区別でレベル 4 は指定できません。

さらに、初期化するときに、トランザクションログとトランザクションログミラーを使うかどうかを選択することができます。この選択は、トランザクションログユーティリティまたは `ALTER DATABASE` 文を使用して後で変更できます。

データベースを暗号化するとき、またはデータベース内のテーブル暗号化を有効にするときには、暗号化キーを指定します。本ソフトウェアではパスワードベースの鍵導出関数 #2 (PBKDF2) を使用します。これは、総当たり攻撃からキーを保護する PKCS#5 標準の一部です。本ソフトウェアでは暗号化キーに対して暗号化ハッシュを繰り返し適用します。ハッシュの適用回数を指定するには、`-kdi` オプションを使用します。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## i 注記

アプリケーションを配備するときは、`dbinit` ユーティリティを使用してデータベースを作成するために、パーソナルデータベースサーバ (`dbeng17`) が必要です。パーソナルデータベースサーバは、その他のデータベースサーバが実行されていない場合にローカルコンピュータで SQL Central からデータベースを作成する場合にも必要です。

## 例

次のコマンドを実行すると、大文字と小文字を区別するデータベースである `spanish.db` が作成されます。このデータベースでは、非 NCHAR データに対して 1252spa 照合が使用されます。NCHAR データには、UCA 照合が指定されます。この場合、ロケールは `es` であり、最初に小文字がソートされます。

```
dbinit -dba DBA,passwd -c -z 1252spa -zn uca(locale=es;case=LowerFirst) spanish.db
```

次のコマンドは、ギリシャ語の照合を設定したデータベースを作成します。

```
dbinit -dba DBA,passwd -z 1253ELL mydb.db
```

## 関連情報

[Unicode 照合アルゴリズム \(UCA\) \[595 ページ\]](#)

- [照合の考慮事項 \[600 ページ\]](#)
- [トランザクションログ \[279 ページ\]](#)
- [テーブル暗号化 \[1621 ページ\]](#)
- [単純難読化と強力な暗号化 \[1615 ページ\]](#)
- [データベースの作成 \(SQL Central の場合\) \[266 ページ\]](#)
- [テーブルの暗号化 \[1625 ページ\]](#)
- [ansi\\_blanks オプション \[646 ページ\]](#)
- [推奨文字セットと照合 \[607 ページ\]](#)

## 1.7.4.14 Interactive SQL ユーティリティ (dbisql)

SQL 文を実行し、データベースに対してスクリプトファイルを実行します。

### 構文

```
dbisql -c "connection-string" [ options ] [ dbisql-statement | dbisql-script-file ]
```

```
dbisql -c "connection-string" -ul [ options ] [ dbisql-statement | dbisql-script-file ]
```

dbisql-statement: 1 つの SQL 文、またはコマンドデリミタによって区切られる一連の SQL 文。

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。同じ名前の環境変数と設定ファイルが存在する場合は、環境変数が使用されます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容を難読化します。Interactive SQL では、暗号化された設定ファイルをサポートしていません。
-C"keyword=value;..."	接続パラメータを指定します。Interactive SQL が接続できない場合は、接続パラメータを入力するウィンドウが表示されます。
-ddelimiter	コマンドデリミタを指定します。デリミタを囲む引用符は省略可能ですが、コマンドシェル自体がデリミタを特別な方法で解釈するときは必ず指定します。  このオプションは、Interactive SQL の command_delimiter オプションの設定を上書きします。

オプション	説明
<code>-dl</code>	<p>ユーザが明示的に実行するすべての文をコマンドウィンドウ (STDOUT) にエコーします。これによって、SQL スクリプトのデバッグ、または Interactive SQL が長文の SQL スクリプトを処理しているときに有用なフィードバックが提供されます。(最後の文字は数値の 1 であり、L の小文字ではありません)。このオプションは、Interactive SQL をコマンドラインプログラムとして実行している場合のみ使用できます。</p>
<code>-datasource</code> <code>DSN-name</code>	<p>接続先の ODBC データソースを指定します。</p>
<code>-filename</code>	<p>[SQL 文] ウィンドウ枠で <code>filename</code> というファイルを (実行しないで) 開きます。</p> <p>ファイル名は引用符で囲んでも囲まなくてもかまいませんが、ファイル名にスペースが含まれている場合は引用符で囲む必要があります。</p> <p>そのファイルが存在しない場合、またはファイルではなく実際にはディレクトリである場合は、Interactive SQL がエラーメッセージを出力して終了します。</p> <p>ファイル名に完全なドライブとパスの仕様が含まれていないときは、現在のディレクトリが基準として想定されます。</p> <p>このオプションは、Interactive SQL をウィンドウベースのアプリケーションとして実行している場合のみサポートされます。</p>
<code>-hana</code>	<p>デフォルトで接続する SAP HANA データベースを指定します。Interactive SQL では、接続するデータベースのタイプに応じて使用可能なオプションがカスタマイズされます。</p> <p>デフォルトに設定されているデータベースのタイプに関係なく、<a href="#">接続ウィンドウのデータベースタイプを変更</a>ドロップダウンリストからデータベースのタイプを選択することで、SAP HANA、SQL Anywhere、SAP IQ、または Ultra Light のデータベースに接続できます。</p>
<code>-host</code> <code>hostname</code>	<p>データベースサーバを実行するコンピュータの <code>hostname</code> または IP アドレスを指定します。現在のコンピュータを表す <code>localhost</code> という名前を使用できます。</p>
<code>-iq</code>	<p>デフォルトで接続する SAP IQ データベースを指定します。Interactive SQL では、接続するデータベースのタイプに応じて使用可能なオプションがカスタマイズされます。</p> <p>デフォルトに設定されているデータベースのタイプに関係なく、<a href="#">接続ウィンドウのデータベースタイプを変更</a>ドロップダウンリストからデータベースのタイプを選択することで、SAP HANA、SQL Anywhere、SAP IQ、または Ultra Light のデータベースに接続できます。</p>

オプション	説明
-nogui	<p>Interactive SQL をコンソールアプリケーションとして実行します。このとき、ウィンドウベースのユーザインタフェースは使用しません。これは、バッチ処理に便利です。</p> <p>または <code>dbisql-statement</code>、<code>dbisql-script-file</code> のいずれかを指定すると、-nogui と見なされます。</p> <p>このモードのとき、Interactive SQL は、処理の成功または失敗をプログラム終了コードを設定することで示します。Windows オペレーティングシステムでは、プログラム終了コードに対して環境変数 ERRORLEVEL が設定されます。</p>
-nohistory	<p>Interactive SQL の終了時に SQL 履歴が保存されないようにします。プログラムが実行されている間は、実行された文は履歴タブに表示されます。</p>
-onerror { continue   exit }	<p>スクリプトファイルから文を読み込み中にエラーが発生した場合の処置を制御します。これは、Interactive SQL をバッチ処理で使用するときに便利です。このオプションは、Interactive SQL の on_error オプションの設定を上書きします。</p> <p>behavior には次のいずれかの値を定義できます。</p> <p><b>Continue</b></p> <p>エラーは無視され、Interactive SQL は文の実行を継続します。</p> <p><b>Exit</b></p> <p>Interactive SQL は終了します。</p>
-portport-number	<p>SQL Anywhere データベースサーバが実行されているポート番号を指定します。デフォルトのポート番号は 2638 です。<sup>1</sup></p>
-q	<p>出力メッセージを表示しません。ユーティリティをクワイエットモードで実行するように設定します。これは、文またはスクリプトファイルを使って Interactive SQL を起動したときにのみ便利です。このオプションを指定した場合、エラーメッセージは表示されますが、次の情報は表示されません。</p> <ul style="list-style-type: none"> <li>警告およびその他の致命的でないメッセージ-we オプションが指定された場合、致命的でないメッセージのみが非表示になります。</li> <li>結果セットの出力</li> </ul>

オプション	説明
-sa	<p>デフォルトで接続する SQL Anywhere データベースを指定します。Interactive SQL では、接続するデータベースのタイプに応じて使用可能なオプションがカスタマイズされます。</p> <p>デフォルトに設定されているデータベースのタイプに関係なく、<a href="#">接続ウィンドウのデータベースタイプを変更</a>ドロップダウンリストからデータベースのタイプを選択することで、SAP HANA、SQL Anywhere、SAP IQ、または Ultra Light のデータベースに接続できます。</p>
-ul	<p>デフォルトで接続する Ultra Light データベースを指定します。Interactive SQL では、接続するデータベースのタイプに応じて使用可能なオプションがカスタマイズされます。</p> <p>デフォルトに設定されているデータベースのタイプに関係なく、<a href="#">接続ウィンドウのデータベースタイプを変更</a>ドロップダウンリストからデータベースのタイプを選択することで、SAP HANA、SQL Anywhere、SAP IQ、または Ultra Light のデータベースに接続できます。</p>
-version	Interactive SQL のバージョン番号を表示します。Interactive SQL 内のヘルプメニューから <a href="#">Interactive SQL について</a> をクリックして、バージョン番号を参照することもできます。
-we	エラーの場合と同様に、SQL 文警告を表示して処理します。そのため、-q オプションが指定された場合も、警告は非表示になりません。
-x	文をスキャンしますが、実行しません。長いスクリプトファイルの構文エラーをチェックする場合に有用です。
<code>dbisql-statement</code>   <code>dbisql-script-file</code>	<p>SQL 文または指定された <code>dbisql-script-file</code> を実行します。</p> <p><code>dbisql-statement</code> または <code>dbisql-script-file</code> が指定されていないと、Interactive SQL は対話型モードになります。このモードでは、文をコマンドウィンドウに入力できます。</p>

<sup>1</sup> このオプションは、Ultra Light では無視されます。

## 備考

Interactive SQL を使用して、データベースのブラウズ、SQL 文の実行、およびスクリプトファイルの実行を行うことができます。また、次の項目に関するフィードバックも提供します。

- 影響を受けたローの数

- 各文に必要な時間
- クエリの実行プラン
- エラーメッセージ

Interactive SQL を使用して、SQL Anywhere データベース、Ultra Light データベース、SAP Sybase IQ データベース、SAP HANA データベース、または汎用の ODBC データベースに接続できます。

Windows の場合、実行プログラムが 2 つあります。

1. バッチスクリプトでは、dbisql.exe ではなく、dbisql または dbisql.com を呼び出す必要があります。実行プログラム dbisql.com は、コンソールアプリケーションとしてリンクされています。
2. 実行プログラム dbisql.exe はウィンドウアプリケーションとしてリンクされており、起動元のコマンドシェルがブロックされません。dbisql.exe をバッチファイルから実行した場合、標準出力または標準のエラーファイルに出力が送信されません。

Interactive SQL のデフォルトのエンコードは、Interactive SQL の default\_isql\_encoding オプションでも一時的に設定できます。

INPUT、OUTPUT、または READ 文の ENCODING 句を使用して、ファイルの読み込みまたは書き込みをするときに使用するエンコードを指定できます。

- INPUT 文
- OUTPUT 文
- READ 文

終了コードは、0 (成功) または 0 以外の値 (失敗) です。0 以外の終了コードが設定されるのは、(SQL 文またはスクリプトファイル名を指定したコマンドラインによって) Interactive SQL をバッチモードで実行した場合だけです。

Ultra Light では、コマンドプロンプトモードのとき、Interactive SQL は、処理の成功または失敗を、プログラム終了コードを設定することで示します。Windows オペレーティングシステムでは、プログラム終了コードに対して環境変数 ERRORLEVEL が設定されます。

Interactive SQL で reload.sql ファイルを実行しており、データベースが暗号化されている場合は、パラメータとして暗号化キーを指定する必要があります。READ 文にキーを指定していない場合、Interactive SQL でキーの入力を要求するプロンプトが表示されます。

次の方法で Interactive SQL を開始できます。

- SQL Central で **ファイル > Interactive SQL を開く** をクリックします。
- スタートメニューから、**スタート > プログラム > SQL Anywhere17 > 管理ツール > Interactive SQL** をクリックします。
- コマンドプロンプトで dbisql コマンドを使用します。

#### 例

次のコマンドは、ユーザ ID DBA とパスワード passwd で、現在のデフォルトサーバに対してスクリプトファイル mycom.sql を実行します。スクリプトファイルにエラーがあった場合は、処理は停止します。

```
dbisql -c "UID=DBA;PWD=passwd" -onerror exit mycom.sql
```

次のコマンドは、Ultra Light の CustDB.udb データベースに対してスクリプトファイル mycom.sql を実行します。-onerror オプションは終了として定義されています。このため、スクリプトファイルにエラーがあった場合、処理は終了します。

```
dbisql -ul -c "UID=DBA;PWD=passwd;DBF=CustDB.udb" -onerror exit mycom.sql
```

次のコマンドを入力すると、現在のデフォルトデータベースにユーザが追加されます。

```
dbisql -c "UID=DBA;PWD=passwd" CREATE USER joe IDENTIFIED BY joespasswd
```

次のコマンドを入力すると、SAP HANA データベースが開かれます。

```
dbisql -hana -c "user=test;password=notSoSecret;host=akbar:30015"
```

## 関連情報

[Interactive SQL の SQL 文 \[996 ページ\]](#)

[Interactive SQL \[980 ページ\]](#)

[Interactive SQL の SQL 文 \[996 ページ\]](#)

[設定ファイル \[1066 ページ\]](#)

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[データベースの作成 \[265 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

[command\\_delimiter オプション \[Interactive SQL\] \[1007 ページ\]](#)

[default\\_isql\\_encoding オプション \[Interactive SQL\] \[1009 ページ\]](#)

[on\\_error オプション \[Interactive SQL\] \[1021 ページ\]](#)

### 1.7.4.15 キーペアジェネレータユーティリティ (createkey)

Mobile Link エンドツーエンド暗号化で使用する RSA のキーペアを作成します。

#### 構文

```
createkey
```

#### 備考

createkey を実行する場合は、次の情報を求めるプロンプトが表示されます。

**RSA のキー長を入力してください (512 ~ 16384)**

このプロンプトは、RSA 暗号化を選択した場合にかぎり表示されます。512 ~ 16384 ビットの間で長さを選択できます。  
パブリックキーを格納するファイルパスを入力してください

生成された、PEM または DER でエンコードされたパブリックキーのファイル名とロケーションを指定します。このファイルは、Mobile Link クライアントで e2ee\_public\_key プロトコルオプションによって指定されます。

プライベートキーを保存するファイル名を入力してください

生成された、PEM または DER でエンコードされたプライベートキーのファイル名とパスを指定します。このファイルは、Mobile Link サーバで e2ee\_private\_key プロトコルオプションによって指定されます。

プライベートキーを保護するためのパスワードを入力してください

任意で、プライベートキーを暗号化するために使用するパスワードを指定します。パスワードを指定しない場合、プライベートキーは暗号化されません。このパスワードは、Mobile Link サーバで e2ee\_private\_key\_password プロトコルオプションによって指定されます。

## 権限

なし。

### 例

次の例では、RSA キーペアを作成します。

```
createkey
SQL Anywhere Key Pair Generator Version 17.0.4.1691
RSA のキー長を入力してください (512 ~ 16384): 2048
キーペア作成中...
パブリックキーを格納するファイルパスを入力してください: rsapublic.pem
プライベートキーを保存するファイル名を入力してください: rsaprivate.pem
プライベートキーを保護するためのパスワードを入力してください: pwd
```

## 1.7.4.16 言語選択ユーティリティ (dblang)

データベースサーバ、SQL Central、Interactive SQL、およびその他のツールで使用される言語を制御するレジストリ設定をレポートし、変更します。

### 構文

```
dblang [ options ] language-code
```

オプション	説明
-m	HKEY_LOCAL_MACHINE の下のレジストリに、言語コードを書き込みます。
-q	クワイエットモードで実行します (メッセージを表示しません)。

オプション	説明
-u	HKEY_CURRENT_USER の下のレジストリに、言語コードを書き込みます。これがデフォルトのロケーションです。

言語コード	言語
EN	英語
DE	ドイツ語
ES	スペイン語
FR	フランス語
IT	イタリア語
JA	日本語
KO	韓国語
LT	リトアニア語
PL	ポーランド語
PT	ポルトガル語
RU	ロシア語
TW	中国語 (繁体文字)
UK	ウクライナ語
ZH	中国語 (簡体字)

## 権限

なし。

## 備考

-m または -u を指定しないと、言語コードは HKEY\_CURRENT\_USER の下のレジストリに書き込まれます。-m と -u の両方を指定すると、両方のロケーションに言語コードを書き込むことができます。

言語コードを指定しないで dblank ユーティリティを実行すると、現在の設定がレポートされます。

### i 注記

言語設定の変更は、ソフトウェアコンポーネントを再起動するまでは検出されません。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

このユーティリティは、設定ファイルからオプションを読み込む `@data` パラメータを受け入れません。

## 例

次のコマンドは、現在の設定を示すウィンドウを表示します。

```
dblang
```

次のコマンドは、設定をドイツ語に変更し、前の設定と新しい設定を示すウィンドウを表示します。

```
dblang de
```

## 関連情報

[SALANG 環境変数 \[551 ページ\]](#)

[Language \(LANG\) 接続パラメータ \[89 ページ\]](#)

## 1.7.4.17 ログ変換ユーティリティ (dbtran)

トランザクションログを SQL スクリプトファイルに変換します。

### 構文

データベースサーバに対して実行する場合。

```
dbtran [ options ] -C { connection-string } -n SQL-script-file
```

トランザクションログに対して実行する場合。

```
dbtran [ options ] [ transaction-log ] [ SQL-script-file ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。

オプション	説明
-a	<p>コミットされていないトランザクションを SQL スクリプトファイルに含めるかどうかを制御します。</p> <p>デフォルトでは、SQL スクリプトファイルにはコミット済みの変更のみ含まれています。ロールバックされた変更や、直前の COMMIT の後に発生した変更は、SQL スクリプトファイルには存在しません。</p> <p>-a を指定した場合、トランザクションログにあるコミットされていないトランザクションに続いて、ROLLBACK 文が SQL スクリプトに書き込まれます。</p>
-c "keyword=value; ..."	<p>このユーティリティをデータベースサーバに対して実行する場合に、接続文字列を指定します。</p>
-d	<p>トランザクションを古いものから新しいものへと順に書き出すことを指定します。これにより出力が読みやすくなります。この機能は、主にデータベースのアクティビティを監視するときに使用されます。</p> <p><b>警告</b></p> <p>年代順で実行中の文では、結果となるデータベースの同一性が保証されないため、年代順の出力はデータベースには適用しないでください。</p>
-ek key	<p>強かに暗号化されているデータベースの暗号化キーを指定します。データベースが強かに暗号化されている場合、データベースまたはトランザクションログを使用するには暗号化キーを指定する必要があります。</p> <p>強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。正しい暗号化キーを指定しないとコマンドが失敗します。</p> <p>データベースサーバに対して -c オプションを使用して dbtran を実行している場合は、-ek オプションではなく、接続パラメータを使用してキーを指定してください。たとえば、次のコマンドでは、データベースサーバ <b>sample</b> からデータベース myenc.db に関するトランザクションログが取得され、その出力が log.sql に保存されます。</p> <pre>dbtran -n log.sql -c "Server=sample; DBF=myenc.db; UID=DBA; PWD=passwd; DBKEY=mykey"</pre>

オプション	説明
-ep	<p>暗号化キーを入力するよう要求します。このオプションを指定すると、暗号化キーを入力するためのウィンドウが表示されます。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。</p> <p>強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。正しい暗号化キーを指定しないとコマンドが失敗します。</p> <p>データベースサーバに対して -c オプションを使用して dbtran を実行している場合は、-ep オプションではなく、接続パラメータを使用してキーを指定してください。たとえば、次のコマンドでは、データベースサーバ <b>sample</b> からデータベース myenc.db に関するトランザクションログが取得され、その出力が log.sql に保存されます。</p> <pre data-bbox="874 869 1460 963">dbtran -n log.sql -c "Server=sample;DBF=myenc.db; UID=DBA;PWD=passwd;DBKEY=mykey"</pre>
-f	<p>最終チェックポイント以降に完了したトランザクションだけを出力します。</p>
-g	<p>auditing データベースオプションがオンの場合は、監査情報を SQL スクリプトファイルに (コメントとして) 追加します。</p> <p>-g オプションは、-a、-d、-t オプションも暗黙で指定したことになります。</p>
-iroffset1,offset2	<p>2 つの指定オフセット間の部分的トランザクションログを出力します。</p>
-i\$source,...	<p>次の 1 つ以上のソース (カンマ区切りのリストで指定) の操作によって変更された、ローに対する操作を出力します。</p> <p>すべて</p> <p>すべてのロー。これはデフォルト設定です。</p> <p><b>SQLRemote</b></p> <p>SQL Remote を使用して変更されたローだけを含みます。SR という短い形式を使用することもできます。</p> <p><b>Local</b></p> <p>レプリケートされないローだけを含みます。</p>
-itowner.table,...	<p>カンマ区切りのリストで指定されたテーブルに対する操作を出力します。各テーブルは、owner.table として指定します。</p>

オプション	説明
<code>-jdate/time</code>	指定した日付または時刻より前の、最新のチェックポイント以降のトランザクションだけを変換します。ユーザは、日付、時刻、日付と時刻を引用符で囲んで引数を指定できます。時刻を省略すると、その日付の開始時刻が使用されます。日付を省略すると、今日の日付が使用されます。日付と時刻の有効なフォーマットは、次のとおりです。"YYYY/MMM/DDHH:NN".
<code>-k</code>	エラーが検出されたときに部分的な .sql ファイルが消去されないようにします。dbtran の実行中にエラーが検出された場合、その時点まで生成された .sql ファイルは、部分的なファイルが間違っ使用されないように通常は消去されます。このオプションを指定すると、破損したトランザクションログからトランザクションをサルベージしようとするときに役立ちます。
<code>-mdirectory</code>	トランザクションログを格納するディレクトリを指定します。-m オプションは、トランザクションが複数のトランザクションログファイルにまたがる可能性がある場合に使用します。SQL 出力ファイルを複数のトランザクションログから生成する場合、ログ変換ユーティリティは、ログからシーケンスへの操作をインターリーブします。このシーケンスは、データベースのリストア時に操作が適用される順序を反映します。このオプションを使用するには -n オプションが必要です。
<code>-nfilename</code>	SQL 出力ファイルを指定します。
<code>-ofilename</code>	指定したファイルに、出力メッセージを追加します。
<code>-q</code>	クワイエットモードで実行します (メッセージを表示しません)。
<code>-r</code>	コミットされなかったトランザクションを削除します。これはデフォルトの動作です。
<code>-s</code>	UPDATE 文の生成方法を制御します。テーブルにプライマリキーまたはユニークなインデックスがない場合、重複ローが存在する可能性があるため、このオプションを指定しないと、ログ変換ユーティリティは標準ではない FIRST キーワードを使って UPDATE 文を作成します。このオプションを使用すると、FIRST キーワードが省略され、SQL 標準との互換性が保持されます。
<code>-sr</code>	SQL Remote がリモートサイトに操作を分配する方法を記述するコメントを生成し、出力ファイルに挿入します。
<code>-t</code>	トリガをスクリプトファイルに含めるかどうかを制御します。デフォルトでは、トリガが実行する動作はスクリプトファイルには含まれません。一致するトリガがデータベースにある場合は、データベースに対してスクリプトファイルを実行すると、トリガの動作が自動的に実行されます。すファイルが実行されるデータベース内に一致するトリガがない場合は、トリガの動作を出力に入れてください。
<code>-userid,...</code>	指定したユーザのトランザクションログだけが出力されるように制限します。
<code>-xuserid,...</code>	指定したユーザ以外のトランザクションログが出力されるように制限します。
<code>-y</code>	確認メッセージを表示することなく、既存のスクリプトファイルを置き換えます。-q を指定する場合、-y も指定しないと操作は失敗します。

オプション	説明
-z	トリガによって生成されたトランザクションを、コメントとして出力ファイルの中を含めます。
transaction-log	変換するログファイルを指定します。-c または -m オプションとは一緒に使用できません。
SQL-file	変換した情報を含む出力ファイルを指定します。transaction-log のみで使用します。

## 権限

データベースサーバに対して dbtran を実行する場合は、BACKUP DATABASE システム権限が必要です。トランザクションログファイルに対して dbtran を実行する場合は、権限は必要ありません。

## 備考

dbtran ユーティリティは、トランザクションログ内の情報を取り出して、それらを一連の SQL 文とコメントとして出力ファイルに入れます。このユーティリティは、次の方法で実行できます。

データベースサーバに対して実行

データベースサーバに対して実行した場合、dbtran は標準的なクライアントアプリケーションとして動作します。これは、-c オプションの後に指定された接続文字列を使用してデータベースサーバに接続し、-n オプションによって指定されたファイルに出力を送ります。

次のコマンドは、サーバ demo17 からのログ情報を変換して、出力をファイル demo.sql に入れます。

```
dbtran -c "Server=demo17;DBN=demo;UID=DBA;PWD=sql" -n demo.sql
```

トランザクションログファイルに対して実行

トランザクションログに対して実行した場合、dbtran はトランザクションログファイルに対して直接作用します。ユーザがこのコマンドを実行できないように、トランザクションログファイルを通常のアクセスから保護してください。

```
dbtran demo.log demo.sql
```

dbtran ユーティリティが実行されると、トランザクションログの初期のログオフセットが表示されます。複数のログファイルが作成される場合、順序を決定するにはこの方法が効果的です。

-c を使用する場合、dbtran はオンライントランザクションログファイルと、オンライントランザクションログファイルと同じディレクトリにあるすべてのオフライントランザクションログファイルを変換しようとします。ディレクトリに複数のデータベース用のトランザクションログファイルが含まれている場合、dbtran がエラーを示す場合があります。この問題を回避するには、各ディレクトリに 1 つのデータベースのみのトランザクションログファイルが含まれていることを確認します。

トランザクションは複数のトランザクションログにまたがる場合があります。トランザクションログファイルに複数のログにまたがるトランザクションが含まれている場合、単一のトランザクションログファイル (たとえば dbtran demo.log) を変換すると、関連しているトランザクションが失われる場合があります。トランザクションが複数のログファイルにまたがる可能性がある場

合に、完全なトランザクションを生成するには、-m オプションを使用します。-m オプションは、指定したディレクトリにあるすべてのログからすべてのトランザクションを含む単一の SQL ファイル (-n によって指定) を生成するよう dbtran に指示します。

次の方法で、ログ変換ユーティリティにアクセスできます。

- SQL Central の [ログファイル変換ウィザード](#) を使用します。
- コマンドプロンプトで、dbtran コマンドを入力します。SQL スクリプトやコマンドファイルに組み込む場合に便利です。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[トランザクションログが複数の場合のデータベースのリカバリ \[934 ページ\]](#)

[auditing オプション \[656 ページ\]](#)

## 1.7.4.18 パフォーマンス統計値ユーティリティ (dbstats) (UNIX)

UNIX でデータベースサーバのパフォーマンス統計値を返します。

### 構文

```
dbstats [ options ] [ interval ] [ iterations ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-dstatistic-definition-string	統計名と、統計をモニタするスコープ (データベースサーバ、データベース、または接続) を指定します。
-eserver-definition-string	モニタするデータベースサーバ名のカンマ区切りのリストを指定します。このオプションを指定しないと、デフォルトのデータベースサーバに関する統計のみが dbstats ユーティリティから返されます。
-l	モニタに使用できる統計がリストされます。
-ofilename	指定したファイルに、出力メッセージを書き込みます。
-vverbosity	指定のレベルでメッセージをレポートします。サポートされる値は、 <i>error</i> 、 <i>warning</i> 、 <i>info</i> です。デフォルトのレベルは <i>info</i> です。

オプション	説明
<code>interval</code>	統計値をレポートする頻度を秒単位で指定します。 <code>interval</code> を指定すると、統計のレポートを停止するまで、統計値が指定の間隔で返されます。
<code>iterations</code>	統計値の合計レポート回数を指定します。 <code>interval</code> に値を指定するには、 <code>iterations</code> 値を指定する必要があります。

## 権限

なし。

## 備考

dbstats ユーティリティでは、共有メモリ接続を使用して、同じコンピュータで稼働しているサーバがモニタされます。このユーティリティは、リモートデータベースサーバのモニタには使用できません。

`statistic-definition-string` の形式は次のようになります。

```
[ [ statistic-name ] [ , [ statistic-name ] , [ ... ] ] ] [ : [ database-servers ] / [ databases ] / [ connections ] ]
```

`database-servers`、`databases`、`connections` の値は、モニタするデータベースサーバ名、データベース名、接続名のカンマ区切りのリストです。名前のカンマ区切りリストを指定するほかに、アスタリスク (\*) を使用し、すべてのデータベースサーバ、データベース、接続のモニタを指定することもできます。サポートされているスコープのいずれかの値を指定しないと、そのスコープはモニタされません。

dbstats ユーティリティでオプションをまったく指定しないと、有効なすべてのスコープについて、統計値 `ActiveReq`、`ConnCount`、`CurrentCacheSize`、`DiskRead`、`DiskWrite`、`MainHeapPages`、`UnschReq` が自動的にモニタされます。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

### 例

デフォルトのデータベースサーバで、すべてのデータベースと接続について `ActiveReq` 統計情報の統計値のみを問い合わせしてから、終了します。

```
dbstats -d ActiveReq
```

デフォルトのデータベースサーバで、すべてのデータベースと接続について `ActiveReq` `statistic` の統計値のみを 1 秒に 1 回ずつ無制限に問い合わせます。

```
dbstats -d ActiveReq 1
```

デフォルトのデータベースサーバで、すべてのデータベースと接続について ActiveReq statistic の統計値のみを 1 秒に 1 回ずつ 5 回だけ問い合わせから、終了します。

```
dbstats -d ActiveReq 1 5
```

すべてのデータベースと接続について ActiveReq と ConnCount の統計値を問い合わせしてから、終了します。

```
dbstats -d "[ ActiveReq, ConnCount ]"
```

データベースサーバの範囲で ActiveReq と ConnCount の統計値を問い合わせしてから、終了します。

```
dbstats -d "[ActiveReq, ConnCount]:*"
```

デフォルトのデータベースサーバで稼働中のデータベース db1 と db2 のみを対象として、データベースの範囲で ActiveReq と ConnCount の統計値を問い合わせしてから、終了します。

```
dbstats -d "[ActiveReq, ConnCount]:/db1,db2/"
```

## 関連情報

[接続、データベース、データベースサーバのプロパティ \[838 ページ\]](#)

### 1.7.4.19 Ping ユーティリティ (dbping)

データベースサーバとデータベースへの接続をテストします。

#### 構文

```
dbping [ options ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-C"keyword=value; ..."	サーバと任意でデータベースへの接続に使用された接続パラメータを指定します。デフォルトでは、dbping は実行中のサーバに接続します。データベースに接続するためには -d オプションを指定します。

オプション	説明
-d	<p>データベースへの接続をテストします。必要があれば、dbping はサーバとデータベースを自動的に起動できます。-d があると、dbping は、接続を確立する通常のクライアントアプリケーションのように動作します。</p> <p>-d オプションを指定した場合、dbping はサーバとデータベースに接続する処理の成功をレポートします。-d オプションを指定しない場合、dbping はサーバを検出すると、処理の成功をレポートします。</p> <p>たとえば、サンプルデータベースを実行する blair という名前のデータベースサーバがある場合、dbping がサーバ blair に接続できるので、次のコマンドは成功します。</p> <pre data-bbox="863 797 1465 842">dbping -c "Server=blair;DBN=other"</pre> <p>次のコマンドは失敗し、other データベースが実行中ではなかったため、"Ping database failed -- specified database not found" というメッセージが表示されます。</p> <pre data-bbox="863 1003 1465 1048">dbping -c "Server=blair;DBN=other" -d</pre>
-en	<p>指定したプロパティによって NULL が返された場合に、dbping が失敗を示すリターンコードを返して終了することを指定します。デフォルトでは、-pc、-pd、-ps のいずれかで指定されたプロパティの値が不明である場合、dbping は NULL を出力し、成功を示すリターンコードを返して終了します。このオプションは、-pc、-pd、-ps を指定した場合のみ使用できます。</p>
-llibrary	<p>使用する ODBC ライブラリを指定します (ファイル拡張子は付けません)。このオプションを使用すると、ODBC ドライバマネージャの使用が回避されるので、UNIX オペレーティングシステムで便利です。</p> <p>たとえば、次のコマンドは、ODBC ドライバを直接ロードします。</p> <pre data-bbox="863 1514 1465 1581">dbping -m -c "DSN=SQL Anywhere 17 Demo;PWD=sql" -l dbodbc17</pre> <p>UNIX でスレッド接続ライブラリを使用する場合は、スレッド版の Ping ユーティリティである dbping_r を使用する必要があります。</p>
-m	<p>ODBC を使用して接続を確立します。デフォルトでは、このユーティリティは、Embedded SQL インタフェースを使用して接続します。</p>
-Ofilename	<p>指定したファイルに、出力メッセージを追加します。</p>

オプション	説明
<p><code>-pcproperty,...</code></p>	<p>指定されている接続プロパティを表示します。プロパティは、カンマで区切って指定します。このオプションを使用するには、データベース接続を確立するために必要な接続情報を指定してください。</p> <p>たとえば、次のコマンドは、接続プロパティとして使用できる <code>fire_triggers</code> オプションの設定を表示します。</p> <pre data-bbox="762 562 1214 600">dbping -c ... -pc fire_triggers</pre>
<p><code>-pdproperty[@db-name],...</code></p>	<p>指定されているデータベースプロパティを表示します。プロパティは、カンマで区切って指定します。</p> <p>たとえば、次のコマンドは、データベースで使用されているページサイズを表示します。</p> <pre data-bbox="762 824 1142 862">dbping -c ... -pd PageSize</pre> <p>必要に応じて、値を取得するデータベースの名前を指定できます。プロパティに <code>db-name</code> を追加してデータベース名が指定されていない場合は、前のプロパティに指定されたデータベース名が使用されます。</p> <p>次のコマンドは、データベース <code>mydb</code> で使用されているページサイズと照合を表示します。</p> <pre data-bbox="762 1144 1098 1205">dbping -c ... -pd   PageSize@mydb,Collation</pre>
<p><code>-psproperty,...</code></p>	<p>指定されているデータベースサーバプロパティを表示します。プロパティは、カンマで区切って指定します。このオプションを使用するには、データベース接続を確立するために必要な接続情報を指定してください。</p> <p>たとえば、次のコマンドは、データベースサーバに対するライセンスシート数またはプロセッサ数を表示します。</p> <pre data-bbox="762 1496 1198 1534">dbping -c ... -ps LicenseCount</pre>
<p><code>-q</code></p>	<p>クワイエットモードで実行します (メッセージを表示しません)。</p>

オプション	説明
-s	dbping を実行しているコンピュータとデータベースサーバを実行しているコンピュータ間のネットワークのパフォーマンス情報を返します。接続速度、遅延時間、スループットの概算値が表示されます。通常は、-c オプションを使用して、サーバ上のデータベースに接続するための接続パラメータを指定します。dbping -s は Embedded SQL 接続でのみ使用できます。-m または -l を指定した場合、-s オプションは無視されます。デフォルトでは、dbping -s は測定する統計ごとに最低 1 秒間、要求のループ処理を実行します。リソースが浪費されることを防ぐために、処理に要する時間にかかわらず、接続と切断は最高 200 回までしか実行されません。低速のネットワークでは、各統計について最低回数の反復処理を実行するために数秒かかることがあります。パフォーマンス統計は概算値であり、クライアントコンピュータとサーバコンピュータの両方がアイドル状態である方が統計値の精度は高くなります。
-stime	-s オプションと同じ情報が返されます。ただし、-st オプションでは、dbping が測定する各統計について要求のループ処理を実行する時間が秒単位で指定されます。このオプションでは、-s を使用した場合より精度の高いタイミング情報を取得できます。
-z	接続するために使用されたネットワーク通信プロトコルと、他の診断メッセージを表示します。このオプションは、Embedded SQL 接続が行われるときのみ使用できます。つまり、このオプションは、-m または -l と組み合わせて使用することはできません。

## 権限

なし。

## 備考

dbping ユーティリティは、接続の問題をデバッグするのに役立つツールです。これは、完全な接続文字列か部分的な接続文字列を取り、サーバの検索、またはデータベースへの接続が成功したかどうかを示すメッセージを返します。

このユーティリティは、Embedded SQL または ODBC 接続に対して使用できます。jConnect (TDS) 接続に対しては使用できません。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 関連情報

[接続プロパティの一覧 \[839 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

トラブルシューティング: Embedded SQL とネットワーク接続パフォーマンスのテスト方法 (dbping) [262 ページ]

トラブルシューティング: 接続文字列のテスト方法 (dbping) [261 ページ]

## 1.7.4.20 プロファイラユーティリティ (dbprof)

プロファイラを起動し、データベースのプロファイリングまたは以前のプロファイリングセッション情報の検証を行えるようにします。

### 構文

```
dbprof [ options ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容を難読化します。
-c "keyword=value; ..."	プロファイリングを開始する接続先のデータベースに対する接続パラメータを指定します。
-f filename	開くプロファイラファイル (.sqlap) を指定します。
-version	プロファイラのソフトウェアバージョン番号を表示します。

### 権限

すべてのユーザがプロファイラファイル (.sqlap) を開くことができますが、データベースのプロファイリングを開始するか、現在のプロファイリングセッションに接続するには、SYS\_RUN\_PROFILER\_ROLE システムロールを持つ必要があります。

### 備考

プロファイラは、データベースで発生するアクティビティをリアルタイムで記録する開発/診断ツールです。

### 例

次のコマンドは、プロファイラを起動し、サンプルデータベースに接続します。

```
dbprof -c "UID=DBA;PWD=sql;DBF=%SQLANY%SAMP17%demo.db"
```

次のコマンドは、プロファイラを起動し、保存されたプロファイリングセッションを開きます。

```
dbprof -f myprofilingsession.sqlap
```

## 関連情報

[SQL Anywhere プロファイラ \[1411 ページ\]](#)

[包括的プロファイリングセッションの実行 \(プロファイラ\) \[1413 ページ\]](#)

[データベースサーバの診断情報のスナップショットの SAP サポートへの送信 \(プロファイラ\) \[1418 ページ\]](#)

[ターゲットプロファイリングセッションの実行 \(プロファイラ\) \[1414 ページ\]](#)

[スタアドプロシージャ、関数、イベントおよびトリガのさまざまな実行結果の比較 \(プロファイラ\) \[1417 ページ\]](#)

[ターゲットプロファイリングセッション中に収集する SQL 文を指定する基準 \(プロファイラ\) \[1416 ページ\]](#)

## 1.7.4.21 サーバ列挙ユーティリティ (dblocate)

TCP/IP ネットワーク上のデータベースサーバを検索します。

### 構文

```
dblocate [ options ] [ host ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-d	検出したサーバごとに、サーバ名とアドレスをリストし、その後各サーバで実行されているデータベースをカンマで区切ってリストします。リストが 160 文字を超える場合は、トランケートされ、末尾に省略記号 (...) が表示されます。  SQL Anywhere 9.0.2 以前のデータベースサーバで実行されているデータベースや、起動時に -dh データベースオプションが指定されたデータベースは、リストされません。

オプション	説明
<code>-dn database-name</code>	<p>指定された名前のデータベースを実行しているサーバのサーバ名とアドレスをリストします。リストが 160 文字を超える場合は、truncate され、末尾に省略記号 (...) が表示されます。</p> <p>SQL Anywhere 9.0.2 以前のデータベースサーバで実行されているデータベースや、起動時に <code>-dh</code> データベースオプションが指定されたデータベースは、リストされません。</p>
<code>-dv</code>	<p>検出したサーバごとに、サーバ名とアドレスを表示し、さらに各サーバで実行されているデータベースを別の行にリストします。このリストはtruncate されないため、このオプションを使用すると、<code>-d</code> オプションではtruncate された部分を確認することができます。</p> <p>SQL Anywhere 9.0.2 以前のデータベースサーバで実行されているデータベースや、起動時に <code>-dh</code> データベースオプションが指定されたデータベースは、リストされません。</p>
<code>-n</code>	<p>コンピュータ名ではなく IP アドレスを出力にリストします。コンピュータ名の検索処理は低速であるため、パフォーマンスの向上につながる場合があります。</p>
<code>-ofilename</code>	<p>指定したファイルに、出力メッセージを追加します。</p>
<code>-pport-number</code>	<p>指定された TCP/IP ポート番号を使用しているサーバについてのみ、サーバ名とアドレスを表示します。指定する TCP/IP ポート番号は 1 ~ 65535 の範囲内にある必要があります。</p>
<code>-q</code>	<p>クワイエットモードで実行します (メッセージを表示しません)。</p>
<code>-sname</code>	<p>指定された名前のサーバについてのみ、サーバ名とアドレスを表示します。このオプションを使用する場合、<code>-ss</code> オプションは使用できません (両方のオプションを使用すると、サーバがまったく検出されない可能性があります)。</p>
<code>-SSsubstr</code>	<p>指定された部分文字列が名前に含まれるサーバについてのみ、サーバ名とアドレスを表示します。このオプションを使用する場合、<code>-s</code> オプションは使用できません (両方のオプションを使用すると、サーバがまったく検出されない可能性があります)。</p>
<code>-v</code>	<p>完全なサーバ名を表示します。デフォルトでは、dblocate の出力において、40 バイトを超えるデータベースサーバ名はtruncate されます。</p> <p>dblocate を含め、バージョン 9.0.2 以前のクライアントは、名前が 40 バイトを超えるバージョン 10.0.0 以降のデータベースサーバに接続できません。</p>

オプション	説明
host	<p>指定された IP アドレスまたはホスト名を持つコンピュータで実行されているデータベースサーバのみを列挙します。たとえば、次のコマンドは、コンピュータ jfrancis 上のサーバを検索します。</p> <pre>dblocate jfrancis</pre> <p>-n を指定するかどうかにかかわらず、ホスト名や IP アドレスはどのような形式でもかまいません。たとえば、IP アドレスが 1.2.3.4 であるコンピュータ myhost.mycompany.com で実行されているサーバがあるとします。このコンピュータ上で実行されているサーバだけを、mycompany.com ドメインのコンピュータからリストする場合は、dblocate myhost、dblocate myhost.mycompany.com、dblocate 1.2.3.4 のいずれも使用できます。</p>

## 権限

なし。

## 備考

サーバ列挙ユーティリティ (dblocate) は、直接接続されているネットワークを介して TCP/IP 上で実行されている SQL Anywhere データベースサーバを検索し、データベースサーバとそのアドレスのリストを出力します。このリストには代替サーバ名も含まれています。

ネットワークによっては、dblocate が結果を出力するまでに数秒かかることがあります。

### i 注記

Mac OS X で 2638 以外の TCP/IP ポートを使用しているデータベースサーバは、-p オプションでその TCP/IP ポートを指定しても、dblocate では検出されません。

別のサブネットにあるデータベースサーバを検出できるのは、dbns17 がすでに使用中の場合のみです。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

データベースサーバは、サーバ自体を LDAP サーバとして登録でき、企業内のすべてのサーバを追跡することができます。これにより、クライアントと dblocate は、サーバが WAN または LAN にあるかどうかにかかわらず、IP アドレスを指定せずにファイアウォールを介してサーバを検索できます。LDAP は TCP/IP でネットワークサーバ上でのみ使用されます。

同じサーバ名が複数回検索された場合、dblocate は -n オプションが指定されていなくても、各ホストの IP アドレスを表示します。同じサーバ名が見つかるのは、複数の IP アドレスを持つコンピュータ (たとえば、コンピュータに複数のネットワークカ

ードがある)でサーバを実行している場合、またはネットワークサーバをリモートコンピュータで実行し、同じ名前を持つパーソナルサーバをローカルコンピュータで実行している場合です。

-sb 0 で起動したデータベースサーバは dblocate では見つかりません。

## 関連情報

[ネームサーバとして LDAP を使用した接続 \[171 ページ\]](#)

[TDI Drivers \(NDIS 5.1\) \(TDI ドライバ \(NDIS 5.1\)\)](#)

[トラブルシューティング: Broadcast Repeater ユーティリティによるデータベースサーバの検索方法 \[254 ページ\]](#)

[-dh データベースオプション \[526 ページ\]](#)

[ServerPort \(PORT\) プロトコルオプション \[225 ページ\]](#)

[トラブルシューティング: データベースサーバの検出方法 \[251 ページ\]](#)

## 1.7.4.22 サーバライセンス取得ユーティリティ (dblic)

データベースサーバまたは Mobile Link サーバに、ソフトウェアライセンスを適用します。

### 構文

```
dblic [ options ] license-file [ "user-name" "company-name" ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。

オプション	説明
<p><code>-ltype</code></p>	<p>ソフトウェアのライセンス契約に記述されているライセンスモデルと一致するライセンスタイプを指定します。サポートされるライセンスタイプは次のとおりです。</p> <p><b>Perseat</b></p> <p>パーシートライセンスは、データベースサーバに対するクライアント接続の数を制限します。パーシートライセンスでは、コンピュータで使用可能なすべての CPU をネットワークデータベースサーバが使用できますが、実行中の SQL Anywhere エディションの最大許容数を超える CPU は使用できません。CPU に複数のコアまたはスレッドが含まれている場合は、CPU のすべてのコアまたはスレッドを使用することもできます。</p> <p>パーソナルサーバでは、1 つの CPU 上のコアは 4 つに制限されています。コアに複数のスレッドが含まれている場合は、そのコアのすべてのスレッドを使用することもできます。</p> <p><b>Core</b></p> <p>コアライセンスは、データベースサーバが使用できる独立したコアの数を制限します。コアベースのライセンスでは、ライセンスで指定された数までのコアをネットワークデータベースサーバが使用しますが、実行中の SQL Anywhere エディションの最大許容数を超える CPU は使用できません。</p> <p>コアライセンスを取得している場合、データベースサーバに対するクライアント接続の数の制限はありません。</p> <p>次の方法でデータベースサーバをさらに制限することができます。</p> <ul style="list-style-type: none"> <li>• <code>-gt</code> データベースサーバオプション</li> <li>• <code>-gta</code> データベースサーバオプション</li> <li>• <code>-gtc</code> データベースサーバオプション</li> <li>• ProcessorAffinity オプションを指定した <code>sa_server_option</code> システムプロシージャ</li> </ul> <p>このオプションは、<code>-k</code> オプションとは一緒に使用できません。</p>
<p><code>-kregistration-key</code></p>	<p>ライセンスキーを指定します。このオプションを使用して、データベースサーバのエディションを変更できます。たとえば、手元にある開発版をワークグループ版に変更する場合は、ワークグループ版の登録キーを指定できます。</p> <p>このオプションは、<code>-l</code> オプションとは一緒に使用できません。</p>
<p><code>-Ofilename</code></p>	<p>指定したファイルに、出力メッセージを追加します。</p>
<p><code>-q</code></p>	<p>クワイエットモードで実行します (メッセージを表示しません)。</p>
<p><code>-ulicense-number</code></p>	<p>ライセンスされるユーザまたはプロセッサの総数を指定します。ライセンスを追加する場合でも、追加ライセンスの数ではなく、総数を指定します。</p>

オプション	説明
license-file	ライセンスされているパーソナルデータベースサーバ、ネットワークデータベースサーバ、または Mobile Link サーバのサーバ実行プログラムまたはライセンスファイルのパスと名前を指定します。  ライセンスファイル名だけを入力することで、サーバ実行プログラムの現在のライセンス情報を表示できます。
user-name	ライセンスのユーザ名を指定します。この名前は、起動時にデータベースサーバメッセージウィンドウに表示されます。名前にスペースが含まれている場合、二重引用符で囲んでください。
company-name	ライセンスの会社名を指定します。この名前は、起動時にデータベースサーバメッセージウィンドウに表示されます。名前にスペースが含まれている場合、二重引用符で囲んでください。

## 権限

なし。

## 備考

サーバライセンス取得ユーティリティでは、SQL Anywhere データベースサーバまたは Mobile Link サーバにライセンスユーザまたはライセンスプロセッサを追加できます。このユーティリティは、ライセンス契約に基づいて、許可されたライセンスユーザ数またはライセンスプロセッサ数の範囲内で使用してください。このコマンドの実行によって、ライセンスが付与されることはありません。また、データベースサーバで使用できる CPU 数が、SQL Anywhere のエディション、-gt、-gtc、-gta の各データベースサーバオプション、または ProcessorAffinity オプションを指定した sa\_server\_option システムプロシージャの影響を受ける場合もあります。

このユーティリティでは、パーソナルデータベースサーバ、ネットワークデータベースサーバ、または Mobile Link サーバによって起動時に表示されるユーザ名や会社名を変更することもできます。

また、ライセンスファイル名だけを入力することで、パーソナルデータベースサーバやネットワークデータベースサーバを起動しないで、現在のライセンス情報を表示することも可能です。

ライセンス情報は、サーバの実行プログラムと同じディレクトリにある .lic ファイルに格納されます。サーバは、実行中の実行プログラムと同じベースファイル名を持つ .lic ファイルを探します。たとえば、データベースサーバの実行プログラムが myserver.exe という名前の場合、サーバは myserver.lic という名前のライセンスファイルを探します。デフォルトでは、次の名前が使用されます。

実行プログラム	ライセンスファイル名
パーソナルデータベースサーバ (dbeng17)	dbeng17.lic
ネットワークデータベースサーバ (dbsrv17)	dbsrv17.lic
Mobile Link サーバ (mlsrv17)	mlsrv17.lic

サーバを起動するときに、対応する .lic ファイルが使用できない場合は、サーバは起動されません。ライセンスファイルは、SQL Anywhere インストールプログラムによって作成されます。dblic ユーティリティは、既存のライセンスを修正するだけで、新しいライセンスファイルを作成することはありません。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

UNIX では、データベースサーバの実行プログラムはデフォルトで書き込みができないので、サーバライセンス取得 (dblic) ユーティリティを使用すると失敗します。実行プログラムは、サーバライセンス取得ユーティリティを使用する前に (chmod +w を使用するなどして) 書き込み可能にしてください。

キャッシュされた接続は、パーシートライセンスとしてカウントされます。

## 例

次のコマンドをデータベースサーバの実行プログラムと同じディレクトリで実行すると、ユーザ名 Sys Admin、会社名 My Co という設定で、50 ユーザのライセンスが、Microsoft Windows ネットワークデータベースサーバに適用されます。コマンドは、1 行に入力してください。

```
dblic -l perseat -u 50 "C:¥Program Files¥SQL Anywhere 17¥Bin32¥dbsrv17.lic" "Sys Admin" "My Co"
```

ライセンスの適用が成功すると、画面に次のメッセージが表示されます。

```
SQL Anywhere サーバライセンス取得ユーティリティバージョン 17.0.4.1691 ライセンスが正常に適用
されました。
Advanced Edition
アドオン: 高可用性
アドオン: インメモリモード
アドオン: スケールアウトノード
ライセンスされるノード: 50
ユーザ: Sys Admin
会社: My Co
インストールキー:
```

次のコマンドは、データベースサーバのライセンスについての情報を返します。

```
dblic "C:¥Program Files¥SQL Anywhere 17¥Bin32¥dbsrv17.lic"
```

## 関連情報

- [-gt データベースサーバオプション \[430 ページ\]](#)
- [-gtc データベースサーバオプション \[432 ページ\]](#)
- [-gta データベースサーバオプション \[431 ページ\]](#)

## 1.7.4.23 Linux 用サービスユーティリティ (dbsvc)

SQL Anywhere サービスの作成、変更、削除を行います。

### 構文

```
dbsvc [ modifier-options ] -d svc
```

```
dbsvc [ modifier-options ] -g svc
```

```
dbsvc [ modifier-options ] -l
```

```
dbsvc [ modifier-options ] -li
```

```
dbsvc [ modifier-options ] -lt
```

```
dbsvc [ modifier-options ] -status svc
```

```
dbsvc [ modifier-options ] -u svc
```

```
dbsvc [ modifier-options ] creation-options -W svc [ options ]
```

```
dbsvc [ modifier-options ] -X svc
```

主要オプション	説明
-d service-name	サービスリストから指定のサービスを削除します。-y を指定すると、確認メッセージを表示せずにサービスを削除します。実行中のサービスは削除できません。
-g service-name	サービスの定義を表示します。
-l	使用できる SQL Anywhere サービスをリストします。
-li	すべてのサービスインタフェースをリストします。
-lt	すべての SQL Anywhere サービスインタフェースをリストします。
-u service-name	service-name という名前のサービスを開始します。
-W executableparameters	新しいサービスを作成するか、同名のサービスが存在する場合はそれを上書きします。-y を指定すると、確認メッセージを表示せずに既存のサービスを上書きします。実行中のサービスは削除できません。 作成するサービスに適したパラメータを指定します。
-X service-name	service-name という名前のサービスを停止します。

作成オプション	説明
<code>-acct</code>	アカウントを指定します。すべてのサービスは、Linux アカウントの下で実行されます。独自に作成したアカウントで実行する場合は、 <code>-a</code> オプションでアカウントを指定する必要があります。
<code>-as</code>	Linux デモンアカウントの下でサービスを実行します。すべてのサービスは、Linux アカウントの下で実行されます。パスワードは必要ありません。 <code>-a</code> または <code>-as</code> のいずれかを必ず使用してください。
<code>-od</code>	システム情報ファイルのロケーションを指定します (必要な場合)。
<code>-pr</code>	Linux プロセスのナイスレベルを設定します。
<code>-rl</code>	サービスを起動するランレベルを指定します。
<code>-rService-name</code>	<p>dbsvc によって生成される init スクリプトの Required-Start ヘッダで、サービスのリストにサービスを追加します。このヘッダは、オペレーティングシステムでサービスの開始順序を決めるのに役立ちます。</p> <p>指定されたサービス名は、すでに存在するかどうかを検証されます。サービス名は、<code>SA_service-name</code> の形式で指定します。</p>
<code>-sstartup</code>	SQL Anywhere サービスの起動時の動作を設定します。起動時の動作は、[自動] または [手動] に設定できます。デフォルトは [手動] です。
<code>-status</code>	サービスの実行ステータスを返します。
<code>-ttype</code>	<p>このサービスのタイプを指定します。次のタイプから選択できます (代替名は括弧内に示します)。</p> <p><b>ネットワーク</b></p> <p>SQL Anywhere ネットワークデータベースサーバ (dbsrv17) のサービスを作成します。</p> <p><b>パーソナル (スタンドアロン)</b></p> <p>これがデフォルトです。SQL Anywhere パーソナルデータベースサーバ (dbeng17) のサービスを作成します。</p> <p><b>DBRemote</b></p> <p>SQL Remote Message Agent のサービスを作成します。</p> <p><b>MobiLink</b></p> <p>Mobile Link サーバ (mlsrv17) のサービスを作成します。</p> <p><b>DBMLSync (MLSync)</b></p> <p>Mobile Link 同期クライアント (dbmlsync) のサービスを作成します。</p> <p><b>DBNS</b></p> <p>Broadcast Repeater ユーティリティ (dbns17) のサービスを作成します。</p> <p><b>RSHOST</b></p> <p>Relay Server ステイトマネージャのサービスを作成します。</p> <p><b>RSOE</b></p> <p>Outbound Enabler のサービスを作成します。</p>

変更オプション	説明
-cm	サービスの作成に使用するコマンドを表示します。このオプションを使用すると、作成コマンドをファイルに出力できます。このファイルを使用して、別のコンピュータにサービスを追加したり、変更が加えられたサービスを元の状態にリストアしたりすることができます。-cm とともに -g オプションまたは -l オプションを指定しないとコマンドは失敗します。-g を指定すると、指定のサービスの作成コマンドが表示されるのに対し、-l を指定するとすべてのサービスの作成コマンドが表示されます。
-q	データベースサーバメッセージウィンドウにメッセージを表示しないようにします。既存のサービスの変更時または削除時にこのオプションを指定する場合、-y も指定しないと操作は失敗します。
-y	確認メッセージを表示することなく、処理を実行します。このオプションは、-w または -d オプションと一緒に使用できます。既存のサービスの変更時または削除時に -q を指定する場合、-y も指定しないと操作は失敗します。
-iinterface	使用するサービスインタフェースを指定します。systemd または lsb のいずれかの値を指定します。使用できる場合、デフォルトは systemd です。使用できない場合は、デフォルトは lsb となります。

## 権限

なし。

## 備考

サービスは、一連のオプションを使ってデータベースサーバやその他のアプリケーションを実行します。このユーティリティを使用して、Linux 上で動作している SQL Anywhere サービスを包括的な方法で管理できます。

通常、サービスの作成環境と実行環境は異なるため、サービスの作成時にはデータベースファイルに完全に修飾された名前を付ける必要があります。また、データソース名にはスペースを入れないください。サービス名、サービス引数、SQL Anywhere のインストールパスでは、スペースおよび特殊文字がサポートされます。

### i 注記

dbsvc コマンドにシェル固有の文字 (セミコロンまたは括弧) が含まれている場合は、サービスに対するパラメータ用に設定ファイルを使用してください。

サービスを作成、編集、削除するには、/etc/init.d ディレクトリに対するパーミッションが必要です。

SQL Anywhere サービスが Linux で実行されている場合、PID ファイルは /var/run ディレクトリで作成されます。このファイルには、dbsvc プロセスの PID が含まれます。ファイルの名前は SA\_service-name.pid です。このファイルを他の Linux ツールで使用して、プロセスを検索したり、サービスをモニタしたりできます。

dbsvc ユーティリティでは、ほとんどの Linux サービスと同じように、サービスファイルが `/etc/init.d` に作成されます。サービス名は、`SA_service-name` の形式で指定します。たとえば、myserv という名前のサービスを作成した場合、このサービスを起動するには、次のようなコマンドを発行します。

```
/etc/init.d/SA_my_serv start
```

次のコマンドは、サービスのステータスを返します。

```
/etc/init.d/SA_my_serv status
```

次のコマンドは、サービスの使用状況に関する情報を返します。

```
/etc/init.d/SA_my_serv
```

### 例

指定のサーバを指定のパラメータで起動する、myserv という名前のパーソナルサーバサービスを作成します。このサーバは LocalSystem ユーザとして実行されます。

```
dbsvc -as -w myserv -n myeng -c 8m "/tmp/demo.db"
```

mynetworkserv という名前のネットワークサーバサービスを作成します。このサーバはローカルアカウントで実行され、コンピュータを再起動すると自動的に起動します。

```
dbsvc -as -t network -w mynetworkserv -x tcpip -c 8m "/tmp/demo.db"
```

サービス myserv についての詳細をすべてリストします。

```
dbsvc -g myserv
```

myserv という名前のサービスを、確認メッセージを表示せずに削除します。

```
dbsvc -y -d myserv
```

mysyncservice という名前のサービスを作成します。

```
dbsvc -as -t dbmlsync -w mysyncservice -c "/tmp/CustDB.db" -o syncinfo.txt
```

service\_1 サービスを作成するコマンドを生成し、そのコマンドをコンソールに出力します。

```
dbsvc -cm -g service_1
```

コンソールには次のように表示されます。

```
'dbsvc -t Standalone -as -y -w "service_1" -n'
```

dbsvc を使用してサービスを起動します。

```
dbsvc -u myserv
```

dbsvc を使用してサービスを停止します。

```
dbsvc -x myserv
```

dbsvc を使用してサービスを取得します。

```
dbsvc -status myserv
```

## 関連情報

[設定ファイル \[1066 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

[Broadcast Repeater ユーティリティ \(dbns17\) \[1077 ページ\]](#)

## 1.7.4.24 Windows 用サービスユーティリティ (dbsvc)

SQL Anywhere サービスの作成、変更、削除を行います。

### 構文

```
dbsvc [ modifier-options ] -d svc
```

```
dbsvc [ modifier-options ] -g svc
```

```
dbsvc [ modifier-options ] -l
```

```
dbsvc [ modifier-options ] -u svc
```

```
dbsvc [ modifier-options ] creation-options -w svc details
```

```
dbsvc [ modifier-options ] -x svc
```

details:

```
<full-executable-path> [ options ]
```

主要オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。 設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-d service-name	サービスリストから指定のサービスを削除します。-y を指定すると、確認メッセージを表示せずにサービスを削除します。実行中のサービスは削除できません。
-g service-name	パスワード以外のサービスの定義をリストします。
-l	使用できる SQL Anywhere サービスをリストします。

主要オプション	説明
<code>-Uservice-name</code>	<code>service-name</code> という名前のサービスを開始します。
<code>-Wexecutableparameters</code>	<p>新しいサービスを作成するか、同名のサービスが存在する場合はそれを上書きします。<code>-y</code> を指定すると、確認メッセージを表示せずに既存のサービスを上書きします。実行中のサービスは削除できません。</p> <p>サービスとして使用する実行プログラムのフルパスを指定します。これは、その下でサービスが実行されるアカウントのパスの中に適切な SQL Anywhere インストールディレクトリがない場合があるからです。</p> <p>作成するサービスに適したパラメータを指定します。</p>
<code>-Xservice-name</code>	<code>service-name</code> という名前のサービスを停止します。
作成オプション	説明
<code>-aacct</code>	<p>Microsoft Windows アカウントを指定します。すべてのサービスは、Microsoft Windows アカウントの下で実行されます。作成したアカウントの下で実行する場合は、<code>-a</code> オプションでアカウントを指定し、<code>-p</code> オプションでパスワードを指定します。</p> <p>サービスとしてログインするパーミッションは、LocalSystem アカウント以外のすべてのアカウントで必要です。サービスとしてログインするパーミッションがアカウントで有効になっていない場合は、有効にするよう求めるプロンプトが表示されます。<code>-y</code> オプションを同時に指定した場合は、プロンプトは表示されず、<code>dbsvc</code> によってサービスとしてログインするパーミッションの付与が試みられます。<code>-y</code> オプションを指定しないで <code>-q</code> オプションを指定した場合は、サービスとしてログインするパーミッションを付与するよう求めるプロンプトが表示されず、<code>dbsvc</code> の実行に失敗します。</p>
<code>-as</code>	Microsoft Windows LocalSystem アカウントの下でサービスを実行します。パスワードは必要ありません。 <code>-a</code> または <code>-as</code> のいずれかを必ず使用してください。すべてのサービスは、Microsoft Windows アカウントの下で実行されます。
<code>-i</code>	<p>アイコンが表示され、これをダブルクリックするとデータベースサーバメッセージウィンドウが表示されます。</p> <p><b>i 注記</b></p> <p>Windows Vista では、これは利用できなくなりました。</p>
<code>-ppassword</code>	サービスが実行される時のアカウントのパスワードを指定します。このオプションは、 <code>-a</code> オプションと一緒に使用します。
<code>-rgdependency,...</code>	作成するサービスが起動できるよう事前に起動しておく必要のある 1 つ以上の起動順序グループを指定します。
<code>-rsdependency,...</code>	<p>リストに含まれるすべてのサービスが起動してから、作成したサービスの起動を許可するように指定します。</p> <p>サービス名の表示名を指定できます。サービス名が検証され、すでに存在するかどうかを確認されます。指定されたサービス名が見つからない場合、表示名が一致するサービスが存在するかどうかサービスユーティリティ (<code>dbsvc</code>) によってチェックされます。</p> <p>接続が正しく開始することを確認するため、TCP/IP 用のトランスポートデータインタフェース (TDI) で依存性を作成してください。</p>

作成オプション	説明
-sstartup	SQL Anywhere サービスの起動時の動作を設定します。起動時の動作は [自動]、[手動]、または [無効] に設定できます。デフォルトは [手動] です。
-sdescription	サービスの説明を示します。説明は、Windows のサービスマネージャに表示されます。
-snname	<p>サービスの名前を示します。この名前は、Windows のサービスマネージャに表示されます。-sn オプションを指定しない場合、デフォルトのサービス名は SQL Anywhere - svc です。たとえば、次のサービスにはデフォルトで SQL Anywhere - myserv という名前が付けられます。</p> <pre> dbsvc -as -w myserv "C:\Program Files\SQL Anywhere 17\Bin32\dbeng17.exe" </pre> <p>サービス名 myserv が Windows のサービスマネージャに表示されるようにするには、次のコマンド (全体を 1 行に入力) を実行する必要があります。</p> <pre> dbsvc -as -sn myserv -w myserv "C:\Program Files\SQL Anywhere 17\Bin32\dbeng17.exe" </pre>
-ttype	<p>このサービスのタイプを指定します。次のタイプから選択できます (代替名は括弧内に示します)。</p> <p><b>ネットワーク</b></p> <p>SQL Anywhere ネットワークデータベースサーバ (dbsrv17) のサービスを作成します。</p> <p><b>パーソナル (スタンドアロン)</b></p> <p>これがデフォルトです。SQL Anywhere パーソナルデータベースサーバ (dbeng17) のサービスを作成します。</p> <p><b>DBRemote</b></p> <p>SQL Remote Message Agent のサービスを作成します。</p> <p><b>MobiLink</b></p> <p>Mobile Link サーバ (mlsrv17) のサービスを作成します。</p> <p><b>DBMLSync (MLSync)</b></p> <p>Mobile Link 同期クライアント (dbmlsync) のサービスを作成します。</p> <p><b>DBNS</b></p> <p>Broadcast Repeater ユーティリティ (dbns17) のサービスを作成します。</p> <p><b>DBLSN (LSN)</b></p> <p>Mobile Link Listener ユーティリティのサービスを作成します。</p> <p><b>DBVSS (VSS)</b></p> <p>SQL Anywhere VSS ライターのサービスを作成します。</p> <p><b>RSOE</b></p> <p>Outbound Enabler のサービスを作成します。</p> <p><b>MLAGENT</b></p> <p>Mobile Link エージェントのサービスを作成します。</p>

変更オプション	説明
-cm	<p>サービスの作成に使用するコマンドを表示します。このオプションを使用すると、作成コマンドをファイルに出力できます。このファイルを使用して、別のコンピュータにサービスを追加したり、変更が加えられたサービスを元の状態にリストアしたりすることができます。-cmとともに-gオプションまたは-lオプションを指定しないとコマンドは失敗します。-gを指定すると、指定のサービスの作成コマンドが表示されるのに対し、-lを指定するとすべてのサービスの作成コマンドが表示されます。</p> <p>指定のサービスが存在しない場合は、サービスを削除するコマンドが生成されます。たとえば、コンピュータにservice_1が存在しない場合、dbsvc -cm -g service_1はservice_1サービスを削除する次のコマンドを返します。</p> <pre>dbsvc -y -d "service_1"</pre> <p>サービスが LocalSystem アカウントを使用しない場合は、パスワードを取り出すことはできないため、生成されるコマンドにはパスワードは含まれません。-a user -p passwordを使用してサービスを作成した場合、出力には-a userのみが含まれます。</p>
-ofilename	<p>サービスユーティリティ (dbsvc) の出力を指定されたファイルに書き込みます。-o オプションは、-d、-g、-l、-u、-w、-x の前に指定してください。例:</p> <pre>dbsvc -o out1.txt -y -as -w mysvc "%SQLANY17% ¥bin32¥dbsrv17" -n mysrv -o c:¥out2.txt</pre> <p>この場合は、dbsvc の出力が out1.txt に書き込まれ、データベースサーバの出力がデータベースサーバの起動時に c:¥out2.txt に書き込まれます。</p>
-q	<p>データベースサーバメッセージウィンドウにメッセージを表示しないようにします。-q を指定する場合は、-o オプションを使用してメッセージを書き込むファイルを指定するようおすすめます。既存のサービスの変更時または削除時にこのオプションを指定する場合、-y も指定しないと操作は失敗します。</p>
-y	<p>確認メッセージを表示することなく、処理を実行します。このオプションは、-w または -d オプションと一緒に使用できます。既存のサービスの変更時または削除時に -q を指定する場合、-y も指定しないと操作は失敗します。</p>

## 備考

サービスは、一連のオプションを使ってデータベースサーバやその他のアプリケーションを実行します。このユーティリティを使用して、Windows 上で動作している SQL Anywhere サービスを包括的な方法で管理できます。サービスユーティリティ (dbsvc) を使用するには、ローカルコンピュータ上で Administrators グループのメンバーである必要があります。

サービスユーティリティにアクセスするには、次の方法があります。

- SQL Central の [サービス作成ウィザード](#) を使用します。
- コマンドプロンプトで、サービスユーティリティを使用します (dbsvc)。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 例

指定のサーバを指定のパラメータで起動する、myserv という名前のパーソナルサーバサービスを作成します。このサーバは LocalSystem ユーザとして実行されます。

```
dbsvc -as -w myserv
"C:¥Program Files¥SQL Anywhere 17¥Bin32¥dbeng17.exe"
-n myeng -c 8m "c:¥temp¥mysample.db"
```

mynetworkserv という名前のネットワークサーバサービスを作成します。このサーバはローカルアカウントで実行され、コンピュータを再起動すると自動的に起動します。

```
dbsvc -as -s auto -t network -w mynetworkserv
"C:¥Program Files¥SQL Anywhere 17¥Bin32¥dbsrv17.exe"
-x tcpip -c 8m "c:¥temp¥mysample.db"
```

サービス myserv についての詳細をすべてリストします。

```
dbsvc -g myserv
```

myserv という名前のサービスを、確認メッセージを表示せずに削除します。

```
dbsvc -y -d myserv
```

Workstation サービスと TDI グループに依存するサービスを作成します。

```
dbsvc -rs lanmanworkstation -rg TDI -w ...
```

相互に依存する 2 つのサービスを作成します。

```
dbsvc -sn Aardvark1 -as -w aardsvc1 dbeng17 c:¥databases¥aard1.db
dbsvc -sn Aardvark2 -as -rs aardsvc1 -w aardsvc2 dbeng17 c:¥databases¥aard2.db
```

Aardvark2 サービス (aardsvc2) が起動されると、Aardvark2 は Aardvark1 に依存するため、Aardvark1 サービス (aardsvc1) が自動的に起動します。同様に、Aardvark1 サービス (aardsvc1) を停止する必要がある場合は、Aardvark2 サービス (aardsvc2) を最初に停止する必要があります。

mysyncservice という名前のサービスを作成します。

```
dbsvc -as -s manual -t dbmlsync -w mysyncservice
"C:¥Program Files¥SQL Anywhere 17¥Bin32¥dbmlsync.exe"
-c "SQL Anywhere 17 CustDB;PWD=sql"
```

service\_1 サービスを作成するためのコマンドを生成し、それを restore-service.bat というファイルに出力します。

```
dbsvc -cm -g service_1 > restore-service.bat
```

restore-service.bat ファイルには以下が含まれています。

```
dbsvc -t Standalone -s Manual -as -y -w "service_1"
"C:¥Program Files¥SQL Anywhere 17¥Bin32¥dbeng17.exe"
```

手動で起動される Mobile Link Listener サービスを作成します。

```
dbsvc -as -i -w myListener
```

```
"C:\Program Files\SQL Anywhere 17\Bin32\dblsn.exe" "@c:\temp\dblsn.opt"
```

myListener サービスを起動します。

```
dbsvc -u myListener
```

myListener サービスを停止します。

```
dbsvc -x myListener
```

データベースサーバの起動時に自動的に開始するボリュームシャドウコピーサービス (VSS) サービスを作成します。

```
dbsvc -as -s Automatic -t vss -w SAVSSWriter  
"C:\Program Files\SQL Anywhere 17\Bin32\dbvss17.exe"
```

## 関連情報

[SQL Anywhere ボリュームシャドウコピーサービス \(VSS\) \[900 ページ\]](#)

[Windows サービス \[340 ページ\]](#)

[Windows サービスの作成 \(SQL Central の場合\) \[341 ページ\]](#)

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

[Broadcast Repeater ユーティリティ \(dbns17\) \[1077 ページ\]](#)

## 1.7.4.25 サーババックグラウンド起動ユーティリティ (dbspawn)

バックグラウンドでデータベースサーバを起動します。

### 構文

```
dbspawn [ options ] server-command
```

オプション	説明
-f	デフォルトのデータベースサーバがすでに存在する場合でも、dbspawn を使用してデータベースサーバを起動します。
-p	データベースサーバの起動後に返される、データベースサーバプロセスのオペレーティングシステムプロセス ID を指定します。例: <pre>新しいプロセス ID は 306 です</pre>
-q	クワイエットモードで実行します (メッセージを表示しません)。

オプション	説明
server-command	データベースサーバを開始するためのコマンドラインを指定します。

## 権限

なし。

## 備考

dbspawn ユーティリティは、バックグラウンドでサーバを起動するために提供されています。このユーティリティは、バックグラウンドでサーバを起動し、終了コードとして 0 (成功) または非ゼロ (失敗) を返します。新しいサーバが正常に起動された場合は、データベースサーバの初期処理が完了し、要求を受信できる状態になるまで、dbspawn は終了しません。

デフォルトでは、dbspawn はデフォルトデータベースサーバが現在実行中のコンピュータでは新しいサーバを起動できません。デフォルトデータベースサーバが実行中のコンピュータでサーバを起動するには、-f オプションを指定します。

dbspawn ユーティリティは、バッチファイルからサーバを起動するのに役立ちます。特に、バッチファイルの後続コマンドが要求を受け入れるサーバを必要とする場合に便利です。

指定するパスに 1 つ以上のスペースが含まれている場合は、パス全体を二重引用符で囲む必要があります。例:

```
dbspawn dbeng17 "c:¥my databases¥mysalesdata.db"
```

指定したパスにスペースがない場合、引用符は必要ありません。

dbspawn ユーティリティでは、@data オプション自身はサポートされませんが、生成されるコマンドにオプションを指定できます。次に例を示します。

```
dbspawn dbeng17 @myconfig.ini
```

## 関連情報

[SQL Anywhere データベースサーバ実行プログラム \(dbsrv17, dbeng17\) \[365 ページ\]](#)

## 1.7.4.26 サーバ停止ユーティリティ (dbstop)

データベースまたはデータベースサーバを停止します。

### 構文

```
dbstop [ options ] [ server-name ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-C"keyword=value;..."	接続文字列を指定します。ネットワークサーバを停止する場合は、サーバを停止するパーミッションのあるユーザ ID を接続文字列に指定する必要があります。デフォルトでは、ネットワークサーバを停止するには SERVER OPERATOR システム権限が必要であり、すべてのユーザがパーソナルサーバを停止できますが、-gk サーバオプションを使用するとその動作を変更できます。  接続パラメータを指定する場合は、サーバ名を指定しないでください。
-d	データベースサーバは停止しません。代わりに、接続文字列で指定したデータベースだけを停止します。-gd サーバオプションは、データベースファイルの停止に必要なパーミッションを制御します。
-Ofilename	指定したファイルに、出力メッセージを追加します。
-q	クワイエットモードで実行します (メッセージを表示しません)。
-x	サーバへのアクティブな接続がある場合、サーバを停止しません。このオプションを使用すると、dbstop はアクティブな接続があってもプロンプトを表示しません。
-y	サーバにアクティブな接続がある場合でも、サーバを停止します。これは、接続パラメータに Unconditional=YES を含めることと同等です。
server-name	現在のコンピュータ上で稼働中のデータベースサーバの名前を指定します。シャットダウン時にパーミッションが一切必要にならないように、データベースサーバを起動してください。パーソナルデータベースサーバは、デフォルトではこのモードで起動します。ネットワークデータベースサーバの場合は、-gk all オプションを指定してください。  サーバ名を指定する場合は、接続パラメータを指定しないでください。

## 権限

ネットワークサーバを停止する場合は、サーバを停止するパーミッションのあるユーザ ID を接続文字列に指定します。デフォルトでは、ネットワークサーバを停止するには SERVER OPERATOR システム権限が必要です。すべてのユーザがパーソナルサーバを停止できます。-gk サーバコマンドラインオプションを使用するとデフォルトの動作を変更できます。

## 備考

サーバ停止ユーティリティは、データベースサーバを停止します。-d オプションを使用して、指定したデータベースを停止できます。

サーバ停止ユーティリティはコマンドプロンプトでのみ実行できます。ウィンドウベースの環境では、データベースサーバメッセージウィンドウでシャットダウンをクリックしてデータベースサーバを停止できます。

オプションを使用して、アクティブな接続がある場合にも接続を停止するかどうか、またサーバを停止するのかデータベースのみを停止するのかを制御できます。

サーバ上にアクティブな接続がある場合の dbstop の動作を制御できます。アクティブな接続がある場合は、サーバを停止するよう求めるプロンプトが dbstop によって表示されます。この動作を変更するには、-x オプションと -y オプションを使用します。

データベースサーバを停止できる場合、すべてのデータベースが終了し、データベースサーバが停止して、同じ名前のサーバを起動して同じデータベースを実行できる状態になってから、dbstop が完了します。dbstop が完了しても、データベースサーバプロセスは動作したままとなることがあり、そのリソース (-o サーバオプションで指定された出力ファイルなど) も使用中の状態が続くことがあります。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

dbstop で SQLCONNECT 環境変数を使用するには、-c オプションを指定します。それ以外の場合に -c オプションを使用すると、予期しない結果となることがあります。

### 例

myserver という名前のサーバを実行していて、データベースは起動していないとします。サーバを停止するには、ユーティリティデータベース名と、サーバ起動時の -su データベースサーバオプションで指定したユーザ ID とパスワードを指定します (たとえば、-su udba,upasswd)。

```
dbstop -c "Server=myserver;DBN=utility_db;UID=udba;PWD=upasswd"
```

myserver という名前のサーバを実行していて、データベース *demo.db* が動作中であるとします。このサーバとデータベースを停止するには、次のように指定します。

```
dbstop -c "UID=DBA;PWD=sql;Server=myserver"
```

myserver という名前のパーソナルサーバを実行しているとします。接続が確立されている状態で、このサーバとデータベースを停止するには、次のように指定します。

```
dbstop -y myservers
```

myserver という名前のサーバを実行していて、データベース *demo.db* が動作中であるとします。他のデータベースやサーバを停止しないでデータベース *demo.db* だけを停止するには、次のコマンドを実行します。

```
dbstop -c "UID=DBA;PWD=sql;Server=myserver;DBN=demo" -d
```

## 関連情報

[設定ファイル \[1066 ページ\]](#)

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

[Unconditional \(UNC\) 接続パラメータ \[109 ページ\]](#)

[-gd データベースサーバオプション \[414 ページ\]](#)

[-gk データベースサーバオプション \[417 ページ\]](#)

## 1.7.4.27 サポートユーティリティ (dbsupport)

エラーや診断情報に関するパフォーマンスデータをテクニカルサポートに送信します。

### 構文

```
dbsupport [ options ] operation [ operation-specific-option ]
```

```
dbsupport configuration-options
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-ofilename	指定したファイルにログ出力メッセージを送信します。
-q	重要なログ出力メッセージだけを表示します。

操作	説明
<p><code>-econfiguration-option</code></p>	<p>指定された設定オプションの設定を表示します。たとえば次のコマンドを実行して、可能な場合にはプロンプトを表示するように <code>dbssupport</code> を設定したとします。</p> <pre data-bbox="762 472 1331 528">dbssupport -cc promptdefy</pre> <p>この場合、<code>dbssupport -ecc</code> コマンドを実行すると、次の設定が返されます。</p> <pre data-bbox="762 629 1331 685">-cc "promptdefy"</pre> <p>設定ファイルの現在の値が明示的に設定されていない場合は、エコーされません。</p>
<p><code>-isubmission-ID [-rN]</code></p>	<p>テクニカルサポートに送信されたクラッシュレポートのステータスをチェックします。</p> <p>たとえば、次のコマンドは送信 ID 66 のステータスを返します。</p> <pre data-bbox="762 931 1331 987">dbssupport -is 66</pre>
<p><code>-iu [-rN]</code></p>	<p>ソフトウェアのビルドに対する更新をチェックします。次の更新があります。</p> <p><b>サポートパッケージ</b></p> <p>サポートパッケージは、1つまたは複数のバグフィックスを含むソフトウェアのサブセットです。これらのバグフィックスは、更新のリリースノートにリストされます。バグフィックス更新を適用できるのは、同じバージョン番号を持つインストール済みのソフトウェアに対してだけです。ソフトウェアでテストを行っている場合は、そのソフトウェアでアプリケーションを徹底的にテストしないかぎり、アプリケーションとともにこれらのファイルを配布しないでください。</p> <p><b>マイナーリリース</b></p> <p>マイナーリリースとは、インストールされたソフトウェアを古いバージョンから同じ主要バージョン番号 (バージョン番号形式は <code>major.minor.build</code>) に更新するソフトウェアの完全セットです。バグフィックスとその他の変更については、アップグレードのリリースノートにリストされます。</p> <p>Interactive SQL または SQL Central を使用して更新をチェックすることもできます。</p>
<p><code>-lc</code></p>	<p>テクニカルサポートに送信されていないすべてのクラッシュレポートのリストを作成します。リストされるレポート名には、<code>-sc</code> オプションを使用できます。</p>

操作	説明
-ls	<p>送信されたすべてのレポートの送信 ID のリストを作成します。例:</p> <pre data-bbox="863 412 1455 456">dbsupport -ls</pre> <p>このコマンドは次のような情報を返します。</p> <pre data-bbox="863 533 1455 949">SQL Anywhere Support Utility Version 17.0.4.1691 Submission ID: 217756 reported: Wed Jul 29 15:54:43 2015 Submission ID: 217757 reported: Wed Jul 29 15:57:07 2015 Submission ID: 217759 reported: Wed Jul 29 16:14:24 2015 Submission ID: 217760 reported: Wed Jul 29 16:14:30 2015 Submission ID: 217761 reported: Wed Jul 29 16:28:12 2015 Submission ID: 221030 reported: Mon Sep 07 12:34:59 2015 Submission ID: 221031 reported: Mon Sep 07 12:35:05 2015</pre>
-pcfilename	クラッシュレポートの情報を表示します。このオプションを使用すると、情報を確認してから送信できます。
-pd	収集された診断情報を表示します。このオプションを使用すると、情報を確認してから送信できます。
-pssubmission-ID	<p>送信された特定のレポートに関する情報を表示します。たとえば、コマンド <code>dbsupport -ps 4</code> を実行すると、送信 4 に関する情報が返されます。</p> <pre data-bbox="863 1281 1455 1393">SQL Anywhere Support Utility Version 17.0.4.1691 Submission ID: 217756 reported: Wed Jul 29 15:54:43 2015</pre>
-sa [ -rnumber-of-submission-retries ]	診断ディレクトリに保存されているすべてのクラッシュレポートと診断情報を送信します。
-screportname [ -rnumber-of-submission-retries ] [ -nr   -rrN ]	<p>クラッシュレポートと診断情報を送信します。例:</p> <pre data-bbox="863 1572 1455 1639">dbsupport -sc SA17_20151220_133828_32116</pre> <p>-lc オプションを使用すると、送信されていないレポートのリストを参照できます。</p>
-sd [ -rnumber-of-submission-retries ]	診断情報だけを送信します。

設定オプション	説明
<pre>-cc [ autosubmit   no   promptDefY   promptDefN ]</pre>	<p>dbssupport によるプロンプトの動作を変更します。このオプションでは次のいずれかのオプションを指定できます。</p> <p><b>autosubmit</b></p> <p>クラッシュレポートを自動的に送信し、診断レポートを定期的に送信します。</p> <p><b>no</b></p> <p>レポートを送信してよいかどうかを確認しません。レポートは送信されません。</p> <p><b>promptDefY</b></p> <p>可能な場合、レポートを送信してよいかどうかを確認します。回答がない場合は、レポートを送信します。</p> <p><b>promptDefN</b></p> <p>可能な場合、レポートを送信してよいかどうかを確認します。回答がない場合は、レポートを送信しません。これがデフォルトです。</p> <p>たとえば、埋め込みの SQL Anywhere をアプリケーションで使用する場合は、レポートを送信しないようにサポートユーティリティを設定することもできます。</p> <p>-cc オプションを指定すると、その値がデフォルト値としてサポートユーティリティ (dbssupport) に使用されます。設定は、診断ディレクトリ内の dbssupport.ini ファイルに保存されます。</p> <p>次のコマンドは、レポートを送信せず、ユーザに対してレポート送信の確認メッセージを表示しないように、サポートユーティリティを設定します。</p> <pre>dbssupport -cc no</pre>
<pre>-cdretry-delay</pre>	<p>レポートの送信が失敗した場合にリトライするまでの遅延時間を秒単位で指定します。デフォルトの遅延時間は 30 秒です。</p> <p>このオプションを指定すると、その値がデフォルト値としてサポートユーティリティに使用されます。設定は、診断ディレクトリ内の dbssupport.ini ファイルに保存されます。</p> <p>次のコマンドでは、送信が失敗した場合に 3 秒間隔で最大 4 回リトライするように指定しています。</p> <pre>dbssupport -cr 4 -cd 3</pre>
<pre>-ceemail-address;email-server[:port ][:user-id;password ]</pre>	<p>クラッシュの発生後に電子メールを送信するアドレスを指定します。電子メールは、email-server SMTP サーバを使用して送信されます。必要に応じて、使用するポートや、SMTP サーバでの認証に使用するユーザ ID とパスワードを指定できます。</p>

設定オプション	説明
<code>-cet</code>	<code>-ce</code> オプションで指定した電子メール設定をテストします。
<code>-cidcustomer-id</code>	<p>送信レポート内でユーザ自身を識別するための文字列を指定します。このオプションを指定すると、その値がデフォルト値としてサポートユーティリティに使用されます。設定は、診断ディレクトリ内の <code>dbsupport.ini</code> ファイルに保存されます。</p> <p>次の例は、<code>dbsupport</code> に対してカスタマ ID 文字列を指定しています。</p> <pre>dbsupport -cid myid@company.com dbsupport -cid "MyClientApp 1.0"</pre>
<code>-cid-</code>	<code>dbsupport.ini</code> ファイルからカスタマ ID 文字列を削除します。
<code>-cp { host [:port ]   autodetect }</code>	<p>エラーレポートの送信に使用する HTTP プロキシのホストとポートを設定します。デフォルトのポートは 80 です。</p> <p>Windows で <code>-cp autodetect</code> を指定し、プロキシサーバとポートが Microsoft Internet Explorer 用にすでに設定されている場合、<code>dbsupport</code> はシステム設定を使用してプロキシサーバとポートを設定します。</p> <p>UNIX で <code>-cp autodetect</code> を指定し、<code>HTTP_PROXY</code> 環境変数が設定されている場合、<code>dbsupport</code> は環境変数の設定を使用してプロキシサーバとポートを設定します。</p>
<code>-cp-</code>	<code>dbsupport.ini</code> ファイルから HTTP プロキシのホストとポートの設定を削除します。
<code>-Crnumber-of-submission-retries</code>	<p>送信が失敗した場合のリトライ回数を指定します。</p> <p>このオプションを指定すると、その値がデフォルト値としてサポートユーティリティに使用されます。設定は、診断ディレクトリ内の <code>dbsupport.ini</code> ファイルに保存されます。</p> <p>次のコマンドでは、操作が失敗した場合に 3 秒間隔で最大 4 回リトライするように指定しています。</p> <pre>dbsupport -cr 4 -cd 3</pre>
操作固有のオプション	説明
<code>-ac</code>	送信にコメントを追加します。
<code>-af</code>	送信にファイルを付加します。 <code>-af</code> オプションを複数回指定すると、送信に複数のファイルを付加できます。

操作固有のオプション	説明
<code>-nr</code>	<p>dbsupport がデータベースサーバで送信のステータスをチェックしないことを指定します。たとえば、次のコマンドは、レポートを送信するだけで、その送信のステータスについてはチェックしません。</p> <pre>dbsupport -nr -sc SA17_20151220_133828_32116</pre> <p>デフォルトでは、送信しようとしている問題について解決策がすでに提示されていないかどうかチェックします。</p>
<code>-rnumber-of-submission-retries</code>	<p>dbsupport がレポートの送信をリトライする最大回数を指定します。0 を指定すると、リトライ回数は無限となります。デフォルト値は 10 です。dbsupport.ini ファイルに <code>-cr</code> オプションが保存されている場合、<code>-r</code> を指定すると <code>-cr</code> の値は上書きされます。</p>
<code>-rdretry-delay</code>	<p>レポートを再送するまでの dbsupport の待ち時間を秒単位で指定します。デフォルト値は 30 です。dbsupport.ini ファイルに <code>-cd</code> オプションが保存されている場合、<code>-rd</code> を指定すると <code>-cd</code> の値は上書きされます。</p>
<code>-rnumber-of-submission-response-retries</code>	<p>dbsupport が送信に対する応答の取得をリトライする最大回数を指定します。0 を指定すると、リトライ回数は無限となります。デフォルト値は 10 です。</p>

## 権限

なし。

## 備考

クラッシュレポートや診断情報レポートなどのパフォーマンスデータは、製品の品質向上に役立てられます。

SQL Anywhere プログラムで致命的なエラーまたはクラッシュが発生した場合は、エラー発生時のプログラムに関する技術情報を含むエラーレポートが作成されます。エラーレポートが作成されると、そのレポートをテクニカルサポートに提出するよう求めるプロンプトが表示されます。

サポートユーティリティ (dbsupport) を使用して、このレポートを提出することもできます。エラーレポートを送信すると、致命的なエラーやアサーションの原因の究明に役立つことがあります。エラーレポートが提出されると、テクニカルサポートの問い合わせで使用できる提出 ID を受け取ります。問題をテクニカルサポートに報告するときにこのエラーファイルを送信することで、テクニカルサポートが問題を速やかに処理することができます。テクニカルサポートに連絡しない場合でも、エラーレポートや診断レポートを送信して、製品の品質向上にご協力ください。

サポートユーティリティ (dbsupport) では、次のタスクを実行できます。

- 診断情報とクラッシュレポートを送信する
- 機能の統計を送信する

- 送信済みおよび未送信のクラッシュレポートに関する情報を表示する
- 送信済みおよび未送信のクラッシュレポートに関する情報を出力する
- 送信のステータスを問い合わせる
- 使用中の SQL Anywhere のビルドに対応するソフトウェア更新のリリース状況を問い合わせる
- 致命的なエラー（アサーション/クラッシュ）を検出した場合の dbsupport の動作を設定する。たとえば、データベースサーバがエラーレポートを送信するたびに、指定したハンドラプログラムを実行するように、dbsupport を設定できます。この機能は、エラー処理プロセスにカスタムアクションを追加する場合に便利です。
- レポートの再送回数を設定する。たとえば、レポートの送信時に、30 秒以内に操作をリトライし、最大 10 回までリトライするように設定できます。この機能によって、サービスが一時的に使用不能となった場合の対処方法を組み込むことができます。

サポートユーティリティの設定は、診断ディレクトリの `dbsupport.ini` ファイルに格納されます。

操作固有のオプションは、`dbsupport.ini` file に保存されている動作を含め、デフォルトの動作に優先させる場合に役立ちます。

グラフィカルな管理ツールで、エラーレポートを送信するオプションを削除するよう設定することができます。

レポートを送信しないことを選択した場合、レポートの内容はハードディスクの診断ディレクトリに維持されます。このディレクトリはソフトウェアプログラムによって自動的に作成されるもので、レポートの保存と、サポートユーティリティ (dbsupport) の設定の格納にのみ使用されます。

#### 診断ディレクトリの場所

診断ディレクトリの場所はプラットフォームによって異なります。

データベースサーバは、次の優先順序に従って診断ディレクトリの有無をチェックします。

#### Windows

1. SADIAGDIR 環境変数で指定されたディレクトリ (設定している場合)。
2. デフォルトの診断ディレクトリ:
  - Windows 7 以降: `%ALLUSERSPROFILE%\SQL Anywhere17\diagnostics`。
3. 現在のディレクトリ。
4. テンポラリディレクトリ。SATMP、TMP、TMPDIR、TEMP の各環境変数を参照してください。

Windows では、診断ディレクトリは、コンピュータのすべてのユーザが書き込みできる場所に置かれます。コンピューターで複数のユーザアカウントを使用していて、エラーレポートをユーザごとに分けておく必要がある場合は、ユーザごとに SADIAGDIR 環境変数を設定して、その特定のユーザだけが読み込み可能および書き込み可能なディレクトリを指定します。

#### UNIX、Linux、OS X

1. SADIAGDIR 環境変数で指定されたディレクトリ (設定している場合)。
2. デフォルトの診断ディレクトリ: `$HOME/.sqlanywhere17/diagnostics`。
3. 現在のディレクトリ。
4. テンポラリディレクトリ。SATMP、TMP、TMPDIR、TEMP の各環境変数を参照してください。

#### i 注記

UNIX では、データベースサーバまたは Mobile Link サーバをデーモンとして実行する場合やユーザが `root/nobody` である場合、ユーザのホームディレクトリにクラッシュレポートを書き込むことは推奨されません。

#### エラーレポートファイル名

エラーレポートのファイル名は、次のように構成されます。

- アプリケーションを識別するプレフィクス

アプリケーションプレフィクス	アプリケーション
ISQL	Interactive SQL ユーティリティ
LSN	Listener ユーティリティ
MLC	Mobile Link クライアント
MLS	Mobile Link サーバ
ML_MON	Mobile Link プロファイラ
SA	パーソナルデータベースサーバまたはネットワークデータベースサーバ
SCJ	SQL Central
SR	SQL Remote

- ソフトウェアのバージョンを示す数値
- アンダースコアで連結された 2 つのフィールド (エラーレポートが作成されたときのタイムスタンプを示す)
- アプリケーションの識別子
- 拡張子 .dmp

たとえば、`SA17_20150620_133828_32116.dmp` は、SQL Anywhere バージョン 17 のデータベースサーバで、2015/06/20 の午後 1:38:28 に、プロセス 32116 から作成されたエラーレポートであることを示します。

## 例

1. この `dbssupport` のデフォルト動作を変更するには、`-cc` オプションを使用します。
  - 次のコマンドは、ユーザにメッセージを表示せず、自動的にレポートを送信するように `dbssupport` を設定します。

```
dbssupport -cc autosubmit
```

- 次のコマンドは、自動的にレポートを送信し、電子メールサーバ `emailserver.company.com` を使用して `myemail@company.com` 宛てに電子メールによる通知を送信するように `dbssupport` を設定します。

```
dbssupport -cc autosubmit -ce  
"myemail@company.com"; "emailserver.company.com:25"
```

- 次のコマンドは、レポートの自動送信を無効にします。

```
dbssupport -cc no
```

2. `-lc` オプションを使用してエラーレポートのリストを表示できます。次のコマンドは、送信されていないすべてのクラッシュレポートのリストを作成します。

```
dbssupport -lc
```

3. `-sa`、`-sc`、または `-sd` オプションを使用してエラーレポートを手動で送信できます。次のコマンドは、診断ディレクトリに保存されているすべてのレポートを送信します。

```
dbssupport -sa
```

4. -ac オプションと -af オプションをそれぞれ指定して、エラーレポートと共にコメントとファイルを送信できます。

```
dbsupport -sc SA17_20150901_113308_3360 -ac "The message.txt file provides more information about this error report." -af c:¥scenario.txt -af c:¥message.txt
```

## 関連情報

[パフォーマンスデータの定期送信 \[974 ページ\]](#)

[ソフトウェアの更新 \[1062 ページ\]](#)

[データベースサーバの診断情報のスナップショットの SAP サポートへの送信 \(プロファイラ\) \[1418 ページ\]](#)

[SADIAGDIR 環境変数 \[550 ページ\]](#)

## 1.7.4.28 トランザクションログユーティリティ (dblog)

データベースのトランザクションログを管理します。

### 構文

```
dblog [ options ] database-file
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-ekkey	強力に暗号化されているデータベースの暗号化キーをコマンドに直接指定します。データベースが強力に暗号化されている場合、データベースまたはトランザクションログを使用するには必ず暗号化キーを指定します。強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。強力に暗号化されたデータベースでは、正しいキーを指定しないとコマンドが失敗します。
-ep	暗号化キーの入力を求めるプロンプトを表示するよう指定します。このオプションを指定すると、暗号化キーを入力するためのウィンドウが表示されます。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。強力な暗号化が適用されたデータベースの場合、-ek または -ep のどちらかを指定します。両方同時には指定できません。強力に暗号化されたデータベースでは、正しいキーを指定しないとコマンドが失敗します。

オプション	説明
<code>-ftimeline</code>	予定表を <code>timeline</code> にリセットし、データベースがレプリケーションに参加できるようにします。このオプションは、SQL Remote 統合データベースを再ロードするときに使用します。予定表は、次の形式で指定する必要があります。" <b>GUIDtimestamp</b> " <code>timestamp</code> には、"YYYY-MM-DD HH:MM:SS.MMMM" の UTC 時間表示形式を使用します。これは dblog 出力と同じ形式です。
<code>-ir</code>	このオプションは、 <code>delete_old_logs</code> オプションのために保存されている SQL Remote ログオフセットをリセットし、必要のないトランザクションログを削除できるようにします。このオプションは、このデータベースで SQL Remote を使用しなくなったが、Mobile Link 同期は引き続き使用する場合に指定します。
<code>-is</code>	このオプションは、 <code>delete_old_logs</code> オプションのために保存されている Mobile Link ログオフセットをリセットし、必要のないトランザクションログを削除できるようにします。このオプションは、このデータベースで Mobile Link 同期を使用しなくなったが、SQL Remote は引き続き使用する場合に指定します。
<code>-mmirror-name</code>	新しいトランザクションログミラーのファイル名を設定します。データベースがトランザクションログミラーを現在使っていない場合、データベースはこの設定された名前を使って起動します。すでにトランザクションログミラーを使っている場合、データベースはトランザクションログミラーとして新しいファイル名を使うように変更します。
<code>-n</code>	トランザクションログとトランザクションログミラーの使用を停止します。トランザクションログを使用しないと、データベースはデータレプリケーションに参加できず、またはデータリカバリにトランザクションログを使えません。SQL Remote または <code>dbmsync</code> トランザクションオフセットが存在する場合は、対応する無視オプション (SQL Remote には <code>-ir</code> 、 <code>dbmsync</code> には <code>-is</code> ) も指定されていないかぎり、トランザクションログを削除することはできません。データベースで監査がオンになっており (監査を先にオフにしないかぎり)、 <code>audit_log</code> オプションでトランザクションログが必要と設定している場合は、トランザクションログの使用を停止することはできません。
<code>-Ofilename</code>	指定したファイルに、出力メッセージを追加します。
<code>-q</code>	クワイエットモードで実行します (メッセージを表示しません)。
<code>-r</code>	1つのトランザクションログファイルを維持し、トランザクションログミラーを使用して停止します。
<code>-tlog-name</code>	新しいトランザクションログのファイル名を設定します。データベースがトランザクションログを現在使っていない場合、データベースは設定されたファイル名を使って起動します。すでにトランザクションログを使っている場合、データベースはトランザクションログとして新しいファイル名を使うように変更します。
<code>-Xn</code>	トランザクションログの現在の相対オフセットを <code>n</code> にリセットし、データベースがレプリケーションに参加できるようにします。このオプションは、SQL Remote 統合データベースを再ロードするときに使用します。

オプション	説明
-Zn	トランザクションログの現在の開始オフセットを n にリセットし、データベースがレプリケーションに参加できるようにします。このオプションは、SQL Remote 統合データベースを再ロードするときに使用します。

## 権限

なし。

## 備考

dblog ユーティリティで、データベースに関連するトランザクションログまたはトランザクションログミラーの名前を表示、変更できます。データベースがトランザクションログやミラーを管理するのを停止したり、開始したりできます。

トランザクションログミラーはトランザクションログの重複コピーであり、データベースによって並列に管理されています。

データベースを初期化するときに、トランザクションログの名前を最初に設定します。トランザクションログユーティリティは、データベースファイル进行处理します。トランザクションログファイル名を変更するときは、データベース上でデータベースサーバを実行しないでください (実行するとエラーメッセージが表示されます)。

このユーティリティは、トランザクションログに関する次のような追加情報も表示します。

- バージョン番号
- トランザクションログファイルの名前
- トランザクションログミラーファイルの名前 (トランザクションログミラーがある場合)
- 現在の相対オフセット

次の方法で、トランザクションログユーティリティにアクセスできます。

- SQL Central の [ログファイル設定の変更ウィザード](#) を使用します。
- Interactive SQL から ALTER DATABASE `dbfile` ALTER LOG 文を使用します。
- コマンドプロンプトで、dblog コマンドを入力します。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 関連情報

[設定ファイル \[1066 ページ\]](#)

[タイムライン \[942 ページ\]](#)

[トランザクションログの場所の変更 \(SQL Central の場合\) \[281 ページ\]](#)

[トランザクションログの場所の変更 \(コマンドラインの場合\) \[283 ページ\]](#)

## 1.7.4.29 アンロードユーティリティ (dbunload)

指定したディレクトリにデータベースをアンロードします。

### 構文

```
dbunload [ options ] [ directory ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。

オプション	説明
<pre>-ac"keyword=value; ..."</pre>	<p>既存のデータベースに接続し、データをそのデータベースに直接再ロードすることによって、データベースのアンロードとその結果を既存のデータベースに再ロードする操作を一度に実行します。</p> <p>たとえば、dbinit ユーティリティを使用して新しいデータベースを作成した後、このオプションを使用して再ロードできます。この方法は、初期化オプションを変更する場合に便利です。</p> <p>次のコマンドは、data.db データベースのコピーを backup.db という既存のデータベースファイルにロードします。</p> <pre>dbunload -c "UID=DBA;PWD=passwd;DBF=data.db" -ac "UID=DBA;PWD=passwd;DBF=backup.db"</pre> <p>新しいデータベースが元のデータベースと同じファイル名を持つ必要がある場合、DBN 接続パラメータを使用して、各データベースにユニークで一時的な名前を指定します。次のコマンドは、同じファイル名 data.db を持つ新しいデータベースに古いデータベースのコピーをロードします。</p> <pre>dbunload -c "UID=DBA;PWD=passwd;DBN=old;DBF= ¥master¥data.db" -ac "UID=DBA;PWD=passwd;DBN=new;DBF= ¥backup¥data.db"</pre> <p>元のデータベースがバージョン 9 以前の SQL Anywhere で作成されており、新しいデータベースがまだ実行されていない場合は、-ac オプションでデータベースサーバの名前を指定する必要があります。例:</p> <pre>dbunload -c "UID=DBA;PWD=passwd;DBF=data.db" -ac "UID=DBA;PWD=passwd;DBF=newdata.db;Server=newserver"</pre> <p>このオプションを指定した場合、データの間コピーはディスク上に作成されないため、コマンドにアンロードディレクトリは指定しないでください。これにより、データのセキュリティが向上します。</p> <p>このオプションを指定した場合、パスワードは自動的にアンロードされません。-up オプションを指定する必要はありません。</p>

オプション	説明
<p><code>-ae</code></p>	<p>エラーが発生した場合でもアンロード操作を続行するように指定するには、データベースの再構築時に (例: <code>-ac</code>, <code>-an</code>, <code>-ar</code> の各オプション)、このオプションを使用します。<code>-ae</code> を指定すると、<code>dbunload</code> は <code>reload.sql</code> スクリプトファイル内の文を実行します。いずれかの文がエラーになると、それらの文はスキップされて、<code>unprocessed.sql</code> スクリプトファイルに記録されます。( <code>-q</code> オプションが指定されていない場合) エラーになった文はコマンドライン出力にも表示されます。エラーが発生したデータベースの再構築を試みた後、エラーの検証と修正を行って、データベースを最初から再構築するか (推奨)、<code>unprocessed.sql</code> スクリプトファイル内のエラーを修正して、未完了のデータベースに修正を適用してください。</p> <p>このオプションは、オンラインデータベースの再構築時 (例: <code>-ao</code> または <code>-aob</code> オプション) には使用しないでください。</p>
<p><code>-andatabase</code></p>	<p>このオプションを使用すると、データベースのアンロード、新規データベースの作成、データのロードを一度に実行できます。</p> <p>ソースデータベースを作成したときに指定されたオプションが、新しいデータベースの作成に使用されます。ただし、サポートされているその他の <code>dbunload</code> オプション (たとえば、ページサイズを変更するときは <code>-ap</code>、テーブル暗号化を有効にするときは <code>-et</code>) を指定して、必要に応じて初期化オプションを変更することができます。</p> <p>たとえば、次のコマンドは、<code>copy.db</code> という名前の新規データベースファイルを作成し、<code>data.db</code> のスキーマとデータをその中にコピーします。</p> <pre data-bbox="762 1317 1292 1400">dbunload -c "UID=DBA;PWD=passwd;DBF=data.db" -an copy.db</pre> <p>このオプションを使用した場合、データの間コピーはディスク上に作成されないため、コマンドにアンロードディレクトリは指定しません。このため、データのセキュリティが向上します。</p> <p>新しいデータベースが作成されると、DB 領域ファイル拡張子に <code>R</code> が追加され、元のデータベースの DB 領域と同じディレクトリに新しいデータベースの DB 領域が作成された場合の名前の競合を防ぎます。たとえば、アンロードされたデータベースの <code>library.db</code> ファイルに <code>library</code> という DB 領域がある場合、新しいデータベースの <code>library</code> DB 領域は <code>library.dbR</code> となります。</p> <p><code>-an</code> には、ファイル名としてデータベースサーバに対する相対パスを指定します。</p> <p>このオプションを指定した場合、パスワードは自動的にアンロードされます。<code>-up</code> オプションを指定する必要はありません。</p>

オプション	説明
<p><code>-ao</code><code>database-name</code></p>	<p>フルバックアップとインクリメンタルバックアップを実行し、運用データベースのログファイルを適用する、再構築されたデータベースを生成します。</p> <p><code>-ao</code> オプションを使用するには、次の条件を満たす必要があります。</p> <ul style="list-style-type: none"> <li>元のデータベースが SQL Anywhere バージョン 17 によって作成されていること。</li> <li>再構築されたデータベースのページサイズ、暗号化アルゴリズム、暗号化キーが元のデータベースと同じであること。</li> <li>中間ファイルが必要なため、再構築を実行するコンピュータに、元のデータベースのデータベース、DB 領域、ログファイルの合計の 2 倍のサイズを保持するための十分な空き領域があること。</li> <li>現在のデータベースで DB 領域が使用されている場合、その DB 領域ファイルがデータベースファイルと同じディレクトリに存在し、絶対パスを使用していないこと。</li> <li>データベースで高可用性機能を使用していないこと。</li> </ul> <p>次の条件を満たすことができない場合、<code>dbunload -aob</code> を使用したデータベースの再構築を検討してください。</p> <ul style="list-style-type: none"> <li>トランザクションが失われないように、<code>dbbackup -wa</code> を使用して最初のバックアップを取る必要があるため、運用サーバに未処理のトランザクションがない状態が一定時間なくてはなりません。</li> <li>運用サーバへの複数回のバックアップや、運用サーバでのトランザクションログの名前変更は許容される必要があります。</li> </ul> <p><code>-c</code> オプションと <code>-ao</code> オプションを同時に使用する場合、接続文字列は運用データベースのものとなり、再構築前の初期データベースのバックアップと、再構築に適用されるトランザクションログのインクリメンタルバックアップの取得で使用されます。</p> <p>データベースの暗号化設定は、オンラインでの再構築中に変更することはできません。元のデータベースの設定は自動的に保持されます。データベースが強力な暗号化を使用している場合、<code>-ek</code> または <code>-ep</code> <code>dbunload</code> オプションを使用して現在の暗号化キーを指定してください。</p> <p>運用データベースのトランザクションログのコピーがデータベースファイルと同じディレクトリにない場合、再構築されたデータベースを再起動することはできません。再構築されたデータベースのトランザクションログのパスは、元の運用データベースでの相対パスまたは絶対パスにかかわらず、データベースファイルと同じディレクトリです。</p>

オプション	説明
-aob	<p>運用データベースのトランザクションログを適用できる再構築データベースを作成します。</p> <p>運用サーバにユーザによる未処理のトランザクションがない状態が一定時間ない場合、または運用サーバへの複数回のバックアップや、運用サーバでのトランザクションログの名前変更が許容されていない場合、ログの名前変更中に処理中のトランザクションがないことを確認し、transaction log rename を使用してフルバックアップを実行します。これにより、dbunload -aob を使用して、取得したバックアップからオンラインで再構築ができるようになります。接続文字列は運用データベースではなく、バックアップデータベースへのものです。バックアップ後は運用データベースのトランザクションログを適用できます。再構築されたデータベースのトランザクションログのパスは、元の運用データベースでの相対パスまたは絶対パスにかかわらず、データベースファイルと同じディレクトリです。</p> <p>データベースの暗号化設定は、オンラインでの再構築中に変更することはできません。元のデータベースの設定は自動的に保持されます。データベースが強力的な暗号化を使用している場合、-ek または -ep dbunload オプションを使用して現在の暗号化キーを指定してください。</p> <p><b>i 注記</b></p> <p>-ao オプションを使用すると、最初のフルバックアップ、再構築、トランザクションログ適用の 1 回以上の反復が含まれるため、可能であればいつでもデータベースを構築できます。-aob オプションと比較して、-ao オプションを使用する場合、再構築の前と後の両方で必要な手順が少なくてすみます。</p>
-aotseconds	<p>運用データベースと再構築されたデータベースのインクリメンタルバックアップの最大時間を指定します。-ao も一緒に指定しないと、このオプションは無視されます。インクリメンタルバックアップと適用時間が指定値よりも大きい場合、別のインクリメンタルバックアップと適用が実行されます。インクリメンタルバックアップと適用を任意のタイミングで停止するには、Q キーを押すと、バックアップの適用後に dbunload が停止します。</p> <p><b>i 注記</b></p> <p>-aot 0 の指定は無効です。seconds を 1 分未満の値に指定すると、無限ループが発生する場合があります (Q キーを押すまで)、また、数百ものトランザクションログの名前変更が運用データベースで発生する場合があります。</p>

オプション	説明
<p><code>-apsize [k]</code></p>	<p>新しいデータベースのページサイズを設定します。たとえば、次のコマンドは、old.db というデータベースを新しいデータベース new.db にアンロードし、ページサイズを 8K に変更します。</p> <pre data-bbox="861 470 1468 571">dbunload -c "UID=DBA;PWD=passwd;DBN=old;DBF=old.db " -an new.db -ap 8k</pre> <p>-an または -ar も一緒に指定しないと、このオプションは無視されます。データベースのページサイズには、2048 (非推奨)、4096、8192、16384、32768 バイトのいずれかを指定できます。デフォルトは元のデータベースのページサイズです。<i>k</i> を使用して、キロバイトの単位を指定します (-ap 4k など)。このオプションとともに -an または -ar のいずれかを指定する必要があります。データベースサーバですでにデータベースが実行中の場合、サーバのページサイズ (-gp オプションで設定) は新規ページサイズを処理するのに十分な容量でなければなりません。</p>

オプション	説明
<p><code>-ar [directory]</code></p>	<p>古いデータベースと同じ設定を持つ新しいデータベースを作成し、新しいデータベースを再ロードして、古いデータベースと置き換えます。ただし、サポートされているその他の dbunload オプション (たとえば、ページサイズを変更するときは <code>-ap</code>、テーブル暗号化を有効にするときは <code>-et</code>) を指定して、必要に応じて初期化オプションを変更することができます。</p> <p>このオプションを使用する場合は、データベースへの他の接続がなく、データベース接続が、ネットワークではなくローカルである必要があります。</p> <p>オプションの <code>directory</code> を指定すると、レプリケーションを行うためにトランザクションログのオフセットがリセットされ、古いデータベースのトランザクションログが指定のディレクトリに移動されます。指定されたディレクトリは、Message Agent が使用する古いトランザクションログを保持している必要があります。トランザクションログ管理は、データベースがレプリケーションで使用されるときだけ実行されます。SQL Remote パブリッシャがない場合、古いトランザクションログは不要なので、指定するディレクトリにはコピーされず、削除されます。</p> <p>新しいデータベースが作成されると、DB 領域ファイル名に R が追加され、元のデータベースの DB 領域と同じディレクトリに新しいデータベースの DB 領域が作成された場合の名前の競合を防ぎます。たとえば、アンロードされたデータベースの <code>library.db</code> ファイルに <code>library</code> という DB 領域がある場合、新しいデータベースの <code>library</code> DB 領域は <code>library.dbR</code> となります。</p> <p>暗号化されたデータベースを再構築する場合は、新しいデータベースで元のデータベースと同じ暗号化キーを使用する必要があります。</p> <p><code>-ar</code> オプションを使用すると、データベースのトランケーションポイントが 0 にリセットされます。</p> <p>このオプションを指定した場合、パスワードは自動的にアンロードされます。<code>-up</code> オプションを指定する必要はありません。</p>
<p><code>-bp</code></p>	<p>BEGIN PARALLEL WORK 文の内部の CREATE INDEX 文と LOAD TABLE 文をまとめて <code>reload.sql</code> ファイルに挿入します。BEGIN PARALLEL WORK 文の内部の文は並列実行されません。</p>

オプション	説明
-c"keyword=value;..."	<p>ソースデータベースの接続パラメータを指定します。</p> <p>たとえば、次の文は、パスワード sql を指定してユーザ ID DBA として接続し、サンプルデータベースをアンロードします。データは c:¥unload ディレクトリにアンロードされます。</p> <pre>dbunload -c "DBF=%SQLANY17% ¥demo.db;UID=DBA;PWD=sql" c:¥unload</pre>
-cm { sql   dbinit }	<p>アンロードされているデータベースと同じデータベースを作成する CREATE DATABASE 文または dbinit コマンドを、コマンドプロンプトに表示します。-an も一緒に指定した場合、表示されるコマンドは、新しいデータベースを作成するコマンドです。</p> <p><b>sql</b></p> <p>reload.sql ファイルに書き込まれる CREATE DATABASE 文を表示します。</p> <p><b>dbinit</b></p> <p>dbinit ユーティリティコマンドを表示します。</p> <p>CREATE DATABASE 文または dbinit コマンドを表示する場合、DBA ユーザ ID、DBA パスワード、暗号化キー (存在する場合) に対してアスタリスクが表示されます。</p> <p>バージョン 10 以前のデータベースサーバで作成したデータベースをアンロードする場合は、データベースを作成するコマンドまたは文は表示されません。</p>
-cp	<p>実行される UNLOAD TABLE 文に COMPRESSED キーワードを追加して、テーブルデータ出力ファイルを圧縮します。このオプションは、-an または -ar が指定されている場合は機能しません。</p>
-d	<p>データベース定義文 (CREATE TABLE、CREATE INDEX など) は生成されません。reload.sql にはデータを再ロードするための文のみが記述されます。</p>
-dc	<p>データベース内のすべての計算カラムを再計算します。デフォルトでは、計算カラムは再計算されません。-dc オプションを指定すると、reload.sql スクリプトに計算カラムの再計算に対応するセクションが追加されます。追加される文は次のような形式です。</p> <pre>ALTER TABLE "owner"."table-name" ALTER "computed-column" SET COMPUTE (compute-expression);</pre> <p>テーブルに CURRENT DATE などのコンテキスト依存の計算カラムがある場合は、-dc オプションを使用するのではなく、ALTER TABLE 文を使用して計算カラムの値を再計算してください。</p>

オプション	説明
<code>-dt</code> directory	再構築で使用するテンポラリディレクトリを指定します。ディレクトリが存在し、運用データベースとトランザクションログファイルのための十分な空き領域がディレクトリにある必要があります。テンポラリディレクトリと再構築されたデータベースが同じディスクパーティションにある場合、新しく作成されるデータベースのための十分な空き領域もある必要があります。このオプションは、 <code>-ao</code> オプションが指定されていない場合は無視されます。このオプションを指定しない場合は、SQL Anywhere テンポラリファイルのデフォルトロケーションが使用されます。
<code>-etable, ...</code>	<p>指定されたテーブルを <code>reload.sql</code> ファイルから除外します。大文字と小文字を区別するデータベースであっても、テーブル名では常に大文字と小文字が区別されません。</p> <p><code>-e</code> オプションを使用して作成した <code>reload.sql</code> ファイルにはすべてのデータベーステーブルが含まれるわけではないので、このファイルを使用してデータベースを再構築しないでください。外部キーによって参照されているテーブルがある場合、そのテーブルの内容がなければ、データベースを再構築できません。</p> <p><code>-e</code> オプションを使用する場合は、必ず <code>-d</code> オプションも使用し、<code>-e</code> で指定したテーブルを除くすべてのテーブルのデータをアンロードしてください。</p>

オプション	説明
<p><code>-eaalgorithm</code></p>	<p>データベース暗号化またはテーブル暗号化 (-et) で使用するエンコードアルゴリズムを指定します。単純な難読化の場合は、(-ek や -ep を指定するのではなく) <code>-ea simple</code> を指定します。単純な難読化は、データベースファイルが不用意に直接アクセスされた場合にデータが表示されないようにすることだけを目的としています。これにより第三者は、ディスクユーティリティを使用してファイルを表示し、データベースのデータを解読することが困難になります。</p> <p>セキュリティを強化するには、128ビットの場合は AES、256ビットの場合は AES256 の強力な暗号化を指定します。FIPS 認定の暗号化を使用するには、128ビットの場合は AES_FIPS、256ビットの場合は AES256_FIPS をそれぞれ指定してください。強力な暗号化を使用するためには、-ek または -ep オプションも指定する必要があります。</p> <p>暗号化されていないデータベースを作成するには、<code>-ea none</code> を指定するか、-ea オプションを指定しません (-e、-et、-ep、-ek オプションも指定しません)。</p> <p>-ea オプションを指定しない場合、デフォルトの動作は次のようになります。</p> <ul style="list-style-type: none"> <li>• -ea NONE (-ek、-ep、-et が指定されない場合)</li> <li>• -ea AES (-ek または -ep が指定される場合) (-et の指定とは無関係)</li> <li>• -ea SIMPLE (-ek または -ep を指定せずに -et を指定する場合)</li> </ul> <p>アルゴリズム名の太文字と小文字は区別されません。</p>
<p><code>-ekkey</code></p>	<p>(-an オプションを使用して) データベースのアンロードと再ロードを実行する場合に、作成する新しいデータベースに対する暗号化キーを <code>dbunload</code> コマンドに指定します。強力的に暗号化されたデータベースを作成する場合、データベースやトランザクションログを使用するには必ず暗号化キーを指定します。データベースの暗号化に使用されるアルゴリズムは、-ea オプションで指定したアルゴリズムです。-ek オプションを指定して、-ea オプションを指定しないと、AES アルゴリズムが使用されます。</p> <p>キーは保護してください。キーのコピーを安全な場所に保管してください。キーを紛失すると、データベースにまったくアクセスできなくなり、リカバリも不可能になります。</p>

オプション	説明
-ep	<p>(-an オプションを使用して) データベースのアンロードと再ロードを実行する場合に、作成する新しいデータベースに対する暗号化キーの指定を求めるプロンプトを表示します。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られます。-an を指定せずに -ep を指定した場合、-ep オプションは無視されます。-ep と -an を指定する場合は、暗号化キーが正確に入力されたことを確認するために、2 回入力してください。キーが一致しない場合は、アンロードは失敗します。</p>
-er	<p>アンロード中に暗号化されたテーブルの暗号化を解除します。</p> <p>テーブル暗号化が有効であるデータベースを再構築するとき、-er または -et を指定して、新しいデータベースのテーブルで暗号化を有効にするかどうかを指定する必要があります。このオプションを指定しないと、データを新しいデータベースにロードしようとしたときにエラーが発生します。</p> <p>次のコマンドは、テーブルの暗号化を解除してデータベース (data.db) をアンロードし、暗号化が有効になっていない新しいデータベース (copy.db) に再ロードします。</p> <pre data-bbox="762 1048 1335 1126">dbunload -an copy.db -er -c "UID=DBA;PWD=passwd;DBF=data.db;DBKEY= 29bN8cj1z"</pre>
-et	<p>新しいデータベースのテーブル暗号化を有効にします (-an または -ar も一緒に指定する必要があります)。-ea オプションを指定せずに -et オプションを指定すると、AES アルゴリズムが使用されます。-et オプションを指定する場合、-ep または -ek も指定しないと操作は失敗します。新しいデータベースのテーブル暗号化設定を変更して、アンロードしているデータベースのテーブル暗号化設定とは異なるものに設定できます。</p> <p>テーブル暗号化が有効であるデータベースを再構築するとき、-er または -et を指定して、新しいデータベースのテーブルで暗号化を有効にするかどうかを指定する必要があります。このオプションを指定しないと、データを新しいデータベースにロードしようとしたときにエラーが発生します。</p> <p>次の例は、テーブルが暗号化されているデータベース (data.db) を、テーブル暗号化が有効になっている新しいデータベース (copy.db) にアンロードし、キーを 34jh として AES_FIPS アルゴリズムを使用します。</p> <pre data-bbox="762 1809 1335 1888">dbunload -an copy.db -et -ea AES_FIPS -ek 34jh -c "UID=DBA;PWD=passwd;DBF=data.db"</pre>

オプション	説明
-g	<p>再ロードの後にマテリアライズドビューとインデックスをリフレッシュするかどうかを制御します。</p> <p><b>マテリアライズドビュー</b></p> <p>MANUAL REFRESH と定義されているマテリアライズドビューは、デフォルトでは再ロード後に初期化されません。このようなマテリアライズドビューを再ロードプロセス中に初期化するには、-g オプションを指定してください。-g を指定すると、データベースサーバによって sa_refresh_materialized_views システムプロシージャが呼び出されます。</p> <p>-g オプションを指定するかどうかを決定するときには、すべてのマテリアライズドビューを初期化すると再ロードプロセスの実行時間が大幅に長くなる可能性があることを考慮に入れてください。しかし、-g オプションを指定しない場合、初期化されていないマテリアライズドビューを最初に使用しようとしたクエリはビューの初期化が完了するまで待つことになり、予想外の遅延が生じることがあります。-g オプションを指定しない場合は、再ロードの完了後に、マテリアライズドビューを手動で初期化することもできます。</p> <p><b>テキストインデックス</b></p> <p>MANUAL REFRESH と定義されているテキストインデックスは、デフォルトでは再ロード後に初期化されません。このようなテキストインデックスを再ロードプロセス中に初期化するには、-g オプションを指定してください。-g を指定すると、データベースサーバによって sa_refresh_text_indexes システムプロシージャが呼び出されます。</p>
-ii	<p>UNLOAD 文を使用してデータベースからデータを抽出し、reload.sql ファイル内の LOAD 文を使用してデータベースにデータを再配置します。これがデフォルトです。</p>
-ix	<p>UNLOAD 文を使用してデータベースからデータを抽出し、reload.sql ファイル内の Interactive SQL の INPUT 文を使用してデータベースにデータを再入力します。</p>

オプション	説明
<code>-k</code>	sa_diagnostic_auxiliary_catalog テーブルにデータを配置します。このテーブルは、テーブル、ユーザ、プロシージャなどのデータベースオブジェクト ID をソースデータベースからトレーシングデータベースにマッピングします。このとき、すべてのヒストグラムがアンロード/再ロードされます。このオプションは、診断トレーシング情報を受け取るトレーシングデータベースを作成する場合に使用します。さらに、このオプションによって、監査が運用データベースで有効になっている場合であっても、トレーシングデータベースでは監査を確実に無効にします。sa_diagnostic_auxiliary_catalog テーブルによって、サーバはトレーシングデータを収集したときの状態を (インデックスコンサルタントの目的などで) シミュレートすることができます。このオプションが特に役立つのは、-n オプションと一緒に指定した場合です。
<code>-kd</code>	データベースを 1 つの DB 領域ファイルに再ロードします。このオプションは、トレーシングデータベースを作成しているコンピュータのディレクトリ構造が運用データベースと同じでない場合に、トレーシングデータベースを作成するのに便利です。
<code>-kdi<math>n</math>umber</code>	暗号化キーのハッシュ回数を指定します。1 から 1000 の間の整数を指定してください。デフォルト値は 2 で、ハッシュが 2000 回適用されます。回数が多いほど、セキュリティは高くなります。データベースを速度の遅いコンピュータ上で稼働させている場合、繰り返し回数が多すぎるとデータベースの起動が遅くなる可能性があります (いったんデータベースが稼働を始めれば、パフォーマンスへの影響はありません)。このオプションは、-ek または -ep オプションと一緒に、あるいは -ar または -an オプションと一緒に指定する必要があります。
<code>-l</code>	データベースを再構築しても SYSTABCOL.max_identity の現在の値を維持します。  デフォルトでは、オートインクリメントカラムがあるテーブルを含むデータベースを再構築すると、データベースサーバでは、テーブルの現在の内容に基づいて、各オートインクリメントカラムに使用できる次の値が計算されます。通常はこの処理で十分です。ただし、値の範囲の最後からローが削除されている場合は、値が再利用される可能性があり、この処理は適切ではありません。  -l オプションを指定すると、オートインクリメント値が含まれるテーブルごとに、sa_reset_identity システムプロシージャへの呼び出しが生成される reload.sql スクリプトに追加され、SYSTABCOL.max_identity の現在の値が維持されます。
<code>-m</code>	レプリケーションに参加しているデータベースでユーザ ID を保存しません。

オプション	説明
<code>-n</code>	データベースのデータをアンロードしません。 <code>reload.sql</code> には、データベースの構造だけを構築する SQL 文が記述されています。 <code>reload.sql</code> ファイルに LOAD TABLE 文または INPUT 文を追加する場合は、 <code>-n</code> ではなく <code>-nl</code> を使用してください。
<code>-nl</code>	構造体をアンロードします ( <code>-n</code> オプションと同じ動作)。ただし、生成される <code>reload.sql</code> ファイルには、各テーブルに対する LOAD TABLE 文または INPUT 文も含まれます。このオプションを使用した場合、ユーザデータはアンロードされません。 <code>-nl</code> を指定するときには、LOAD/INPUT 文を生成できるようにデータディレクトリも指定する必要があります。ただし、このディレクトリにファイルは書き込まれません。このオプションを指定すると、データをアンロードしない再ロードスクリプトを生成できます。データをアンロードするには、 <code>-d</code> を指定します。データベースにデータのアンロードが不要なテーブルがある場合は、 <code>dbunload -d -e table-name</code> と指定することによって、そのテーブルのデータをアンロードの対象から除外することができます。

オプション	説明
<p><code>-no</code></p>	<p>データベースオブジェクトを名前順にアンロードします。デフォルトでは、<code>dbunload</code> は作成された順にオブジェクトを生成します。作成順序が異なる同一のオブジェクトが複数のデータベース内にある場合は、<code>-no</code> オプションを指定すると、データベーススキーマを比較する際に役立ちます。</p> <p><code>-no</code> を指定すると、オブジェクトの定義は <code>reload.sql</code> ファイル内でオブジェクトタイプのアルファベット順で分類されます。</p> <ul style="list-style-type: none"> <li>• ユーザ</li> <li>• グループメンバーシップ</li> <li>• テーブル</li> <li>• インデックスと外部キー</li> <li>• ビュー</li> <li>• プロシージャ</li> <li>• ファンクション</li> <li>• トリガ</li> <li>• イベント</li> <li>• Web サービス</li> </ul> <p>オブジェクトの定義は、所有者と名前の順で出力されます。外部キー、ロール名、トリガ名などの 3 つ目の要素が順序に含まれることがあります。</p> <p><code>-no</code> オプションは、<code>-n</code>、<code>-nl</code>、<code>-ar</code>、<code>-an</code>、<code>-ac</code> オプションと一緒に使用できません。同じバージョンのデータベースサーバを使用して作成されたデータベースの再ロードスクリプトを比較するときは、比較を簡単にするために、<code>-no</code> オプションを使用してください。これは、オブジェクトの定義間にわずかな違いがあるからです。</p> <p><b>警告</b></p> <p>オブジェクトの作成順序は重要な意味を持つため、生成されたファイルを使用して新しいデータベースを作成しないでください。たとえば、プロシージャ <code>p2</code> がプロシージャ <code>p1</code> を呼び出し、<code>p1</code> が結果セットを返す場合、<code>p1</code> を <code>p2</code> の前に定義することが重要です。<code>-no</code> オプションを指定して生成された <code>reload.sql</code> スクリプトファイルを実行しようとすると、エラーが発生します。</p> <p>このオプションを <code>-up</code> オプションと一緒に指定すると、エラーが返されます。</p>
<p><code>-Ofilename</code></p>	<p>指定したファイルに、出力メッセージを追加します。このファイルのロケーションは、<code>dbunload</code> を実行するディレクトリに相対します。</p>
<p><code>-pchar</code></p>	<p>外部アンロード (<code>dbunload -x</code> オプション) で使用するデフォルトのエスケープ文字 (<code>¥</code>) を別の文字に変更できます。このオプションは、このユーティリティをコマンドプロンプトで実行する場合のみ使用できます。</p>

オプション	説明
-q	クワイエットモードで実行します (メッセージを表示しません)。このオプションは、このユーティリティをコマンドプロンプトで実行する場合のみ使用できます。-q を指定する場合、-y も指定する必要があります。-y を指定しなければ、reload.sql がすでに存在する場合、アンロードは失敗します。
-qr	テーブルのロードやインデックスの作成を行うときに、進行メッセージの作成と表示をできないようにします。
-rreload-file	生成される再ロード SQL スクリプトファイルの名前とディレクトリを変更します。デフォルトは現在のディレクトリの reload.sql です。ディレクトリは、サーバではなく、クライアントアプリケーションの現在のディレクトリに相対します。
-ss	再ロード SQL スクリプトファイルにカラム統計を出力しないようにします。
-ttable,...	<p>アンロードするテーブルのリストを指定します。デフォルトでは、すべてのテーブルがアンロードされます。-n オプションと一緒に使うと、テーブル定義のセットだけをアンロードできます。大文字と小文字を区別するデータベースであっても、テーブル名では常に大文字と小文字が区別されません。</p> <p>-t オプションを使用して作成した reload.sql ファイルにはすべてのデータベーステーブルが含まれるわけではないので、このファイルを使用してデータベースを再構築しないでください。外部キーによって参照されているテーブルがある場合、そのテーブルの内容がなければ、データベースを再構築できません。</p> <p>-t オプションを使用する場合は、必ず -d オプションも使用し、-t で指定したテーブルのデータをアンロードしてください。</p>
-u	データの順序付けにインデックスを使用できないようにします。矛盾したインデックスを持つデータベースをアンロードする場合に、このオプションを使用すると、矛盾したインデックスがデータの順序付けに使われることはありません。通常、各テーブルでは、プライマリキーまたはクラスタドインデックスが定義されている場合、それに基づいてデータが順序付けられます。
-up	<p>ユーザパスワードをアンロードします (ハッシュ値)。再ロードを伴うアンロード (-ac、-an、-ar オプション) を実行している場合、このオプションを指定する必要はありません。</p> <p>このオプションを -no オプションと一緒に指定すると、エラーが返されます。</p>
-v	アンロードしているテーブルの名前と、アンロードが完了したローの数を表示します。このオプションは、dbunload をコマンドプロンプトで実行する場合にのみ使用できます。
-xi	dbunload クライアントへデータをアンロードし、生成された再ロードスクリプトファイル reload.sql 内の LOAD 文を使用することで外部アンロードを実行し、データベースにデータを再配置します。

オプション	説明
-xx	dbunload クライアントヘデータをアンロードし、生成された再ロードスクリプトファイル reload.sql 内の Interactive SQL の INPUT 文を使用することによって外部アンロードを実行し、データベースにデータを再配置します。
-y	確認メッセージを表示することなく、既存のスクリプトファイルを置き換えます。-q を指定する場合、-y も指定する必要があります。-y を指定しなければ、dbunload が既存のスクリプトファイルを検出した場合、アンロードは失敗します。
directory	アンロードされたデータを格納するディレクトリを指定します。reload.sql スクリプトファイルは、常にユーザの現在のフォルダと相対的になります。

## 権限

再ロードを伴う新しいデータベースへのアンロードの場合 (-an, -ao, -aob) は、SELECT ANY TABLE、SERVER OPERATOR および ACCESS USER PASSWORD システム権限が必要です。再ロードを伴う既存データベースへのアンロードの場合 (-ac) は、SERVER OPERATOR システム権限は必要ありません。

再ロードを伴わないアンロードの場合は、SELECT ANY TABLE システム権限が必要です。

パスワードをアンロードする (-up オプション) には、ACCESS USER PASSWORD システム権限が必要です。

## 備考

再ロードを伴うアンロード (-ac, -an, -ar オプション) を実行している場合にのみ、デフォルトでパスワードがアンロードされます。それ以外の場合は、-up オプションを指定してパスワードをアンロードしてください。

dbunload ユーティリティを実行するときは、アクティブなトランザクション (DDL を含む) がデータベースで発生していないことを確認してください。

バージョン 10.0.0 以降のデータベースで dbunload を使用する場合は、使用する dbunload のバージョンが、データベースへのアクセスに使用するデータベースサーバのバージョンと一致する必要があります。dbunload のバージョンがデータベースサーバのバージョンより古いまたは新しい場合は、エラーがレポートされます。

アンロードユーティリティを使ってデータベースをアンロードし、指定したディレクトリの中にデータファイルのセットを入れることができます。アンロードユーティリティは、データベースを再構築するために、Interactive SQL スクリプトファイルを作成します。また、各テーブルのすべてのデータを、カンマ区切りの形式で、指定したディレクトリ内のファイルにアンロードします。バイナリデータはエスケープシーケンスを使って正しく再現できます。

内部アンロード/再ロードを実行すると、UPDATE ISYSUSER 文の発行により、各ユーザの現在のステータスに関する情報がアンロードされます。外部アンロード/再ロードでは、この情報は含まれず、すべてのユーザのステータスがリセットされます。

データベースをアンロードしてから再ロードして再構築すると、再構築されたデータベースが元のデータベースより小さくなる場合があります。データベースのサイズが縮小されるのは、インデックス変更が原因と考えられます。問題やデータが失われたことを示すものではありません。

アンロードユーティリティ (dbunload) を使用してバージョン 11 以前のデータベースを再構築すると、元のデータベースでチェックサムがオフになっていても、新しいデータベースではデフォルトでグローバルチェックサムが有効になります。おすすめできる方法ではありませんが、ALTER DATABASE 文を使用して、データベースのグローバルチェックサムを無効にすることができます。

### **i** 注記

リカバリが必要なバージョン 9 以前のデータベースは、バージョン 10 以降のアンロードユーティリティ (dbunload) を使用して再ロードできません。このようなデータベースはバージョン 9 以前の dbunload を使用して再ロードしてください。

また、アンロードユーティリティを使って、既存のデータベースから直接新しいデータベースを作成できます。これにより、通常のディスクファイルに書き込まれたデータベースの内容に関するセキュリティ問題が生じる可能性を回避できます。

レプリケーションに参加しているリモートデータベースに対するデータベース抽出 (アンロード) の場合、特殊な考慮事項がいくつかあります。

-gl データベースサーバオプションは、データベースからデータをアンロードするときに必要なパーミッションを制御します。

ユーザ ID dbo は、データベース内のシステムオブジェクトを所有します。これにはビューやストアドプロシージャも含まれません。

アンロードユーティリティでは、データベース作成時に dbo ユーザ ID 用に作成されたオブジェクトをアンロードしません。データベースをアンロードすると、システムプロシージャの再定義など、これらのオブジェクトに加えられた変更は失われます。データベースの初期化以降に dbo ユーザ ID によって作成されたオブジェクトは、アンロードユーティリティでアンロードされ、これらのオブジェクトは保存されます。

データベースをアンロードした場合、システムオブジェクトに対するパーミッションの変更はアンロードされません。必要に応じて、新しいデータベースでパーミッションの付与または取り消しを行う必要があります。

### **➔** ヒント

データベースを再構築する前に、データを含めないでデータベースを再ロードし、次のようなコマンドを実行して再ロードプロセスを検証してください。

```
dbunload -n -an new.db -c "UID=your-user-id;PWD=your-password;DBF=original-database-file"
```

元のデータベースに問題が見つかった場合は、修正してから再構築してください。

デフォルトモードの場合、または -ii か -ix を使用する場合、データを格納するために dbunload が使用するディレクトリは、ユーザの現在のディレクトリではなく、データベースサーバとの相対ディレクトリになります。

-xi または -xx を使用する場合、ディレクトリはユーザの現在のフォルダとの相対ディレクトリになります。

テーブルのリストがない場合、すべてのデータベースをアンロードします。テーブルのリストがある場合、リストにあるテーブルだけをアンロードします。

アンロードされたデータには、reload.sql ファイルで生成された LOAD TABLE 文のカラムリストが含まれています。カラムリストのアンロードにより、テーブル内のカラムを並べ替えることができます。テーブルは削除または再作成することができ、その後 reload.sql を使用して再移植できます。

dbunload が生成する LOAD TABLE 文は、検査制約と計算カラムを無効にします。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

データベースの再構築後にデータベースのマテリアライズドビューを再表示してください。

トレーシング情報は、データベースのアンロードまたは再ロード操作の一部としてアンロードされません。トレーシング情報を別のデータベースに転送する場合は、sa\_diagnostic\_\* テーブルの内容をコピーして、手動で転送してください。ただし、この方法はおすすめしません。

## 内部アンロードと外部アンロード、内部再ロードと外部再ロード

次のオプションによって、内部と外部のアンロードと再ロードの組み合わせが提供されます: -ii、-ix、-xi、-xx。内部文 (UNLOAD/LOAD) を使用すると、外部コマンド (Interactive SQL の INPUT と OUTPUT 文) に比べ、パフォーマンスが大幅に向上します。ただし、内部文はサーバから実行されるため、ファイルとディレクトリは、データベースサーバに対する相対パスを使用します。外部コマンドを使用すると、ユーザの現在のディレクトリに対する相対パスを使用することになります。

UNLOAD 文を使用して、サーバに相対してアンロードするか、クライアントに相対してアンロードするかを指定できます。

データベースのアンロード、再ロード、または再構築を行うために外部アンロードや再ロードを行う場合、データベースの文字セットと dbunload を実行しているホストシステムの文字セットとの間に互換性がないと、文字セットを変換することでデータが破損する可能性があります。これは、データベースの文字セットとホストシステムの文字セットの間で変換が行われることが原因です。

この問題を回避するため、データベース用の接続文字列にデータベースの文字セットを指定します (-c オプションと -ac オプションを使用)。たとえば、データベースの文字セットが UTF-8 の場合は、接続文字列に "charset=utf-8" を含めます。

```
dbunload -c UID=user-ID;PWD=password;  
CHARSET=utf-8;DBF=filename -ac UID=user-ID;  
PWD=password;CHARSET=utf-8;Host=host-name -xx
```

外部アンロードを行うと、reload.sql の先頭に、コメントアウトされた CREATE DATABASE 文が含まれます。この文を使用して、アンロードしているデータベースと同じデータベースを作成できます。

アンロードしたデータベースがバージョン 9 以前の SQL Anywhere で作成され、カスタム照合があった場合は、COLLATION 句は次のようになります。

```
COLLATION collation-label DEFINITION collation-definition
```

ここで collation-definition はカスタム照合を指定する文字列です。

カスタム照合を保持する唯一の方法は、データベースを 1 ステップ (内部アンロード) で再構築することです。データベースをアンロードしてから、作成したデータベースにスキーマとデータをロードする場合は、提供されるいずれかの照合を使用する必要があります。

アンロードしたデータベースが強力な暗号化を使用して作成されていた場合は、CREATE DATABASE 文の KEY 句の値が疑問符 3 つ (???) として表示されます。

## アンロードの失敗

テーブルデータの再ロードとテーブル上のインデックスの再構築が完了した後で、`-ar` または `-an` を使用してデータベースの内部再構築を行っているときに障害が発生した場合、`dbunload` は現在のディレクトリに `unprocessed.sql` というファイルを作成します。このファイルには、障害が発生したために実行されなかった文が記録され、障害の原因となった文もコメントとして示されています。次に、`unprocessed.sql` ファイルの例を示します。

```
-- The database reload failed with the following error:
-- ***** SQL error: the-SQL-error
-- This script contains the statements that were not executed as a
-- result of the failure. The statement that caused the failure is
-- commented out below. To complete the reload, correct the failing
-- statement, remove the surrounding comments and execute this script.
/*
the failing statement
go
*/
setuser "DBA"
go
... the remainder of the statements to be processed
```

このファイルがあると、障害の原因となった文を修正、削除、または変更できます。`unprocessed.sql` ファイルは、すべてのテーブルデータと参照整合性制約が再ロードされた後に作成されます。Interactive SQL を使用して新しいデータベースに接続し、更新された `unprocessed.sql` ファイルを実行できます。この方法で、再構築を最初からやり直さずにデータベースの再構築を完了でき、場合によっては時間を大幅に節約できます。

`unprocessed.sql` ファイルを生成するとき、`dbunload` は停止して失敗を示すエラーコードを返し、再構築が失敗したことを他のツールやスクリプトに通知します。

`-ar` を使用したデータベースの内部再構築で障害が発生すると、新しいデータベースファイルとトランザクションログファイルに `.dbr` と `.logr` というファイル拡張子がそれぞれ付けられます。次の手順を使用して、`unprocessed.sql` ファイルを適用し、再ロードを手動で完了します。

1. 新しいデータベースを起動します。
2. 更新済みの `unprocessed.sql` ファイルを適用します。
3. データベースを停止します。
4. 新しいディレクトリに `original-name.db` と `original-name.log` を移動します。
5. `original-name.dbr` と `original-name.logr` ファイルを `original-name.db` と `original-name.log` に名前をそれぞれ変更します。
6. 次のコマンドを実行します。

```
dblog -t original-name.log original-name.db
```

## 暗号化されたデータベース

テーブル暗号化が有効であるデータベースを再構築するとき、`-er` または `-et` を指定して、新しいデータベースのテーブルで暗号化を有効にするかどうかを指定する必要があります。このオプションを指定しないと、データを新しいデータベースにロードしようとしたときにエラーが発生します。

強力な暗号化が適用されたデータベースをアンロードするには、暗号化キーを指定する必要があります。DatabaseKey (DBKEY) 接続パラメータを使用して、コマンドに暗号化キーを指定できます。または、標準の表示で入力するのではなく暗号化キーを入力するよう求める場合は、次のように `-ep` サーバオプションを使用できます。

```
dbunload -c "DBF=myenc.db;START=dbeng17 -ep"
```

`-an` オプションを使用してデータベースをアンロードし、新しいデータベースに再ロードするときに、`-ek` または `-ep` オプションを使用して新しいデータベースに対して暗号化キーを設定する場合は、次の点を考慮してください。

- 元のデータベースが強力に暗号化されている場合は、`-ek` または `-ep` オプションではなく、`-c` オプションで DatabaseKey (DBKEY) 接続パラメータを使用して、元のデータベースに対するキーを指定する必要があります。
- `-ek` または `-ep` オプションを使用すると、暗号化されていないデータベースをアンロードし、強力に暗号化された新しいデータベースに再ロードできます。`-ep` および `-an` を使用するときは、キーが正しいことを確認してください。正しくない場合は、アンロードは失敗します。
- 元のデータベースが強力に暗号化されていても、`-ek` または `-ep` オプションを使用しないと、新しいデータベースは単純な難読化でエンコードされます。
- `-an` が指定されていない場合は、`-ek` と `-ep` オプションは無視されます。dbunload の `-ek` と `-ep` オプションが新しいデータベースに適用されるのに対し、データベースサーバ (dbeng17/dbsrv17) オプションと DBKEY= は既存のデータベースに適用されます。
- 同期またはレプリケーションに使用しているデータベースを再構築する場合、dbunload は、`-ek` または `-ep` オプションで指定された暗号化キーが元のデータベースおよび再構築されるデータベースの暗号化キーであると見なします。

新しいデータベースを暗号化するとき、または新しいデータベース内のテーブル暗号化を有効にするときには、暗号化キーを指定します。本ソフトウェアではパスワードベースの鍵導出関数 #2 (PBKDF2) を使用します。これは、総当たり攻撃からキーを保護する PKCS#5 標準の一部です。本ソフトウェアでは暗号化キーに対して暗号化ハッシュを繰り返し適用します。ハッシュの適用回数を指定するには、`-kdi` オプションを使用します。

## データベースの再構築

データベースをアンロードするには、DBA ユーザとパスワードを指定し、DBF= 接続パラメータでデータベースを参照して、`-up` オプションを使用してパスワードをアンロードして dbunload を実行します。

データベースを再ロードするには、新しいデータベースを作成し、生成された `reload.sql` スクリプトファイルを Interactive SQL を介して実行します。

アンロードと再ロードを一度に行うには、dbunload を実行するときに、上記のアンロードに必要なオプションを指定するとともに、`-an` オプションで新しいデータベースファイルの名前を指定します。`-ac` と `-an` オプションの説明を参照してください。

### 例

次の例では、データベースを `reload.sql` ファイルにアンロードします。CREATE INDEX 文と LOAD TABLE 文は、BEGIN PARALLEL WORK 文の内部にまとめられます。

```
dbunload -c "UID=DBA;PWD=sql;DBF=mydemo.db" -bp -r reload.sql
```

## 関連情報

[単純難読化と強力な暗号化 \[1615 ページ\]](#)

[診断トレーシング \(廃止予定\) \[1438 ページ\]](#)

[ログインポリシー \[613 ページ\]](#)

[チェックサムを使用した破損の検出 \[947 ページ\]](#)

[DatabaseKey \(DBKEY\) 接続パラメータ \[67 ページ\]](#)

### 1.7.4.30 アップグレードユーティリティ (dbupgrad)

システムテーブルとビューの更新、新しいデータベースオプションの追加、すべてのシステムストアプロシージャの再作成、jConnect サポートのインストール、トランザクションログのアーカイブ、新しいトランザクションログの作成を行います。

#### i 注記

アップグレードユーティリティ (dbupgrad) を使用して、SQL Anywhere 9.0.2 以前のデータベースを SQL Anywhere17 にアップグレードすることはできません。SQL Anywhere 9.0.2 以前のデータベースを SQL Anywhere17 にアップグレードするには、アンロードと再ロードを実行してデータベースを再構築する必要があります。

また、このユーティリティはデータベースを停止して再起動します。

#### 構文

```
dbupgrad [ options ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-C"keyword=value; ..."	接続パラメータを指定します。  たとえば、次のコマンドは、sample17 というデータベースにパスワード passwd を指定してユーザ DBA として接続し、jConnect サポートをインストールせずにアップグレードします。  <pre>dbupgrad -c "UID=DBA;PWD=passwd;DBF=c:¥sa17¥sample17.db"</pre>

オプション	説明
<code>-i</code>	jConnect メタデータのサポートを除外します。jConnect JDBC ドライバを使用してシステムカタログ情報にアクセスする場合は、このオプションを使用しないでください。このオプションを指定した場合でも、システムカタログ情報にアクセスしないかぎり、jConnect を使用できます。SQL Central または ALTER DATABASE UPGRADE 文を使用して、jConnect メタデータのサポートを後から追加することもできます。
<code>-nrs</code>	アップグレード後にデータベースを停止しますが、再起動はしません。このオプションを指定しない場合は、AutoStop (ASTOP) 接続パラメータを Yes に設定しないかぎり、アップグレード後にデータベースが再起動されます。
<code>-ofilename</code>	指定されたファイルに出力メッセージを書き込みます。
<code>-pd [ Y   N ]</code>	<p>16.0 より前のシステムプロシージャのうち、権限を必要とする操作を実行するものを、定義側 (所有者) の権限で実行するの、呼び出し側の権限で実行するのを指定します。</p> <p><code>-pd N</code> を指定すると、システムプロシージャが呼び出し側 (システムプロシージャを実行しようとしているユーザ) の権限で実行されます。</p> <p><code>-pd Y</code> を指定すると、システムプロシージャが定義側 (システムプロシージャを所有しているユーザまたはロール) の権限で実行されます。</p> <p>何も指定しなければ、アップグレードされるデータベースの現在の設定が維持されます。16.0 より前のデータベースをアップグレードする場合は、これによってシステムプロシージャが定義側の権限で実行されます。</p> <p>この設定は、バージョン 16.0 以降で追加されたシステムプロシージャには影響を与えず、ユーザ定義のプロシージャを実行する場合のデフォルトの動作にも影響を与えません (特に指定しないかぎり、定義側の権限で実行されます)。</p>
<code>-q</code>	クワイエットモードで実行します (ウィンドウを表示しません)。

## 権限

ALTER DATABASE システム権限が必要です。また、このデータベースに他の接続がないことが必要です。

## 備考

### 警告

データベースファイルをバックアップしてからアップグレードしてください。

dbupgrad ユーティリティは、このソフトウェアの以前のバージョンで作成されたデータベースをアップグレードし、現在のバージョンが提供する機能を使用できるようにします。アップグレードできる以前のバージョンは SQL Anywhere 10.0.0 です。このソフトウェアの初期のリリースで作成したデータベースに対して、そのバージョン以降のデータベースサーバを実行できますが、データベースを作成した後に導入された機能の中には、データベースをアップグレードしないかぎり使用できないものもあります。

アップグレードユーティリティを使用して、現在ミラーリングされているデータベースをアップグレードすると、エラーメッセージが返されます。

### 注記

データベースのアップグレード後にデータベースのマテリアライズドビューを再表示してください。

### 注記

データベースファイルの物理的な再編成を必要とする機能は、dbupgrad を使用しても使用できるようなりません。そのような機能には、インデックスの拡張やデータの格納に関する変更が含まれています。これらの拡張機能を利用するには、データベースのアンロードと再ロードを行ってください。

データベースのアップグレードでは、データベースをアンロードして再ロードする必要はありません。

アップグレード中、トランザクションログがアーカイブされます。アップグレード後、データベースの起動時に新しいトランザクションログが作成されます。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

ユーザ定義のアップグレードは、アップグレードユーティリティ (dbupgrad) ではサポートされません。

## 関連情報

[設定ファイル \[1066 ページ\]](#)

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

[データベースのバックアップとリカバリ \[894 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

## 1.7.4.31 検証ユーティリティ (dbvalid)

テーブルとマテリアライズドビューのインデックスとキーを検証します。

### 構文

```
dbvalid [ options ] [ object-name, ... ]
```

オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。  設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。
-C"keyword=value;..."	データベース接続パラメータを指定します。  たとえば、次のコマンドは、データベース c: ¥salesdata.db (テーブルとマテリアライズドビューを含む) を検証します。  <pre>dbvalid -c "UID=DBA;PWD=passwd;DBF=c:¥salesdata.db"</pre>
-d	データベース内のすべてのページが正しいオブジェクトに属しているかどうかを検証し、チェックサム検証を実行します。-d オプションはインデックスの正当性を検証します。-d オプションを -i、-s、または -t オプションと一緒に使用することはできません。
-fx	プライマリキー/外部キー参照整合性チェックを無効にします。-fx オプションは、-i オプションとは一緒に使用できません。
-i	object-name をインデックスのリストとして定義します。

オプション	説明
<p><code>-immode</code></p>	<p>データベースサーバが自動起動の場合、インメモリモードを使用するように定義します (接続文字列の StartLine パラメータで、<code>-im</code> データベースサーバオプションを設定する)。</p> <p>モードは、次のいずれかです。</p> <p><b>none</b></p> <p>インメモリモードは使用されません。</p> <p><b>c</b></p> <p>チェックポイント専用です。このモードは、パフォーマンスの向上が必要で、最新のチェックポイントの実行後にコミットされたトランザクションが失われても構わないアプリケーションでの使用に適しています (データベースサーバをチェックポイント専用モードで実行すると、データベースサーバはコミット操作ごとにチェックポイントの設定を実行しません)。</p> <p><b>nw</b></p> <p>書き込みを行いません。コミットされたトランザクションはディスク上のデータベースファイルに書き込まれません。データベースが停止またはクラッシュするとすべての変更は失われるため、データベースファイルは常に元の状態に保たれます。</p> <p><b>v</b> または <b>validation</b> (デフォルト)</p> <p>インメモリの検証です。データベースのバックアップコピーまたはそのトランザクションログファイルを変更せずにデータベースのバックアップコピーを検証するには、このモードを使用します。</p>
<p><code>-Ofilename</code></p>	<p>指定したファイルに、出力メッセージを追加します。</p>
<p><code>-q</code></p>	<p>出力メッセージをクライアントに表示しません。<code>-o</code> オプションを使用してメッセージをファイルに書き込むことは可能です。</p>
<p><code>-s</code></p>	<p>チェックサムを使用してデータベースを検証します。チェックサムは、データベースページがディスク上で変更されたかどうかを判断するために使用します。チェックサム検証では、データベースの各ページをディスクから読み取り、チェックサムが存在すればそのチェックサムを計算します。計算されたチェックサムがページに保存されているチェックサムと異なる場合、ディスク上でページが変更されていることを意味し、エラーが返されます。無効なページのページ番号がデータベースサーバメッセージウィンドウに表示されます。<code>-s</code> オプションは、<code>-d</code>、<code>-i</code>、<code>-t</code>、または <code>-fx</code> オプションとは一緒に使用できません。</p>
<p><code>-t</code></p>	<p><code>object-name</code> をテーブルとマテリアライズドビューのリストとして定義します。</p>
<p><code>-wl</code></p>	<p>テーブルに適用されている排他データロックを使用して検証を実行します。トランザクションは、テーブルデータまたはスキーマを読み込むことができますが、修正することはできません。</p>

オプション	説明
-ws	テーブルに適用されているスナップショットアイソレーションを使用して検証を実行します。トランザクションは、データを読み込むことも修正することもできます。このオプションを使用するには、データベースのスナップショットアイソレーションが有効になっている必要があります。この句ではスナップショットアイソレーションを使用するため、パフォーマンスに影響することがよくあります。
object-name	検証するテーブル、マテリアライズドビュー、インデックスの名前を指定します。  -i を使用した場合、object-name は検証するインデックスを表します。

## 権限

VALIDATE ANY OBJECT システム権限が必要です。特定のインデックスを指定する場合は、インデックスが作成されるテーブルの所有者であるか、VALIDATE ANY OBJECT システム権限を持っている必要があります。

## 備考

検証ユーティリティを使用すると、データベース内のテーブルとマテリアライズドビューの一部またはすべてについて、インデックスとキーを検証できます。検証ユーティリティでは、データベース内のすべてのページが正しいオブジェクトに属し、ページチェックサムが正しいことを確認するデータベースファイル構造を検証できます。デフォルトでは、dbvalid はデータベース内のすべてのテーブル、マテリアライズドビュー、インデックスを検証し、データベースファイル構造を検証します。

テーブルを検証するときに、dbvalid はすべてのテーブルのインデックスも検証して、テーブル内の一連のローと値が各インデックスに含まれる一連のローと値に一致するかどうかを確認します。テーブル内のすべての BLOB もトラバースされ、BLOB アロケーションマップが検証されて、孤立した BLOB が検出されます。また、検証ユーティリティは、すべてのインデックスページの物理構造、インデックスハッシュ値の順序、インデックスの一意性要件 (指定されている場合) も検査します。-fx オプションが指定されない場合、各外部キー値は、参照整合性制約がそのまま残っているかどうかについて、対応するプライマリキーテーブルで検索されます。

-i オプションが指定されたときには、dbvalid はオブジェクトリスト内の各インデックスを検証します。インデックスの検証は、テーブルの検証とまったく同じように処理されますが、指定されたインデックスとその下にあるテーブルが検証されることだけが異なります。インデックスが外部キーの場合は、-fx オプションが指定されないかぎり、それぞれの値がプライマリキーテーブルで検索されます。

データベースクリーナーが実行している間にデータベースの検証を開始すると、クリーナーの実行が終了するまで検証は実行されません。

検証ユーティリティを通常のバックアップと一緒に使用すると、データベースのデータの整合性を保持できます。データベースのバックアップコピーを検証する場合は、バックアップのコピーを作成し、そのコピーを検証してください。この処理によって、リカバリに使用されるファイルが変更されないようにすることができます。

## 警告

データベースとトランザクションログのバックアップコピーには、どのような変更でも加えるべきではありません。バックアップ中に処理中のトランザクションがなかった場合、または BACKUP DATABASE WITH CHECKPOINT LOG RECOVER か WITH CHECKPOINT LOG NO COPY を指定した場合は、読み込み専用モードを使用するか、バックアップデータベースのコピーを検証することによって、バックアップデータベースの妥当性をチェックできます。

一方、トランザクションの処理中だった場合、または BACKUP DATABASE WITH CHECKPOINT LOG COPY を指定した場合は、検証の開始時にデータベースサーバがデータベースのリカバリを実行する必要が生じます。リカバリを実行するとデータベースのバックアップコピーに変更が加えられるため、元のデータベースからの以降のトランザクションログファイルが適用されなくなります。この場合は、インメモリ検証を使用してデータベースとトランザクションが変更されないようにするか、データベースのコピーを検証してください。

検証ユーティリティの実行によってデータベースを自動的に起動する場合、データベースは読み込み専用モードで起動されます。この処理が、バックアップまたはリカバリのプランの一環として検証を行う場合に、データベースに変更が加えられることを防いでいます。

検証ユーティリティが、読み込み専用モードで起動していない実行中のデータベースに接続すると、警告が表示されます。この警告は、検証しようとしているデータベースを、リカバリプランの中で使用できないことを知らせるものです。バックアップの実行によって、dbbackup で作成したほとんどのデータベースには、リカバリが必要であることを示すマークが付けられます。検証対象のデータベースがリカバリを必要としており、そのデータベースを読み込み/書き込みモードで起動したい場合は、dbvalid の実行前にデータベースを起動するか、DBS 接続パラメータに有効な値を指定してください。

次のコマンドはどちらも、mycopy.db データベースのリカバリが必要な場合に dbvalid を実行できるようにします。

```
dbvalid -c "UID=DBA;PWD=passwd;DBF=mycopy.db;DBS=-n mycopy"
```

```
dbvalid -c "UID=DBA;PWD=passwd;DBF=mycopy.db;DBS=-dh"
```

## 警告

検証は、検証対象のテーブルに対する排他アクセスを取得せず、検証中にテーブルが変更された場合には、失敗をレポートします。-wl または -ws オプションを指定しない場合は、テーブルまたはデータベース全体の検証は、テーブルに変更を加えている接続がないときに実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

検証ユーティリティは、グローバルチェックサムが有効になっていないデータベースで、チェックサム違反に関する警告を返すことがあります。これは、データベースサーバが、チェックサムが有効かどうかに関係なく、重要なデータベースページのチェックサムを自動的に計算するからです。データベースの一部のページにチェックサムが存在する場合があります。これは、データベースが書き込みチェックサムを有効にして起動されたためです。

データベースサーバは、信頼性の低いストレージメディア (リムーバブルドライブなど) 上で実行されているデータベースに対して自動的にチェックサムを作成します。

終了コードは、0 (成功) または 0 以外の値 (失敗) です。

## 関連情報

[設定ファイル \[1066 ページ\]](#)

- [データベースのバックアップとリカバリ \[894 ページ\]](#)
- [チェックサムを使用した破損の検出 \[947 ページ\]](#)
- [データベースの検証 \(SQL Central\) \[944 ページ\]](#)
- [ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)
- [DatabaseSwitches \(DBS\) 接続パラメータ \[70 ページ\]](#)
- [-im データベースサーバオプション \[437 ページ\]](#)

## 1.7.4.32 バージョン診断ユーティリティ (dbversion)

指定された実行プログラムに関する情報を返します。

### 構文

```
dbversion executable-name
```

### 備考

このユーティリティは、UNIX でのみ使用でき、SQL Anywhere の実行プログラムに関する情報を返します。

### 例

次にコマンド例を示します。

```
$ dbversion /opt/sqlanywhere17/bin32/dbversion
```

このコマンドは、次のように dbversion 実行プログラムに関する情報を返します。

```
SQL Anywhere バージョン診断ユーティリティバージョン
17.0.4.1691
/opt/sqlanywhere17/bin32/dbversion: dbversion GA 17 0 0 1403 linux 2012/10/10
15:32:50 nothr 32 production
```

フィールド	説明
dbversion	実行プログラム名を返します。
GA	インストールの種類を示す 2 文字のコードを返します。
17	メジャーバージョン番号を返します。
0	マイナーバージョン番号を返します。
0	サポートパッケージ番号を返します。
1403	ビルド番号を返します。
linux	オペレーティングシステムコードを返します。
2012/10/10 15:32:50	ビルド日時のスタンプを返します。

フィールド	説明
nothr	スレッドモデルを返します (nothr または posix)。
32	実行プログラムのビット数を返します (32 または 64)。
production	production または debug を返します。

## 関連情報

[-v データベースサーバオプション \[498 ページ\]](#)

## 1.8 パフォーマンスの向上、診断、モニタリング

モニタリング、データベースパフォーマンスの改善、および問題の診断を行う方法に関するヒントを参照してください。

このセクションの内容:

[SQL Anywhere モニタ、コックピット、およびプロファイラの比較 \[1206 ページ\]](#)

コックピットはデータベースサーバの実行内容を示し、モニタは現在使用可能なデータベースサーバを示します。また、プロファイラはデータベース遅延の原因調査に役立ちます。

[モニタリング \[1206 ページ\]](#)

データベースおよびデータベースサーバのステータスをモニタリングするために、たくさんのツールを使用することができます。

[パフォーマンス \[1349 ページ\]](#)

データベースおよびデータベースサーバのパフォーマンスを改善するために、たくさんのツールを使用することができます。

[診断 \[1411 ページ\]](#)

データベースおよびデータベースサーバの問題を診断するために、たくさんのツールを使用することができます。

## 関連情報

[SQL Anywhere コックピット \[1349 ページ\]](#)

[SQL Anywhere モニタ \[1207 ページ\]](#)

[SQL Anywhere プロファイラ \[1411 ページ\]](#)

[SQL Anywhere プロファイラ \[1411 ページ\]](#)

## 1.8.1 SQL Anywhere モニタ、コックピット、およびプロファイラの比較

コックピットはデータベースサーバの実行内容を示し、モニタは現在使用可能なデータベースサーバを示します。また、プロファイラはデータベース遅延の原因調査に役立ちます。

### コックピット

コックピットを使用して、データベースサーバや稼働中のデータベースで発生している内容を迅速に確認し、問題が発生した場合に電子メール通知を受信します。コックピットでは、データベースサーバの問題を示すだけでなく、接続ユーザやデータベース、データベースサーバのプロパティ、データベースサーバのメッセージも容易に表示します。

### プロファイラ

プロファイラを使用して、開発中および運用中のアプリケーションのトラブルシューティングや微調整を行います。プロファイラでは、ストアードプロシージャ、トリガ、イベントに含まれる SQL 文など、データベースで実行されるすべての SQL 文を記録することができます。ストアード SQL プロファイリング機能を使用して、こうしたストアード SQL オブジェクトの実行時間を別の実装と比較し、ベンチマークを実行します。すべての文をプロファイリングすることにより、ブロックやデッドロックが発生した接続の詳細を確認します。

### モニタ

SQL Anywhere モニタを使用して、企業内のすべてのサーバの正常性と可用性を確認し、問題の発生時に通知の電子メールを受信します。モニタにより、複数のデータベース、データベースサーバ、Mobile Link サーバ、サーバファームのほか、Web サーバ、プロキシサーバ、ホストコンピュータを処理します。

### 関連情報

[SQL Anywhere コックピット \[1349 ページ\]](#)

[SQL Anywhere プロファイラ \[1411 ページ\]](#)

[SQL Anywhere モニタ \[1207 ページ\]](#)

## 1.8.2 モニタリング

データベースおよびデータベースサーバのステータスをモニタリングするために、たくさんのツールを使用することができます。

このセクションの内容:

#### [SQL Anywhere モニタ \[1207 ページ\]](#)

SQL Anywhere モニタは、SQL Anywhere データベース、Mobile Link サーバ、Mobile Link サーバファームの正常性や可用性に関する情報を示すブラウザベースの管理ツールで、単に "モニタ" とも呼ばれます。Web サーバ、プロキシサーバ、およびホストコンピュータの可用性についての情報も提供することができます。

#### [Solution Manager \[1300 ページ\]](#)

SAP Solution Manager は全体的な SAP ランドスケープの一部として、SQL Anywhere をモニタすることができるツールを提供しています。

#### [SQL Anywhere SNMP Extension Agent \[1300 ページ\]](#)

SQL Anywhere SNMP Extension Agent を SNMP 管理アプリケーションとともに使用して、データベースを管理できます。

## 1.8.2.1 SQL Anywhere モニタ

SQL Anywhere モニタは、SQL Anywhere データベース、Mobile Link サーバ、Mobile Link サーバファームの正常性や可用性に関する情報を示すブラウザベースの管理ツールで、単に "モニタ" とも呼ばれます。Web サーバ、プロキシサーバ、およびホストコンピュータの可用性についての情報も提供することができます。

モニタには次の機能があります。

#### データの常時収集

他の多くの管理ツールとは異なり、SQL Anywhere モニタは、ユーザがブラウザでログインしていない時も含め、メトリックを常時収集します。また、シャットダウンされるまでメトリックの収集を続けます。

#### 電子メールによる警告の通知

メトリックが収集されると、モニタでは、収集したメトリックを検証し、リソースの異常を示す状況を検出した場合に、警告の電子メールを送信します。

#### ブラウザベースのインターフェース

いつでもブラウザを使用してモニタにログインし、収集された警告やメトリックを表示できます。

複数のデータベース、Mobile Link サーバ、Mobile Link サーバファーム、Web サーバ、プロキシサーバ、およびホストコンピュータをモニタ

1 つのツールで、同一または異なるコンピュータで実行されている複数のリソースを同時にモニタリングできます。

#### パフォーマンスへの影響を最小化

モニタリングによってパフォーマンスが低下することはないため、開発環境や運用環境でモニタを日常的に使用できます。

モニタは、次のタスクを担当するユーザを、そのユーザがデータベース管理者かどうかに関係なく支援するよう設計されています。

- データベースまたは Mobile Link サーバがネットワークに接続されていることを確認します。
- データベースまたは Mobile Link サーバに十分なディスク領域またはメモリがあることを確認します。
- ユーザがブロックされていないこと、またはクエリの実行に長時間かかっていないことを確認します。
- 所定の期間内に Mobile Link サーバが実行した同期の回数を確認します。
- Web サーバ、プロキシサーバ、またはホストコンピュータが使用可能であることを確認します。

このセクションの内容:

#### [モニタのアーキテクチャ \[1209 ページ\]](#)

モニタは、他のコンピュータ上で実行されているデータベース (読み込み専用スケールアウトシステムとミラーリングシステムに関わっているデータベースを含む)、Mobile Link サーバおよび Mobile Link サーバファームからメトリックやパフォーマンスデータを収集します。その間、別のコンピュータがブラウザ経由でモニタにアクセスします。さらに、Web サーバ、プロキシサーバ、およびホストコンピュータの可用性もモニタすることができます。

#### [モニタを使用するためのクイックスタート \[1211 ページ\]](#)

データベース、Mobile Link サーバ、および Mobile Link サーバファームをモニタします。

#### [モニタの起動 \[1212 ページ\]](#)

モニタを起動してログインします。

#### [モニタの停止 \[1214 ページ\]](#)

Monitor Production Edition を停止する場合は、モニタサービスとメトリックの収集を停止します。

#### [モニタにリモートでログイン \[1216 ページ\]](#)

モニタの URL をブラウザして、モニタにログインします。

#### [モニタからのログアウト \[1217 ページ\]](#)

モニタからログアウトします。

#### [アクセシビリティ機能のカスタマイズ \[1218 ページ\]](#)

モニタのフォントサイズとカラーテーマを変更します。

#### [ダッシュボードおよびウィジェット \[1218 ページ\]](#)

概要ダッシュボードでは、モニタリングされているリソースの正常性や可用性の概要が表示されます。デフォルトで、概要ダッシュボードには、リソースリストウィジェットと警告リストウィジェットがあります。

#### [モニタの管理ウィンドウ \[1225 ページ\]](#)

モニタ管理者として、モニタリングするリソース (たとえば、データベース、Mobile Link サーバおよび Mobile Link サーバファーム) を選択し、次の操作を行うことができます。

#### [リソース \[1227 ページ\]](#)

リソースとは、データベース (読み込み専用スケールアウトシステムまたはミラーリングシステムのデータベースを含む)、Mobile Link サーバ、Mobile Link サーバファーム、または Web サービスのことです。

#### [メトリック \[1245 ページ\]](#)

モニタは、データベース、Mobile Link サーバ、および Mobile Link サーバファームから、次のような多くのメトリックを収集して保存します。

#### [モニタのユーザ \[1266 ページ\]](#)

モニタユーザがモニタにログインするには、ユーザ名とパスワードを使用します。

#### [警告 \[1273 ページ\]](#)

警告とは、モニタ管理者またはオペレータによる注目を要する、リソースに関する特定の状況または状態のことです。

#### [モニタのバックアップ \[1281 ページ\]](#)

モニタでは、デフォルトで、毎日 1 回夜中の 12 時にメトリックのメンテナンスが実行されます。メンテナンスの対象はメトリックで、警告は対象ではありません。

#### [モニタがインストールしたデータベースオブジェクトのリスト \[1283 ページ\]](#)

データベースをモニタリング対象のリソースとして追加する場合にインストールされるオブジェクトがいくつかあります。

#### [リソースデータベースからのモニタリングオブジェクトの削除 \[1284 ページ\]](#)

---

リソースをモニタに追加したとき、インストールされたリソースデータベースからモニタリングオブジェクトを削除します。

[モニタ通信のセキュリティ \[1285 ページ\]](#)

モニタとブラウザ間、およびモニタとモニタリング対象のリソース間の通信を保護できます。

[モニタのトラブルシューティング \[1286 ページ\]](#)

モニタ管理者がモニタのトラブルシューティングする方法は、いくつかあります。

[チュートリアル: モニタによるリソースのモニタリング \[1288 ページ\]](#)

データベース、または Mobile Link サーバのモニタリングを設定します。

## 関連情報

[SQL Anywhere プロファイラ \[1411 ページ\]](#)

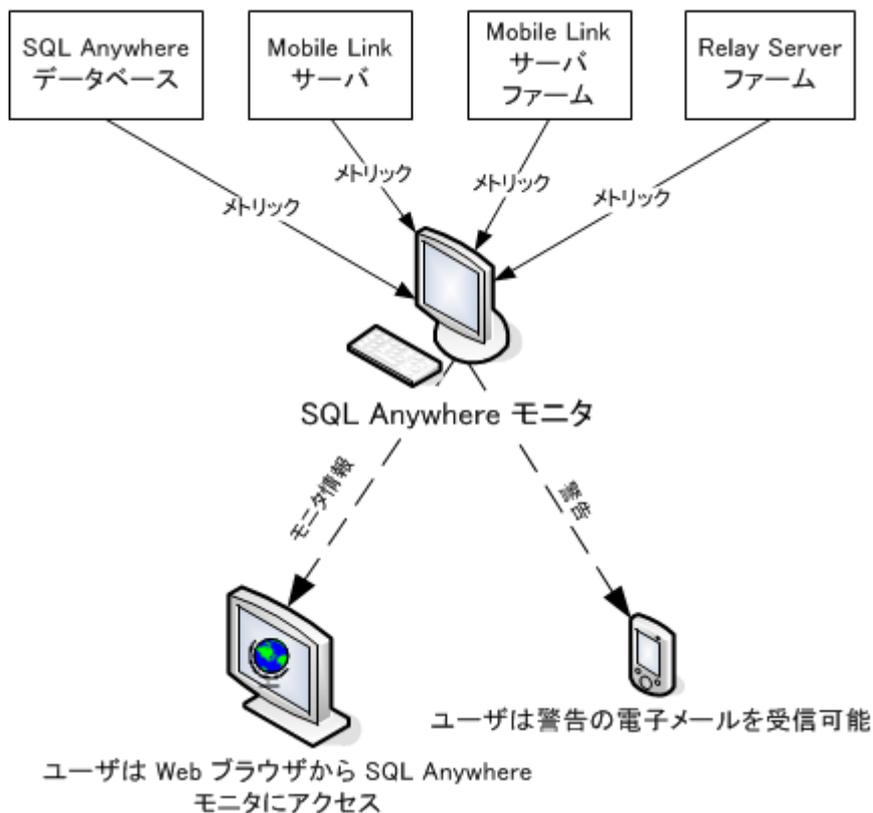
[Windows パフォーマンスモニタ \[1434 ページ\]](#)

[SQL Anywhere コックピット \[1349 ページ\]](#)

[パフォーマンス統計値ユーティリティ \(dbstats\) \(UNIX\) \[1138 ページ\]](#)

### 1.8.2.1.1 モニタのアーキテクチャ

モニタは、他のコンピュータ上で実行されているデータベース（読み込み専用スケールアウトシステムとミラーリングシステムに関わっているデータベースを含む）、Mobile Link サーバおよび Mobile Link サーバファームからメトリックやパフォーマンスデータを収集します。その間、別のコンピュータがブラウザ経由でモニタにアクセスします。さらに、Web サーバ、プロキシサーバ、およびホストコンピュータの可用性もモニタすることができます。



モニタは、SQL Anywhere の組み込みの HTTP サーバを介して Web ブラウザで実行される Flash ベースのアプリケーションです。

モニタには、次の 2 つのエディションがあります。

#### SQL Anywhere モニタ Developer Edition

モニタ Developer Edition は、開発とテストの目的で使用されます。モニタ Developer Edition はデフォルトでインストールに含まれており、インストールされたソフトウェアをバックエンドで使用します。

#### SQL Anywhere モニタ Production Edition

Production Edition は、配備と運用の目的で使用されます。別個にインストールされ、サービスとして実行されます。Production Edition には、すべての機能が搭載された SQL Anywhere インストールが含まれます。このエディションは、SQL Anywhere の特定のエディションでのみ使用できます。

## 前提条件

モニタがインストールされているコンピュータの場合

- Windows ファイアウォールが有効になっている Windows システムでモニタを実行するには、ポート 4950 のポート例外を追加する必要があります。
- FIPS 認定の暗号化を使用して保護されたリソースをモニタリングするには、別途ライセンスが必要です。

- モニタはモニタリング対象のリソースと同じコンピュータ上で実行できますが、特に運用環境では、データベースサーバ、Mobile Link サーバまたはその他のアプリケーションへの影響を最小限に抑えるために、モニタを別のコンピュータ上で実行することをお奨めします。
- 運用環境では、モニタ Production Edition を使用することをお奨めします。
- モニタ Developer Edition を実行している場合は、SQL Anywhere Server 17 がインストールされている必要があります。モニタ Developer Edition では、インストールされた SQL Anywhere をバックエンドで使用します。

#### モニタにアクセスするコンピュータの場合

- オペレーティングシステムに対応している Adobe Flash Player の最新バージョンをインストールします。モニタには、Adobe Flash Player のバージョン 10 との下位互換性があります。
- ブラウザで JavaScript を有効にします。
- SQL Anywhere モニタへのアクセスに使用するブラウザが実行されているコンピュータが、モニタがインストールされているネットワークに接続されていることを確認します。  
モニタには、モニタが実行されているのと同じコンピュータからアクセスできますが、特に運用環境では、別のコンピュータからアクセスすることをお奨めします。

## 制限事項

- コンピュータ上で一度に実行できるモニタのエディションは 1 つだけです。
- Linux では、インストールしたユーザだけがモニタ Developer Edition を実行できます。
- Linux では、Linux 管理者権限を持つユーザだけがモニタ Production Edition をインストールし、実行できます。
- モニタは、TCP/IP 経由でデータベースに接続します。ローカルデータベースサーバ (dbeng17) 上で実行中のデータベースをモニタリングするには、データベースサーバを -x all または -x tcpip オプションで起動します。
- クエリの最適化やアプリケーションの速度の測定には使用できません。データベースやアプリケーションのパフォーマンスをチューニングするために、**プロファイラ** または **コックピット** などのツールを使用することができます。
- 個々の同期に関する情報は提供しません。タイミングなどの同期別の統計を含む、個々の同期に関する詳細情報を取得するには、Mobile Link プロファイラを使用します。
- データベースのバージョン、Mobile Link サーバ、および Mobile Link サーバファームについては、<http://scn.sap.com/docs/DOC-63186> を参照してください。

## 関連情報

[モニタ通信のセキュリティ \[1285 ページ\]](#)

### 1.8.2.1.2 モニタを使用するためのクイックスタート

データベース、Mobile Link サーバ、および Mobile Link サーバファームをモニタします。

1. モニタリングするデータベース、または Mobile Link サーバを起動します (実行されていない場合)。
2. ネットワークに常時接続されるコンピュータにモニタをインストールします。

SQL Anywhere モニタはモニタリング対象のリソースと同じコンピュータ上で実行できますが、特に運用環境では、モニタを別のコンピュータ上で実行して、リソースまたはその他のアプリケーションへの影響を最小限に抑えることをおすすめします。

3. SQL Anywhere モニタを起動し、ブラウザで開きます。
4. モニタに管理者としてログインします。デフォルトのユーザは、名前 `admin` とパスワード `admin` を持つ管理者です。

#### **i** 注記

次のタスクを実行するには、モニタに管理者としてログインしてください。

5. モニタ管理者として、左のナビゲーションメニューで、**ツール** > **管理** をクリックして、モニタリング対象のリソースを追加します。
6. モニタ管理者として、新規ユーザを追加し、admin ユーザのパスワードを変更します。
7. モニタ管理者として、リソースに対する警告のスレッシュホールドを設定します。
8. モニタ管理者として、バックアップとメンテナンスのスケジュールを設定します。
9. **閉じる** をクリックして、**管理** ウィンドウを終了します。
10. **概要** をクリックします。リソースリストウィジェットに、モニタリング対象のリソースが表示されます。

## 関連情報

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

[モニタのユーザ \[1266 ページ\]](#)

[モニタのバックアップ \[1281 ページ\]](#)

[モニタの起動 \[1212 ページ\]](#)

[データベースリソースの追加 \[1228 ページ\]](#)

[Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

[モニタユーザの作成 \[1267 ページ\]](#)

[警告 \[1273 ページ\]](#)

### 1.8.2.1.3 モニタの起動

モニタを起動してログインします。

## 前提条件

モニタの別のコピーがコンピュータ上で動作していないことを確認します。

## コンテキスト

Monitor Production Edition は、デフォルトで、インストール時に自動的にサービスとして実行され、コンピュータの起動時に開始します。Monitor Production Edition を停止した場合は、次の指示に従って再起動できます。

## 手順

1. モニタがインストールされているコンピュータで、モニタを起動します。

### Windows の場合

1. **スタート** > **プログラム** > **SQL Anywhere17** > **管理ツール** > **SQL Anywhere モニタ** をクリックします。
2. または、Monitor Production Edition に対して、モニタのインストールディレクトリにある bin32 または bin64 ディレクトリから samonitor バッチファイルを実行してサービスを再度開始することもできます。

```
samonitor.bat start
```

### Linux の場合

モニタのインストールディレクトリの bin32 または bin64 ディレクトリから samonitor.sh スクリプトを実行します。

```
./samonitor.sh launch
```

### i 注記

Monitor Developer Edition を起動している場合は、モニタのアイコンがシステムトレイに表示されます。

モニタはリソースに接続し、モニタに指定されているすべてのリソースのメトリックの収集を開始します。

2. モニタにログインするための URL をブラウズします。デフォルト URL は、<http://localhost:4950> です。

### i 注記

モニタにリモートでログインしている場合、<http://computer-name:4950> をブラウズします。computer-name はモニタが稼働しているコンピュータの名前です。

3. ログインします。

ブラウザで、モニタのユーザ名とパスワードを入力します。モニタのユーザ ID とパスワードでは、大文字と小文字が区別されます。ユーザには、管理者、オペレータ、および読み込み専用ユーザの 3 つの異なるタイプがあります。それぞれのユーザタイプには、異なる権限が付与されています。デフォルトのユーザは、名前 **admin** とパスワード **admin** を持つモニタの管理者です。

[このコンピュータにログイン情報を記憶](#)をクリックし、2 週間またはログアウトするまでセッションが維持されるようにします。[このコンピュータにログイン情報を記憶をオフ](#)にすると、ブラウザを閉じるか、ログアウトしたときにセッションの有効期限が切れます。

## 結果

モニタが起動します。

## 次のステップ

モニタは何も通知せずにバックグラウンドで稼働するように設計されています。モニタとの対話は、ブラウザインタフェース経由で行います。メトリックを収集するには、バックグラウンドでモニタの実行を継続することをお奨めします。

## 関連情報

[モニタのユーザ \[1266 ページ\]](#)

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

[モニタの停止 \[1214 ページ\]](#)

[モニタからのログアウト \[1217 ページ\]](#)

[モニタにリモートでログイン \[1216 ページ\]](#)

### 1.8.2.1.4 モニタの停止

Monitor Production Edition を停止する場合は、モニタサービスとメトリックの収集を停止します。

## コンテキスト

### 警告

ほとんどの場合、モニタの実行は継続し、ブラウザを閉じることをお奨めします。ブラウザを閉じて、メトリックの収集は停止しません。

データベースといった特定のリソースを指定して、モニタを停止することもできます。

## 手順

モニタを停止します。

### Windows の場合

次のオプションのうちの 1 つを選択してください。

オプション	アクション
Monitor Developer Edition	システムトレイでモニタのアイコンを右クリックし、 <a href="#">SQL Anywhere モニタの終了</a> をクリックします。
Monitor Production Edition	モニタのインストールディレクトリにある bin32 ディレクトリから samonitor.bat を実行します。 <pre>samonitor.bat stop</pre>

#### Linux の場合

次のオプションのうちの 1 つを選択してください。

オプション	アクション
Monitor Developer Edition	<pre>./samonitor.sh stop</pre>
Monitor Production Edition	<pre>samonitor.sh stop</pre>

モニタのインストールディレクトリの bin32 または bin64 ディレクトリから samonitor.sh スクリプトを実行します。

## 結果

モニタが停止し、すべてのリソースのメトリックの収集を停止します。

## 関連情報

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

[リソースのモニタリングの停止 \[1239 ページ\]](#)

[モニタの起動 \[1212 ページ\]](#)

[モニタにリモートでログイン \[1216 ページ\]](#)

[モニタからのログアウト \[1217 ページ\]](#)

## 1.8.2.1.5 モニタにリモートでログイン

モニタの URL をブラウザして、モニタにログインします。

### 前提条件

モニタへのログインに使用するコンピュータは、モニタが実行されているネットワークに接続されている必要があります。

### 手順

1. モニタがインストールされているコンピュータで、モニタを起動します。
2. モニタにアクセスしているコンピュータで、モニタにログインするためのデフォルト URL `http://computer-name:4950` をブラウザします。`computer-name` は、モニタが実行されているコンピュータの名前です。たとえば、`http://samplehost:4950` のように記述します。
3. 要求された場合は、モニタのユーザ名とパスワードを入力します。モニタのユーザ ID とパスワードでは、大文字と小文字が区別されます。

### 結果

リモートコンピュータからモニタにログインしました。

### 関連情報

[モニタのユーザ \[1266 ページ\]](#)

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

[モニタの起動 \[1212 ページ\]](#)

[モニタの停止 \[1214 ページ\]](#)

[モニタからのログアウト \[1217 ページ\]](#)

## 1.8.2.1.6 モニタからのログアウト

モニタからログアウトします。

### コンテキスト

モニタからログアウトしても、メトリックの収集には影響しません。

### 手順

右上隅のログアウトをクリックします。

モニタへのログイン時に、このコンピュータにログイン情報を記憶をクリックすると、ブラウザを閉じてもモニタからログアウトしません。

### 結果

モニタからログアウトしました。

### 関連情報

[リソースのモニタリングの停止 \[1239 ページ\]](#)

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

[モニタの起動 \[1212 ページ\]](#)

[モニタの停止 \[1214 ページ\]](#)

[モニタにリモートでログイン \[1216 ページ\]](#)

## 1.8.2.1.7 アクセシビリティ機能のカスタマイズ

モニタのフォントサイズとカラーテーマを変更します。

### 手順

1. [同期の設定](#)をクリックします。
2. 次のオプションのうちの 1 つを選択してください。

オプション	アクション
フォントサイズの変更	<a href="#">フォントサイズ</a> ドロップダウンリストから新しいフォントサイズを選択します。
カラーテーマを変更します。	次のオプションのうちの 1 つを選択してください。 <ul style="list-style-type: none"><li>○ 標準</li><li>○ ハイコントラスト (黒)</li><li>○ ハイコントラスト (白)</li></ul>

3. [保存](#)をクリックします。

## 1.8.2.1.8 ダッシュボードおよびウィジェット

概要ダッシュボードでは、モニタリングされているリソースの正常性や可用性の概要が表示されます。デフォルトで、[概要ダッシュボード](#)には、[リソースリストウィジェット](#)と[警告リストウィジェット](#)があります。

### リソースリストウィジェット

[リソースリストウィジェット](#)には、モニタリングされているリソースがリストされた表が含まれ、各リソースの全体的な正常性に関する表示が提供されます。表には常に、[SQL Anywhere モニタ](#)という名前のデフォルトリソースがあります。このリソースは、モニタ自体の正常性についてレポートします。[SQL Anywhere モニタ](#)リソースを変更したり、このリソースのモニタリングを停止したりできません。

リソースに関する詳細情報を表示するには、[リソースリストウィジェット](#)にあるリソース名をクリックします。これで、選択したリソースのダッシュボードが開きます。

[リソースリストウィジェット](#)で、[ステータスカラム](#)には、モニタとそのリソース間の接続に関する情報が示されます。[ステータスカラム](#)には、リソースに対していずれかのユーザによる何らかの操作が必要かどうかを示されます。

リソースのモニタの状態は次のいずれかになります。

アイコン	ステータス	説明
アイコンなし	正常	リソースに未解決の警告はありません。

アイコン	ステータス	説明
	警告あり	リソースに1つ以上の未解決の警告があります。
	使用不可	リソースは使用できません。たとえば、リソースはダウンしています。
	モニタ停止	ブラックアウトのため、またはユーザがリソースのモニタリングを手動で停止したため、リソースはモニタリングされていません。

## 警告リストウィジェット

警告リストウィジェットには、モニタリング対象のリソースからの警告がリストされます。警告に関する詳細情報を表示するには、警告リストにある警告をクリックしてから、詳細をクリックします。これで、警告の詳細を表示したウィンドウが開きます。

警告の状態は次のいずれかになります。

アイコン	ステータス	説明
	アクティブ	警告の条件は引き続き適用されます。警告は解決されていません。
	非アクティブ	警告のトリガとなった問題は存在しなくなりましたが、警告は解決または削除されていません。
	解決済み	モニタ管理者またはオペレータが、その警告に解決済みのマークを付けました。

警告の重大度のレベルは次のいずれかになります。

アイコン	重大度	説明
	重大度高	重大度高の警告は、ユーザがすぐに注意を向ける必要がある問題を示します。たとえば、リソースがディスク領域の低スレッショルド値を超えた場合、重大度高の警告が発行されます。
	重大度中	重大度中の警告は、拡大する可能性があるため、ユーザが注意を向ける必要がある問題を示します。たとえば、リソースが CPU 使用率のスレッショルドを超えた場合、重大度中の警告が発行されます。

アイコン	重大度	説明
	重大度低	重大度低は問題を示します。たとえば、リソースが接続に失敗した場合、重大度低の警告が発行されます。

このセクションの内容:

[ウィジェットの管理 \[1220 ページ\]](#)

ダッシュボードからウィジェットを追加、編集、または削除します。

[SQL Anywhere スケールアウトトポロジウィジェット \[1221 ページ\]](#)

読み込み専用スケールアウトシステムおよびミラーリングシステムのトポロジは、SQL Anywhere スケールアウトトポロジウィジェットを作成すると、表示できます。

[ダッシュボードの管理 \[1223 ページ\]](#)

ダッシュボードを追加、編集、および削除するか、または既存のダッシュボードからテンプレートを作成します。

[モニタを使用した接続の切断 \[1224 ページ\]](#)

モニタを使用してデータベースリソースへの接続を表示して切断します。

[時刻の表示方法の概要 \[1224 ページ\]](#)

サーバ時間オフセットメトリックを使用して、表示される時間のオフセットを決定します。

## 関連情報

[警告 \[1273 ページ\]](#)

### 1.8.2.1.8.1 ウィジェットの管理

ダッシュボードからウィジェットを追加、編集、または削除します。

## コンテキスト

次の表には、リソースに対して作成可能なさまざまなタイプのメトリックウィジェットが含まれています。

メトリックのタイプ	説明
主要パフォーマンス指標	モニタリングしているリソースのために収集された情報を表示します。メトリックは、グラフと表のどちらでも表示を選択できます。
警告	リソースに対する警告のリストを表示します。
リソース	リソースのリストを表示します。
SQL Anywhere の接続	データベース接続のリストを表示します。

メトリックのタイプ	説明
SQL Anywhere のスケールアウトポロジ	ミラーリングおよびスケールアウトシステムのトポロジを表示します。

## 手順

次のオプションのうちの 1 つを選択してください。

オプション	アクション
ウィジェットの追加	<ol style="list-style-type: none"> <li>1. ダッシュボードを開きます。</li> <li>2. ダッシュボードの右上で、<b>カスタマイズ</b>をクリックし、<b>ウィジェットの追加</b>をクリックします。</li> <li>3. <b>必要なウィジェットのタイプを指定してください</b>フィールドで、<b>必要なウィジェットのタイプ</b>をクリックしてから、<b>次へ</b>をクリックします。</li> <li>4. <b>このウィジェットの名前を指定してください</b>フィールドでウィジェットの名前を指定します。</li> <li>5. <b>このウィジェットでどの警告を確認したいですか?</b>フィールドでリソースを指定します。</li> <li>6. ウィザードの指示に従います。</li> <li>7. <b>作成</b>をクリックします。</li> </ol>
ウィジェットの編集	<ol style="list-style-type: none"> <li>1. ウィジェットで、ウィジェットの右上隅にあるドロップダウン矢印をクリックします。</li> <li>2. ウィジェットの設定を編集します。</li> </ol>
ウィジェットの削除	<ol style="list-style-type: none"> <li>1. ウィジェットで、ウィジェットの右上隅にあるドロップダウン矢印をクリックします。</li> <li>2. ウィジェットを削除します。</li> </ol>

### 1.8.2.1.8.2 SQL Anywhere スケールアウトポロジウィジェット

読み込み専用スケールアウトシステムおよびミラーリングシステムのトポロジは、SQL Anywhere スケールアウトポロジウィジェットを作成すると、表示できます。

#### トポロジウィジェットでの読み込み専用スケールアウトシステムの表示

読み込み専用スケールアウトシステム用の SQL Anywhere スケールアウトポロジウィジェットは、次のフィールドとその可能な値を表示します。

サーバ名

スケールアウトシステムのノード名。

タイプ

サーバタイプ。タイプは、読み込み専用スケールアウトシステムが作成されたときにサーバが定義された方法によって、*partner - primary* または *コピー* のいずれかになります。

#### ステータス

ノードの接続状況。接続しましたまたは切断しましたのいずれか 1 つになります。

#### 接続

このノードへの現在の接続数。

#### 最終更新

サーバがステータスを最後に更新した時刻。

## トポロジウィジェットでのデータベースミラーリングシステムの表示

データベースミラーリングシステム用の *SQL Anywhere* スケールアウトトポロジウィジェットは、次のフィールドとその可能な値を表示します。

#### サーバ名

ミラーリングシステムにデータベースサーバの名前を表示します。

#### タイプ

サーバタイプが一覧表示され、次のいずれかの値が表示されます。

##### **partner - primary**

この値によって、現在プライマリとして動作しているサーバが識別されます。

##### **partner - mirror**

この値によって、現在ミラーとして動作しているサーバが識別されます。

##### **arbiter**

この値によって、監視サーバが識別されます。

#### ステータス

各サーバの接続状況が表示されます。

#### 接続

サーバへの現在の接続数が表示されます。

#### 最終更新

サーバのステータスが更新された最新の時刻が表示されます。

## 関連情報

[データベースミラーリング \[1660 ページ\]](#)

[読み込み専用スケールアウト \[1733 ページ\]](#)

[SQL Anywhere モニタ \[1207 ページ\]](#)

[チュートリアル: モニタによるリソースのモニタリング \[1288 ページ\]](#)

[チュートリアル: 読み込み専用のスケールアウトシステムの作成 \[1753 ページ\]](#)

[チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

## 1.8.2.1.8.3 ダッシュボードの管理

ダッシュボードを追加、編集、および削除するか、または既存のダッシュボードからテンプレートを作成します。

### コンテキスト

ダッシュボードは、各ユーザに固有です。テンプレートダッシュボードでは、ユーザは新しいダッシュボードに表示されるデフォルトのレイアウトとウィジェットを指定します。ダッシュボードのテンプレートを *SQL Anywhere* モニタリソース上に設定することはできません。

### 手順

1. モニタにログインします。
2. 次のオプションのうちの 1 つを選択してください。

オプション	アクション
ダッシュボードの追加	<ol style="list-style-type: none"><li>1. <b>ダッシュボード</b> ウィンドウ枠で、<b>新規追加</b> をクリックします。</li><li>2. ウィンドウの指示に従ってダッシュボードを追加します。</li></ol>
ダッシュボードの編集	<ol style="list-style-type: none"><li>1. ダッシュボードを開きます。</li><li>2. ダッシュボードの右上で、<b>カスタマイズ</b> をクリックし、次に <b>設定</b> をクリックします。</li></ol>
ダッシュボードの削除	<ol style="list-style-type: none"><li>1. ダッシュボードを開きます。</li><li>2. ダッシュボードの右上で、<b>カスタマイズ</b> をクリックし、次に <b>削除</b> をクリックします。</li></ol>
ダッシュボードテンプレートを作成するには、次の手順に従います。	<ol style="list-style-type: none"><li>1. テンプレートとして使用するダッシュボードを開きます。</li><li>2. ダッシュボードの右上で、<b>カスタマイズ</b> をクリックし、<b>テンプレートとして設定</b> をクリックします。</li></ol>

### 関連情報

[ウィジェットの管理 \[1220 ページ\]](#)

[ウィジェットの管理 \[1220 ページ\]](#)

## 1.8.2.1.8.4 モニタを使用した接続の切断

モニタを使用してデータベースリソースへの接続を表示して切断します。

### 前提条件

DROP CONNECTION システム権限を持っている、データベースのユーザのユーザ名とパスワードを知っている必要があります。

### 手順

1. モニタにログインします。
2. データベースリソースのダッシュボードを開きます。  
▶ **ダッシュボード** ▶ `name-of-dashboard` ▶ をクリックします。
3. **接続**ウィジェットで、接続を選択し、接続名の横にある **x** をクリックします。
4. プロンプトが表示されたら、データベースのユーザ名とパスワードを入力します。  
このクレデンシャルは、データベースとの接続および指定した接続の切断に使用されます。これにより、クレデンシャルがモニタから削除されます。
5. **OK** をクリックします。

## 1.8.2.1.8.5 時刻の表示方法の概要

**サーバ時間オフセット**メトリックを使用して、表示される時間のオフセットを決定します。

### コンテキスト

モニタで表示される時刻は、24 時間表示に基づいており、モニタが実行されているコンピュータのローカル時間に対応しています。

### 手順

1. モニタにログインします。
2. リソースのダッシュボードを開きます。

3. サーバ情報ウィジェットで、サーバ時間オフセットメトリックを確認します。

## 結果

サーバ時間オフセットには、モニタが実行されているコンピュータの時刻と、モニタデータの表示に使用しているコンピュータの時刻の差異が記録されます。

## 関連情報

[メトリック \[1245 ページ\]](#)

### 1.8.2.1.9 モニタの管理ウィンドウ

モニタ管理者として、モニタリングするリソース (たとえば、データベース、Mobile Link サーバおよび Mobile Link サーバファーム) を選択し、次の操作を行うことができます。

#### **i** 注記

管理ウィンドウにアクセスできるのはモニタ管理者だけです。

モニタ管理者は、次の操作を実行できます。

- リソースの追加、編集、削除。
- ユーザの追加および編集。
- モニタをバックアップおよび設定することができます。
- メッセージログの表示。
- 例外レポートの表示。

このセクションの内容:

[メッセージログの表示 \[1226 ページ\]](#)

メッセージログをテーブル内に表示します。最新のメッセージは先頭になります。

[例外レポートの表示 \[1226 ページ\]](#)

モニタで致命的なエラーやクラッシュが発生したときの、状況の詳細に関する例外レポートを表示します。

## 関連情報

[モニタのユーザ \[1266 ページ\]](#)

[モニタのバックアップ \[1281 ページ\]](#)

### 1.8.2.1.9.1 メッセージログの表示

メッセージログをテーブル内に表示します。最新のメッセージは先頭になります。

#### 前提条件

モニタの管理者である必要があります。

#### コンテキスト

メッセージログには、モニタの動作に関するモニタからの情報メッセージが表示されます。

#### 手順

1. モニタにログインします。
2. **▶ ツール ▶ 管理** をクリックします。
3. **メッセージログ** をクリックします。

#### 結果

メッセージログが表示されます。

### 1.8.2.1.9.2 例外レポートの表示

モニタで致命的なエラーやクラッシュが発生したときの、状況の詳細に関する例外レポートを表示します。

#### 前提条件

モニタの管理者である必要があります。

## 手順

1. モニタにログインします。
2. **ツール** > **管理** をクリックします。
3. **例外レポート** をクリックします。

## 結果

例外レポートが表示されます。

### 1.8.2.1.10 リソース

リソースとは、データベース (読み込み専用スケールアウトシステムまたはミラーリングシステムのデータベースを含む)、Mobile Link サーバ、Mobile Link サーバファーム、または Web サービスのことです。

モニタ管理者は、リソースをモニタに追加し、モニタリングを開始できます。

#### SQL Anywhere モニタリソース

SQL Anywhere モニタという名前のデフォルトリソースは、モニタ自体の正常性についてレポートします。このデフォルトリソースは、モニタが実行されているコンピュータをモニタリングし、モニタに問題が発生したときに警告を送信する場合に役立ちます。このリソースを変更したり、このリソースのモニタリングを停止することはできません。

このセクションの内容:

##### [データベースリソースの追加 \[1228 ページ\]](#)

データベース (読み込み専用スケールアウトシステムとミラーリングシステムのデータベースを含む) をリソースとしてモニタに追加します。

##### [Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

Mobile Link サーバファームを、モニタでモニタリングするリソースとして追加します。

##### [Mobile Link サーバファームリソースの追加 \[1232 ページ\]](#)

Mobile Link サーバファームをモニタリングするには、目的のリソースをモニタに追加します。

##### [Web サーバ、プロキシサーバ、またはホストコンピュータの可用性をモニタするリソースを追加 \[1233 ページ\]](#)

Web サーバ、プロキシサーバ、ホストコンピュータの可用性をモニタする Web サービスリソースを作成し、使用不可の場合は通知を受信します。

##### [CSV ファイルによる複数のリソースの追加 \[1234 ページ\]](#)

値をカンマで区切った CSV (comma-separated values) ファイルを作成し、それをインポートすることによって、モニタに複数のリソースを追加します。

##### [リソースのモニタリングの開始 \[1239 ページ\]](#)

リソースの起動をモニタリングする条件がいくつかあります。

#### [リソースのモニタリングの停止 \[1239 ページ\]](#)

モニタでメトリックを収集しない場合は、リソースのモニタリングを停止する必要があります。

#### [モニタでモニタリングするデータベースリソースの修復 \[1243 ページ\]](#)

データベースリソースを修復して、モニタによるモニタリングを再開します。

#### [リソースの削除 \[1244 ページ\]](#)

リソースを削除して、モニタがそのリソースをモニタリングすることを永久に停止し、そのリソースについて収集したメトリックを破棄します。

## 関連情報

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

### 1.8.2.1.10.1 データベースリソースの追加

データベース (読み込み専用スケールアウトシステムとミラーリングシステムのデータベースを含む) をリソースとしてモニタに追加します。

## 前提条件

モニタの管理者である必要があります。

バージョン 16 以降のデータベースを追加するには、SYS\_SAMONITOR\_ADMIN\_ROLE ロールを持つ、データベース内のユーザのユーザ ID とパスワードが必要です。それより前のバージョンのデータベースを追加するには、DBA 権限を持つ、データベース内のユーザのユーザ ID とパスワードが必要です。

## コンテキスト

一度に 1 つずつリソースを追加したり、CSV ファイルを使用して複数のリソースを追加したりできます。

読み込み専用スケールアウトシステムとその他のデータベースサーバで、ルートノード用のリソース設定には違いはありません。リソースが設定されると、モニタはそのリソースが読み込み専用のスケールアウトシステムのルートノードであることを検出します。

## 手順

1. モニタにログインします。
2. **管理**をクリックします。
3. **リソース**をクリックします。
4. **追加**をクリックします。
5. **リソースの追加**ウィンドウで、次の手順に従います。
  - a. **SQL Anywhere サーバ**をクリックします。
  - b. **次へ**をクリックします。
  - c. **名前**フィールドにリソースの名前を入力します。これは、モニタに表示された名前です。**次へ**をクリックします。
  - d. **ホスト**フィールドで、データベースサーバが実行されているコンピュータのホスト名または IP アドレスを指定します。現在のコンピュータを表す localhost という名前を使用できます。この値は必須です。

ミラーリングシステムのプライマリデータベースをモニタリングしている場合は、プライマリサーバとミラーサーバを実行しているコンピュータのホスト名とポート番号を、カンマで区切って指定します。次に例を示します。my-primary-server:2638,my-mirror-server:49152 が含まれます。
  - e. (省略可) **ポート**フィールドで、データベースが実行されている TCP/IP ポートを指定します。

ミラーリングシステムのプライマリデータベースをモニタリングしている場合は、**ポート**フィールドが空であることを確認してください。
  - f. (省略可) **データベース**フィールドでデータベース名を指定します。
  - g. (省略可) **サーバ**フィールドでサーバ名を指定します。

ミラーリングシステムのプライマリデータベースをモニタリングしている場合は、**サーバ**フィールドに、プライマリサーバの代替サーバ名を入力します。データベースミラーリングシステムのプライマリサーバとして動作するデータベースサーバに接続するためにクライアントで使用する名前を指定します。
  - h. (省略可) **その他**フィールドで、接続パラメータの文字列を、セミコロンで区切って指定します。
  - i. **次へ**をクリックします。
  - j. 残りの指示に従います。
  - k. **作成**をクリックします。
6. プロンプトが表示されたら、SYS\_SAMONITOR\_ADMIN\_ROLE ロールまたは DBA 権限を持つユーザのユーザ ID とパスワードを入力し、**閉じる**をクリックします。

これらのクレデンシャルは次の操作を行う場合に使用されます。

  - データベースに接続します。
  - sa\_monitor\_user という名前の新しいユーザを作成します。モニタは、sa\_monitor\_user を使用してデータベースに接続し、モニタリングします。
  - データベースをモニタリングするために sa\_monitor\_user が必要とするデータベースオブジェクトをインストールします。
  - モニタからユーザのクレデンシャルを破棄します。sa\_monitor\_user はモニタリングされるデータベースに追加されるので、そのデータベースの外部にユーザクレデンシャルを格納する必要はありません。
7. **閉じる**をクリックします。
8. モニタする Web サービスリソースを作成するオプション付きのプロンプトウィンドウが、表示されます。モニタされる Web サービスリソースを設定するには、**はい**をクリックします。それ以外の場合は、**いいえ**をクリックします。
9. **閉じる**をクリックします。

## 結果

リソースが追加され、モニタリングが自動的に開始されます。[概要ダッシュボードのリソースリスト](#)にリソースが表示されます。

## 次のステップ

オプションでリソースのダッシュボードを追加することができます。デフォルトでは、リソースの追加時にダッシュボードは追加されません。

ダッシュボードにウィジェットを追加できます。読み込み専用スケールアウトシステムおよびミラーリングシステムのトポロジを表示するには、[SQL Anywhere スケールアウトポロジウィジェット](#)を追加する必要があります。

## 関連情報

[SQL Anywhere スケールアウトポロジウィジェット \[1221 ページ\]](#)

[CSV ファイルによる複数のリソースの追加 \[1234 ページ\]](#)

[ウィジェットの管理 \[1220 ページ\]](#)

[ダッシュボードの管理 \[1223 ページ\]](#)

[モニタの起動 \[1212 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[モニタがインストールしたデータベースオブジェクトのリスト \[1283 ページ\]](#)

## 1.8.2.1.10.2 Mobile Link サーバリソースの追加

Mobile Link サーバファームを、モニタでモニタリングするリソースとして追加します。

### 前提条件

モニタの管理者である必要があります。

### コンテキスト

一度に1つずつリソースを追加したり、CSV ファイルを使用して複数のリソースを追加したりできます。

ECC 暗号化を使用する、16 より前のバージョンの Mobile Link サーバをモニタリングするには、Mobile Link サーバの起動時に、ECC を使用しないネットワークプロトコルオプションの 2 番目のセットを指定する必要があります。

モニタは ECC 暗号化をサポートしていません。ECC 暗号化を使用する、16 より前のバージョンの Mobile Link サーバをモニタリングするには、Mobile Link サーバの起動時に 2 つのポート番号を指定する必要があります。モニタで、Mobile Link サーバリソースのポート番号を指定するプロンプトが表示されたら、ECC 暗号化に関連付けられていない番号を指定してください。

## 手順

1. モニタにログインします。
2. **管理**をクリックします。
3. **リソース**をクリックします。
4. **追加**をクリックします。
5. **リソースの追加**ウィンドウで、次の手順に従います。
  - a. モニタリングするリソースのタイプをクリックします。
  - b. **次へ**をクリックします。
  - c. **名前**フィールドにリソースの名前を入力します。これはモニタに表示される名前です。**次へ**をクリックします。
  - d. **ホスト**フィールドで、Mobile Link サーバが実行されているコンピュータのホスト名または IP アドレスを指定します。現在のコンピュータを表す localhost という名前を使用できます。この値は必須です。
  - e. **次へ**をクリックします。
  - f. 残りの指示に従います。

Mobile Link の場合は、Mobile Link サーバへの接続に使用される Mobile Link ユーザ ID とパスワードを入力する必要があります。Mobile Link ユーザ ID とパスワードはモニタに格納されます。
  - g. **作成**をクリックします。
6. **閉じる**をクリックします。
7. **閉じる**をクリックします。

## 結果

リソースが追加され、モニタリングが自動的に開始されます。**概要**ダッシュボードの**リソースリスト**にリソースが表示されます。

## 次のステップ

オプションでリソースのダッシュボードを追加することができます。デフォルトでは、リソースの追加時にダッシュボードは追加されません。

ダッシュボードにウィジェットを追加できます。読み込み専用スケールアウトシステムおよびミラーリングシステムのトポロジを表示するには、**SQL Anywhere スケールアウトポロジ**ウィジェットを追加する必要があります。

## 関連情報

[SQL Anywhere スケールアウトポロジウィジェット \[1221 ページ\]](#)

[CSV ファイルによる複数のリソースの追加 \[1234 ページ\]](#)

[ダッシュボードの管理 \[1223 ページ\]](#)

[モニタの起動 \[1212 ページ\]](#)

[ウィジェットの管理 \[1220 ページ\]](#)

[Host 接続パラメータ \[82 ページ\]](#)

[モニタがインストールしたデータベースオブジェクトのリスト \[1283 ページ\]](#)

### 1.8.2.1.10.3 Mobile Link サーバファームリソースの追加

Mobile Link サーバファームをモニタリングするには、目的のリソースをモニタに追加します。

#### 前提条件

モニタの管理者である必要があります。

Mobile Link サーバファーム内のサーバはすべて、Mobile Link サーバファームに追加する前に、モニタにリソースとしてすでに追加されている必要があります。

#### コンテキスト

一度に1つずつリソースを追加したり、CSV ファイルを使用して複数のリソースを追加したりできます。

#### 手順

1. モニタにログインします。
2. [管理](#)をクリックします。
3. [リソース](#)をクリックします。
4. [追加](#)をクリックします。
5. [リソースの追加](#)ウィンドウの指示に従って、Mobile Link サーバファームをモニタリングするためのリソースを追加します。

モニタは、Mobile Link サーバファームおよびファームを構成する各サーバについてのメトリックを収集します。Mobile Link サーバファームをリソースとして追加するときは、そのファームの一部であるサーバを指定する必要があります。

6. [作成](#)をクリックします。

7. 閉じるをクリックします。
8. 閉じるをクリックします。

## 結果

リソースが追加され、モニタリングが自動的に開始されます。[概要ダッシュボードのリソースリスト](#)にリソースが表示されます。

## 次のステップ

オプションでリソースのダッシュボードを追加することができます。デフォルトでは、リソースの追加時にダッシュボードは追加されません。

## 関連情報

- [SQL Anywhere スケールアウトポロジウィジェット \[1221 ページ\]](#)
- [CSV ファイルによる複数のリソースの追加 \[1234 ページ\]](#)
- [ウィジェットの管理 \[1220 ページ\]](#)
- [ダッシュボードの管理 \[1223 ページ\]](#)
- [モニタの起動 \[1212 ページ\]](#)
- [Host 接続パラメータ \[82 ページ\]](#)
- [モニタがインストールしたデータベースオブジェクトのリスト \[1283 ページ\]](#)

### 1.8.2.1.10.5 Web サーバ、プロキシサーバ、またはホストコンピュータの 可用性をモニタするリソースを追加

Web サーバ、プロキシサーバ、ホストコンピュータの可用性をモニタする Web サービスリソースを作成し、使用不可の場合は通知を受信します。

## 前提条件

モニタの管理者である必要があります。

## 手順

1. モニタで、**管理**をクリックします。
2. **リソース**をクリックして、**追加**をクリックします。
3. **Web サービス - Web サーバ、プロキシサーバ、またはホスト**をクリックし、**リソースの追加**ウィンドウの指示に従います。
4. **作成**をクリックします。
5. **閉じる**を 2 回クリックします。

## 結果

リソースが追加され、モニタリングが開始されます。**概要ダッシュボードのリソースリスト**にリソースが表示されます。

### 例

異なるモニタのインスタンスの可用性をモニタする Web サービスリソースを作成します。

## 1.8.2.1.10.6 CSV ファイルによる複数のリソースの追加

値をカンマで区切った CSV (comma-separated values) ファイルを作成し、それをインポートすることによって、モニタに複数のリソースを追加します。

## 前提条件

モニタの管理者である必要があります。

## コンテキスト

データベースミラーリングシステムにリソースとしてプライマリデータベースを追加するときに、CSV ファイルは使用できません。その代わりに、**管理**ウィンドウを使用します。

## 手順

1. CSV ファイルを作成します。

CSV ファイルの各行には、単一のリソースについての情報が含まれています。行内のカンマで区切られた各項目は、リソースの属性を記述します。項目の順序が重要です。

オプションの項目の場合、その前にある項目（他のオプションの項目も含む）を指定する必要があります。たとえば、データベース名を指定する場合、データベースサーバ名とポート番号も指定する必要があります。暗号化タイプを指定するには、接続タイプも指定する必要があります。モニタリングの URL サフィックスを指定するには、ポート番号も指定する必要があります。

#### データベースリソースの CSV ファイルフォーマット

行内の位置	項目	説明
1	リソースタイプ	<b>sa</b> と入力して、リソースがデータベースであることを指定します。この項目は必須です。
2	リソース名	モニタでデータベースに使用するリソース名を指定します。この項目は必須です。
3	ユーザ名	データベースへの接続に使用するユーザ名を指定します。バージョン 16 以降のデータベースを追加するには、SYS_SAMONITOR_ADMIN_ROLE ロールを持つ、データベース内のユーザのユーザ名を指定する必要があります。それより前のバージョンのデータベースを追加するには、DBA 権限を持つ、データベース内のユーザのユーザ名を指定する必要があります。
4	パスワード	データベースへの接続に使用するユーザパスワードを指定します。この項目は必須です。
5	ホスト	データベースサーバを実行するコンピュータのホスト名または IP アドレスを指定します。現在のコンピュータを表す localhost という名前を使用できます。この項目は必須です。
6	ポート	データベースサーバが実行されているポート番号を指定します。デフォルトのポート番号は 2638 です。この項目はオプションです。デフォルト値は 0 です。
7	サーバ	データベースが接続されるデータベースサーバの名前を指定します。この項目はオプションです。
8	Database	データベースの名前を指定します。この項目はオプションです。

行内の位置	項目	説明
9	接続パラメータ	データベースへの接続時に使用する追加の接続パラメータを指定します。 option=value の組み合わせをセミコロンで区切ったリストで、接続パラメータを取得します。デフォルト値は "" です。この項目はオプションです。

次に、2つのデータベースリソースを含む CSV ファイルの例を示します。

```
sa,demo,DBA,sql,localhost,2638,demo17
sa,demo3,DBA,sql,localhost,49152,demo17,demo
```

### Mobile Link リソースの CSV ファイルフォーマット

行内の位置	項目名	説明
1	リソースタイプ	<b>ml</b> と入力してリソースが Mobile Link サーバであることを指定します。  この項目は必須です。
2	リソース名	モニタでの表示に使用するリソース名を指定します。この項目は必須です。
3	ユーザ名	リソースに接続するためのユーザ名を指定します。この項目は必須です。
4	パスワード	リソースに接続するためのパスワードを指定します。この項目は必須です。
5	ホスト (Mobile Link サーバ)	サーバが稼働しているコンピュータのホスト名または IP アドレスを指定します。 現在のコンピュータを表す localhost という名前を使用できます。この項目は必須です。
6	ポート (Mobile Link サーバ)	Mobile Link サーバが実行されているポート番号を指定します。Mobile Link のデフォルトポート番号は 2439 です。この項目はオプションです。デフォルト値は 0 です。
7	接続タイプ (Mobile Link サーバ)	リソースへの接続に使用する方法を指定します。次のいずれかを入力します。 <ul style="list-style-type: none"> <li>○ <i>tcpip</i> これはデフォルトです。</li> <li>○ <i>http</i></li> <li>○ <i>https</i></li> <li>○ <i>tls</i></li> </ul> この項目はオプションです。

行内の位置	項目名	説明
8	暗号化タイプ (Mobile Link サーバ)	<p>接続の暗号化に使用するメソッドを指定します。次のいずれかを入力します。</p> <p><b>N</b> 暗号化なし。これはデフォルトです。</p> <p><b>R</b> RSA 暗号化。</p> <p><b>F</b> FIPS 認定 RSA 暗号化</p> <p>この項目はオプションです。</p>
9	接続パラメータ (Mobile Link サーバ)	<p>データベースへの接続時に使用する追加の接続パラメータを指定します。option=value の組み合わせをセミコロンで区切ったリストで、接続パラメータを取得します。この項目はオプションです。</p>

次に、2 つの Mobile Link サーバリソースを含むインポートファイルの例を示します。

```
ml,demo2,DBA,sql,localhost
ml,demo4,DBA,sql,localhost,0,tcPIP,N
```

#### Mobile Link サーバファームリソースの CSV ファイルフォーマット

行内の位置	項目名	説明
1	リソースタイプ	<p><b>mf</b> と入力してリソースが Mobile Link サーバファームであることを指定します。</p> <p>この項目は必須です。</p>
2	リソース名	<p>モニタでの表示に使用するリソース名を指定します。この項目は必須です。</p>
3	Mobile Link サーバ名	<p>Mobile Link サーバファームに含める Mobile Link サーバリソースを指定します。Mobile Link サーバリソースをカンマで区切られたリストとしてリストします。この項目は必須です。</p>

次に、2 つの Mobile Link サーバリソースと 1 つの Mobile Link サーバファームリソースを含むインポートファイルの例を示します。

```
ml,demo2,DBA,sql,localhost
ml,demo4,DBA,sql,localhost,0,tcPIP,N
mf,collection_demo,demo2,demo4
```

#### Web サービスリソースの CSV ファイルフォーマット

行内の位置	項目名	説明
1	リソースタイプ	<b>ws</b> と入力してリソースが Web サービスであることを指定します。 この項目は必須です。
2	リソース名	モニタでの表示に使用するリソース名を指定します。この項目は必須です。
3	サービスタイプ	モニタするサービスのタイプを指定します。使用可能な値は、次のとおりです。 <b>W</b> Web サーバ <b>P</b> プロキシ <b>H</b> ホスト
4	Use Proxy	Web サービスへの接続を試みる時にプロキシを使用するには <b>True</b> を指定します。 デフォルトは <b>False</b> です。この項目はオプションです。
5	プロキシホスト	Use Proxy が <b>True</b> に設定されている場合、使用するプロキシのホストを指定します。空白のままにすると、Web ブラウザが使用するデフォルトのプロキシホストが使用されます。この項目はオプションです。
6	プロキシポート	Use Proxy が <b>True</b> に設定されている場合、使用するプロキシのポートを指定します。空白のままにすると、Web ブラウザが使用するデフォルトのプロキシポートが使用されます。この項目はオプションです。
7	説明	Web サービスの説明を指定します。この項目はオプションです。

2. モニタにログインします。
3. **管理**をクリックします。
4. **リソース**をクリックします。
5. **インポート**をクリックします。
6. CSV ファイルを検索し、**開く**をクリックします。

CSV ファイルのリソースがモニタに追加され、モニタリングが開始されます。

7. **閉じる**をクリックします。
8. **閉じる**をクリックします。

## 結果

概要ダッシュボードのリソースリストにインポートされたリソースが追加されます。

## 関連情報

[データベースリソースの追加 \[1228 ページ\]](#)

[Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

### 1.8.2.1.10.7 リソースのモニタリングの開始

リソースの起動をモニタリングする条件がいくつかあります。

- モニタの起動時 (自動的に開始)。モニタを起動すると、デフォルトで、モニタはリソースに接続し、指定されているすべてのリソースのメトリックを収集します。
- モニタ管理者がリソースを追加したとき (自動的に開始)。リソースが追加されると、モニタは、リソースへの接続とモニタリングの開始を試みます。
- ブラックアウト期間が終了したとき (自動的に開始)。モニタは、リソースへの接続とモニタリングの再開を自動的に試みます。
- モニタ管理者が、[管理ウィンドウを開き](#)、[リソースをクリック](#)し、リストからリソースを選択し、[起動](#)をクリックしたとき。

## 関連情報

[データベースリソースの追加 \[1228 ページ\]](#)

[Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

### 1.8.2.1.10.8 リソースのモニタリングの停止

モニタでメトリックを収集しない場合は、リソースのモニタリングを停止する必要があります。

たとえば、目的のリソースが使用不可のときにモニタリングを停止します。そうしないと、リソースが使用可能になるまで警告を受信し続けることとなります。デフォルトリソースであるモニタを除き、リソースのモニタリングはいつでも停止できます。

リソースのモニタリングを停止すると、モニタは次のように動作します。

- 該当するリソースのメトリックの収集を停止します。
- 該当するリソースに関する警告の発行を停止します。

リソースのモニタリングを停止するには、次の 2 つの方法があります。

定期的に繰り返されるブラックアウト期間をスケジュールする

この方法は、次の状況が当てはまる場合に適しています。

- リソースのモニタリングを繰り返し停止する必要がある場合。たとえば、毎月末に定期メンテナンスを行う場合です。
- リソースが使用不可になる期間があらかじめわかっている場合。たとえば、定期メンテナンスに 4 時間かかることがわかっている場合です。
- モニタリングを自動的に再開する必要がある場合。ブラックアウト期間が終わると、モニタはリソースへの再接続を行い、データの収集を続けます。

この方法を使用するには、ブラックアウト期間を設定して、指定した時刻にモニタによるモニタリングが自動的に停止されるようにする必要があります。

#### モニタリングの手動停止

この方法は、次の状況が当てはまる場合に適しています。

- まれにしか発生しないタスクや 1 回限りのタスクのためにモニタリングを停止する必要がある場合。たとえば、リソースが実行されているコンピュータを特別なメンテナンスのためにオフラインにする必要があり、モニタリングを停止する必要がある場合です。
- モニタリングを後で再開するときに立ち会える場合。リソースが手動で停止されている場合、モニタはモニタリングの再開を待機します。

リソースのモニタリングを永久に停止する場合は、該当するリソースをモニタから削除できます。

このセクションの内容:

[ブラックアウトの設定 \[1240 ページ\]](#)

モニタを使用してブラックアウトを設定します。

[リソースのモニタリングの停止 \[1242 ページ\]](#)

リソースのモニタリングを停止します。

## 関連情報

[リソースの削除 \[1244 ページ\]](#)

### 1.8.2.1.10.8.1 ブラックアウトの設定

モニタを使用してブラックアウトを設定します。

## 前提条件

モニタの管理者である必要があります。

## コンテキスト

ブラックアウトはモニタがメトリックの収集を行わない時間で、該当リソースのローカル時間に対応します。

ブラックアウトの完了によって、モニタのリソースへの再接続とデータの収集の継続が要求されます。ブラックアウト期間が終わると、モニタはリソースに再度接続し、データの収集を続けます。

## 手順

1. モニタにログインします。
2. **▶ ツール ▶ 管理 ▶**をクリックします。
3. **リソース**をクリックします。
4. 目的のリソースをクリックし、**設定**をクリックします。
5. **ブラックアウト**をクリックします。
6. **新規**をクリックします。
7. **新規ブラックアウト期間**ウィンドウで、ブラックアウトの日付と時刻 (24 時間表記) を指定します。

この日時は、対象リソースであるデータベースまたは Mobile Link サーバが存在しているコンピュータのローカル時間に対応します。

8. **保存**をクリックします。
9. **保存**をクリックします。
10. **閉じる**をクリックします。
11. **閉じる**をクリックします。

## 結果

ブラックアウトが設定されます。

## 関連情報

[時刻の表示方法の概要 \[1224 ページ\]](#)

[データベースリソースの追加 \[1228 ページ\]](#)

[Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

[リソースのモニタリングの停止 \[1242 ページ\]](#)

## 1.8.2.110.8.2 リソースのモニタリングの停止

リソースのモニタリングを停止します。

### 前提条件

モニタの管理者である必要があります。

### 手順

1. モニタにログインします。
2. **ツール** > **管理** をクリックします。
3. **リソース** をクリックします。
4. 目的のリソースをクリックし、**停止** をクリックします。

#### **i** 注記

Mobile Link サーバファームのモニタリングを停止すると、ファームを構成する Mobile Link サーバのモニタリングは停止します。

5. **閉じる** をクリックします。

### 結果

指定したリソースのモニタリングが停止します。

### 関連情報

[データベースリソースの追加 \[1228 ページ\]](#)

[Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

[ブラックアウトの設定 \[1240 ページ\]](#)

## 1.8.2.1.10.9 モニタでモニタリングするデータベースリソースの修復

データベースリソースを修復して、モニタによるモニタリングを再開します。

### 前提条件

モニタの管理者である必要があります。

バージョン 16 以降リソースを修復するたびに、SYS\_SAMONITOR\_ADMIN\_ROLE ロールを持つデータベース内のユーザのユーザ ID とパスワードを指定する必要があります。それより前のバージョンのデータベースでは、修復するたびに、DBA 権限を持つ、データベース内のユーザのユーザ ID とパスワードを指定する必要があります。

### コンテキスト

リソースを修復すると、該当リソースのデータベースオブジェクトが再インストールされます。修復できるのはデータベースリソースのみです。デフォルトリソースである *SQL Anywhere モニタ* という名前のモニタを修復することはできません。

### 手順

1. モニタにログインします。
2. **管理**をクリックします。
3. **リソース**をクリックします。
4. 修復するデータベースリソースをクリックします。
5. 目的のリソースが現在モニタリングされている場合は、**停止**をクリックします。
6. **修復**をクリックします。
7. プロンプトが表示されたら、データベースのユーザ ID とパスワードを入力します。このユーザクレデンシャルはデータベースとの接続に使用され、後でモニタから破棄されます。
8. **修復**をクリックします。
9. **OK**をクリックします。
10. **開始**をクリックして、リソースのモニタリングを再開します。
11. **閉じる**をクリックします。

### 結果

リソースが修復され、モニタリングオプションは変更されないままになります。

## 関連情報

[モニタがインストールしたデータベースオブジェクトのリスト \[1283 ページ\]](#)

### 1.8.2.1.10.10 リソースの削除

リソースを削除して、モニタがそのリソースをモニタリングすることを永久に停止し、そのリソースについて収集したメトリックを破棄します。

#### 前提条件

モニタの管理者である必要があります。

#### コンテキスト

リソースを削除する場合は、使用されなくなったリソースなど、モニタリングが不要になったことが確実なリソースに限定してください。*SQL Anywhere* モニタリソースは削除できません。

データベースリソースを削除しても、データベースにインストールされたモニタリングオブジェクトは削除されません。

#### 手順

1. モニタにログインします。
2. [管理](#)をクリックします。
3. [リソース](#)をクリックします。
4. リソースをクリックします。
5. [削除](#)をクリックします。
6. [はい](#)をクリックします。

#### **i** 注記

Mobile Link サーバファームを削除すると、ファームを構成する Mobile Link サーバを削除するよう要求されます。指定された Mobile Link サーバを削除する場合は、[はい](#)をクリックします。Mobile Link サーバファームのみを削除し、指定された Mobile Link サーバファームを削除しない場合は、[いいえ](#)をクリックします。

7. [閉じる](#)をクリックします。

## 結果

リソースが削除されます。

## 関連情報

[リソースのモニタリングの停止 \[1239 ページ\]](#)

[リソースデータベースからのモニタリングオブジェクトの削除 \[1284 ページ\]](#)

### 1.8.2.1.11 メトリック

モニタは、データベース、Mobile Link サーバ、および Mobile Link サーバファームから、次のような多くのメトリックを収集して保存します。

これには以下が含まれますが、そのみに限定されません。

- リソースが実行中かどうか。
- リソースを実行しているコンピュータが、正常に実行されているかどうか、およびネットワークに接続されているかどうか。
- リソースが要求を受信して処理しているかどうか。
- 実行時間が長いクエリ、ブロックされたユーザなど、明らかな問題がないかどうか。
- 所定の期間内に Mobile Link サーバが実行した同期の回数。
- バックエンドサーバが HTTP 要求の処理に要した平均時間。

このセクションの内容:

[SQL Anywhere データベースリソースのカスタムメトリックの定義 \[1246 ページ\]](#)

メトリックをカスタマイズして、リソースからデータを収集し、モニタでそのメトリックに関する警告を受信します。

[データベースリソースのメトリックのリスト \[1249 ページ\]](#)

モニタはデータベースから多くのメトリックを収集し格納します。

[Mobile Link サーバリソースのメトリックのリスト \[1252 ページ\]](#)

モニタは Mobile Link サーバリソースから多くのメトリックを収集し格納します。

[Mobile Link サーバファームリソースのメトリックのリスト \[1257 ページ\]](#)

モニタは Mobile Link サーバファームリソースの多くのメトリックを収集し格納します。

[リソースの収集間隔の設定 \[1263 ページ\]](#)

メトリックがモニタによって収集される頻度を変更します。

[メトリックのエクスポート \[1264 ページ\]](#)

グラフや表を関連付けられたメトリックを XML ファイルにエクスポートします。

[メトリックの再表示 \[1265 ページ\]](#)

再表示間隔を変更します。この間隔は、リソースの収集間隔とは異なります。

## 1.8.2.111.1 SQL Anywhere データベースリソースのカスタムメトリックの定義

メトリックをカスタマイズして、リソースからデータを収集し、モニタでそのメトリックに関する警告を受信します。

### 前提条件

リソースデータベースでファンクションを作成するには、CREATE PROCEDURE または CREATE ANY OBJECT のいずれかが必要です。

EXECUTE 権限をモニタユーザ sa\_monitor\_user に付与するには、EXECUTE が必要です。

### コンテキスト

カスタムメトリックによって、ユーザはデータベースリソースのモニタが収集するデータを定義することができます。ユーザが指定した警告基準に基づく警告が "警告リスト" ウィジェットに表示されます。

SQL Anywhere モニタは、収集期間ごとに SQL Anywhere リソースの値ファンクションを呼び出して、カスタムメトリックデータを記録します。

### 手順

1. リソースデータベースでの操作:
  - a. ユーザ定義ファンクションを作成します。
  - b. sa\_monitor\_user EXECUTE 権限をこのファンクションに付与します。
2. モニタでの操作:
  - a. ツールウィンドウ枠で**管理**をクリックします。目的のリソースを選択し、**設定**をクリックします。
  - b. **カスタムメトリック**タブをクリックします。
  - c. **新規**をクリックします。
  - d. ユーザ定義ファンクションの設定を指定します。
  - e. **OK**をクリックします。
  - f. **保存**をクリックします。
  - g. **閉じる**をクリックします。

### 結果

カスタムメトリックが作成されます。メトリックの警告スレッシュホールドを指定するには、**警告スレッシュホールド**タブをクリックして、**カスタムメトリック**をクリックします。指示に従います。

## 例

1. データベースユーザは以下を行うことができます。
  - テーブルの注文数をモニタします。
  - 全注文数が 100000 を超える場合、警告を発行します。
1. リソースデータベースで、ユーザは以下のユーザ定義ファンクションを作成します。

```
CREATE FUNCTION "dba"."get_order_count" ()
  RETURNS INT
  NOT DETERMINISTIC
  BEGIN
    DECLARE "currentCount" INT;
    SELECT COUNT(*) INTO currentCount FROM SalesOrders;
    RETURN "currentCount";
  END;
```

2. モニタで、ユーザはカスタムメトリックを以下の設定にします。

表 1: カスタムメトリックの設定

設定	値
Name	<b>OrderCount</b>
Display units	<b>Orders</b>
Minimum	<b>0</b>
Maximum	
Data type	<b>Integer</b>
Value function owner	<b>dba</b>
Value function name	<b>get_order_count</b>

3. ユーザはスレッシュホールド値が 100000 の警告を設定します。  
OrderCount メトリックのグラフを表示するように、モニタでウィジェットが設定されます。
2. 別のユーザが以下を行うこともできます。
    - トランザクションが完了までに 5 秒以上かかった場合、警告を受信します。
  1. リソースデータベースで、ユーザは以下のテーブルとイベントを作成します。

```
CREATE TABLE "dba"."order_transaction_time" (
  "pkey" INT PRIMARY KEY DEFAULT AUTOINCREMENT,
  "transaction_time" TIMESTAMP NOT NULL,
  "number_of_seconds" INTEGER NOT NULL
);
CREATE EVENT
"DBA"."truncateData" SCHEDULE "truncateSchedule" START TIME '23:59' EVERY
24 HOURS
HANDLER
BEGIN
  // delete data older than 2 weeks
  DELETE FROM order_transaction_time
  WHERE DATEDIFF(second,transaction_time, current time) > 1209600;
END;
```

トランザクションが完了すると、order\_transaction\_time テーブルにエントリが追加され、トランザクションが完了した時刻とトランザクションの実行にかかった時間が記録されます。

- ユーザはユーザ定義ファンクション `average_order_time_five_min_average` を作成して、このファンクションの EXECUTE 権限をモニタユーザ `sa_monitor_user` に付与します。このファンクションによって、最後の 5 分間の平均トランザクション時間が戻されます。

```
CREATE FUNCTION "dba"."average_order_time_five_min_average"()
RETURNS INTEGER
NOT DETERMINISTIC
BEGIN
  DECLARE "max_order_time" INTEGER;
  // return average transaction time over the last 5 minutes
  SELECT IF AVG(number_of_seconds) IS NOT NULL
    THEN CAST(AVG(number_of_seconds) AS INTEGER )
  ELSE
    0
  ENDIF
  INTO max_order_time
  FROM "DBA"."order_transaction_time"
  WHERE DATEDIFF(second,transaction_time,current time) < 300;
  RETURN max_order_time;
END
GRANT EXECUTE ON max_order_time to sa_monitor_user.
```

- モニタで、ユーザは平均注文時間メトリックを作成します。

表 2: カスタムメトリックの設定

設定	値
Name	<b>Average order time</b>
Data type	<b>Integer</b>
Display units	<b>Seconds</b>
Minimum	<b>0</b>
Maximum	
Value function owner	<b>dba</b>
Value function name	<b>order_time_five_min_average</b>

- ユーザはスレッシュホールド値が 5 の警告を設定します。  
平均トランザクション時間が 5 秒を超過すると、警告がトリガされます。

## 関連情報

[警告スレッシュホールドの指定 \[1278 ページ\]](#)

## 1.8.2.1.11.2 データベースリソースのメトリックのリスト

モニタはデータベースから多くのメトリックを収集し格納します。

メトリック名	説明
アクティブな HTTP 要求	HTTP 要求をアクティブに処理している HTTP 接続数を返します。 応答を送信した HTTP 接続は含まれません。
アクティブな HTTPS 要求	HTTPS 要求をアクティブに処理しているセキュア HTTPS 接続数を返します。応答を送信した HTTPS 接続は含まれません。
監視サーバステータス	次のいずれかの値を返します。  接続しました。  監視サーバはプライマリサーバに接続していません。 切断しました。  監視サーバはプライマリサーバに接続していません。 0  この構成には監視サーバがありません。
ダーティキャッシュ	ダーティな (書き込みが必要な) キャッシュページの数を返します。
固定されたキャッシュ	固定キャッシュページ数を返します。
キャッシュ置換	キャッシュ内の置換されたページ数を返します。
キャッシュサイズ	現在のキャッシュサイズ (キロバイト単位) を返します。
接続数	データベースとの現在の接続数を返します。
CPU 使用率	データベースサーバによって使用された CPU 時間の割合を返します。
データベース名	データベースの名前を返します。
ディスク読み取り	データがディスクから読み込まれている速度を示します (キロバイト数/秒単位)。この値は DiskRead プロパティを基に計算されます。
ディスク書き込み	データがディスクに書き込まれている速度を返します (キロバイト数/秒単位)。この値は DiskWrite プロパティを基に計算されます。
ファイルサイズ	メインデータベースファイルのサイズを返します。
空きディスク領域	ディスクの空き領域のサイズを返します。
ホスト	データベースサーバを実行しているコンピュータの名前を返します。 通常は、コンピュータのホスト名です。

メトリック名	説明
HTTP 接続	データベースサーバで現在開いている HTTP 接続数を返します。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。
HTTP セッション	データベースサーバ内のアクティブおよび休止状態の HTTP セッション数を返します。
HTTPS 接続	データベースサーバで現在開いている HTTPS 接続数を返します。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。
最後の確認時刻	モニタがリソースのデータを最後に収集した時刻を返します。
ライセンスの種類	ライセンスタイプを返します。ネットワークシート (per-seat) または CPU ベースのいずれかです。
ライセンスを持つ会社	ライセンスされた会社の名前を返します。
ライセンスされたシート	ライセンスされたシートまたはプロセッサの数を返します。
ライセンスを持つユーザ	ライセンスされたユーザの名前を返します。
長いクエリ情報	実行時間が長いクエリのスレッシュホールドを超えたクエリを返します。
メインヒープページ	グローバルサーバデータ構造に使用されているページ数を返します。
最大キャッシュサイズ	許容最大キャッシュサイズを返します。
ミラーモード	データベースのミラーリングが使用されていない場合、このデータベースではミラーリングが有効になっていませんと表示されます。ミラーリングが有効の場合、SET MIRROR OPTION 文で指定された同期モードが同期の場合は同期、それ以外の場合は非同期と表示されます。
ミラーステータス	次のいずれかの値を返します。  同期しています  ミラーサーバが接続されていないか、プライマリサーバのログページの読み込みが完了していません。同期実行モードが非同期である場合にも、この値が返されます。  同期しました  ミラーサーバが接続され、プライマリサーバ上でコミットされたすべての変更が反映されています。
オペレーティングシステム	ソフトウェアが実行されているオペレーティングシステムを返します。

メトリック名	説明
オペレーティングシステムのバージョン	ソフトウェアを実行しているオペレーティングシステムの名前を示します。この名前には、ビルド番号やサービスパックなどの情報も含まれています。
パートナーステータス	次のいずれかの値を返します。 接続しました ミラーサーバはプライマリサーバに接続しています。 切断しました ミラーサーバはプライマリサーバに接続されていません。
ピークキャッシュサイズ	現在のセッションでキャッシュが到達した最大値 (キロバイト単位) を返します。
プロセッサのアーキテクチャ	プロセッサタイプを表す文字列を返します。
処理されたクエリ	クエリの処理率 (クエリ/秒) を返します。
シート数	ネットワークサーバに接続しているユニークなクライアントネットワークアドレスの数を返します。
サーバの言語	ロケール言語を示します。これは、データベースサーバのユーザによる使用が想定されている言語です。
サーバ名	現在の接続におけるデータベースサーバの名前を返します。
サーバ時間オフセット	モニタが実行されているコンピュータの時刻と、モニタデータの表示に使用しているコンピュータの時刻の差異が記録されます。
サーバタイプ	モニタリングされるデータベースサーバのタイプ (パーソナルまたはネットワーク) を返します。
サーバのバージョン	実行中のソフトウェアのバージョンを返します。
開始時刻	データベースが起動された時刻を返します。
総ディスク領域	ディスクのサイズを返します。
次の時刻から使用不可	リソースが使用不可になった時間を返します。
スケジュールされていない要求	スケジュールされていない要求の数を返します。
未送信のエラーレポート	該当データベースについて未送信のエラーレポートの数を返します。ソフトウェアがクラッシュしたときに、エラーレポートが送信されません。

### 1.8.2.111.3 Mobile Link サーバリソースのメトリックのリスト

モニタは Mobile Link サーバリソースから多くのメトリックを収集し格納します。

メトリック名	説明
同期を認証中	現在サーバで、ユーザ認証フェーズにある要求の数を示します。
キャッシュされているリモートタスク要求	キャッシュされたリモートタスク要求数を示します。
コマンドプロセスサステージ長	同期タスクのキューの長さを示します。
コミット率	コミットが発生する頻度を示します。
コミット	コミットの合計数を示します。
同期の完了率	サーバの同期の正常完了率を示します。
接続しているクライアント	接続しているクライアント数を示します。
接続が閉じた割合	TCP 接続が閉じた割合を示します。
接続が開いた割合	TCP 接続が開いた割合を示します。
接続が拒否された割合	TCP 接続がサーバによって拒否された割合を示します。
統合データベースタイプ	統合データベースのタイプを示します。たとえば、 <i>SQL Anywhere</i> のように記述します。
統合データベースのバージョン	統合データベースのバージョンが表示されます。
CPU 合計時間	Mobile Link サーバによって使用される CPU 時間の量 (マイクロ秒) を示します。
CPU 使用率	Mobile Link サーバによって使用された CPU 時間の割合を示します。
CPU	使用中の CPU の数を示します。
現在の TCP 接続	この Mobile Link サーバで現在使用中の TCP 接続の数を示します。
使用中のデータベース接続	使用中のデータベース接続の数を示します。
データベースワークスレッドの数	データベースワークスレッドの数を示します。
同期をダウンロード中	ダウンロードの準備、ダウンロードのフェッチ、ダウンロード確認の待機、同期の終了の各フェーズにある現在の同期数を示します。
ドライバ名	統合データベースのドライバの名前を示します。
ドライバのバージョン	統合データベースのドライバのバージョンを示します。

メトリック名	説明
エラー	エラーの合計数を示します。
同期の失敗率	サーバの同期の失敗率を同期数/秒の単位で示します。
失敗した同期	失敗した同期の合計数を示します。
ファイル転送	現在接続されている mfiletransfer 数を示します。
終了した同期の割合	完了した同期の割合を示します。
Mobile Link キャッシュ用ディスク空き領域	Mobile Link キャッシュ用の、テンポラリディスク上の空きディスク領域をバイト単位で示します。
ハートビートステージ長	同期以外の定期的なタスクのキューの長さを示します。
ホスト	Mobile Link サーバを実行しているコンピュータの名前を示します。通常は、コンピュータのホスト名です。
統合を使用できますか?	Mobile Link サーバが統合データベースに接続できない場合に <i>True</i> を示します。
統合データベースはクラスタ化されているか?	Mobile Link サーバでクラスタ化された統合データベースが使用されている場合は、 <i>True</i> が表示されます。
プライマリサーバはわかっているか?	サーバでプライマリサーバが認識されている場合は、 <i>Yes</i> と表示されます。サーバでプライマリサーバが認識されていないか、プライマリサーバがどれであってもかまわない場合は、 <i>No</i> が表示されます。
プライマリサーバであるか?	そのサーバがプライマリサーバかである場合は、 <i>Yes</i> が表示されます。
最後の確認時刻	モニタリソースのデータを最後に収集した時刻を示します。
ライセンスを持つ会社	ライセンスされた会社の名前を示します。
ライセンスを持つユーザ	ライセンスされたユーザの名前を示します。
リスナ	現在接続されているリスナの数を示します。
最大キャッシュサイズ	Mobile Link サーバで使用される最大キャッシュサイズを示します。
最大クライアント数	クライアントの最大数を示します。
データベース接続の最大数	mIsrv17 の -cn オプションまたは -w オプションで設定されたデータベース接続の最大数を示します。
最大同期時間	同期が発生するまでにかかった最大時間を示します。最も古いアクティブな同期の経過時間 (マイクロ秒)。
最大 TCP 接続数	mIsrv17 の -nc オプションで設定された TCP 接続の最大数を示します。
許可されるアップロードの最大数	データベースへの同時アップロードの最大数を示します。

メトリック名	説明
接続の最大待ち時間	アクティブな同期でデータベースを待機していた最長時間を示します。
使用メモリ	使用中の RAM のバイト数を示します。サーバが Windows 以外のプラットフォームで実行されている場合は、追跡されたメモリの値を示します。
モニタ	現在接続されているモニタおよび Mobile Link プロファイラの数を示します。
Notifier ステージ長	Notifier ワークキューの長さを示します。
オペレーティングシステム	ソフトウェアを実行しているオペレーティングシステムを示します。
ストリームスタック内のページ	ネットワークストリームで保持されているページの数を示します。
ロックされているページ	メモリにロードされているキャッシュページ数を示します。
スワップインしたページ	ディスクから読み込まれたページの合計数を示します。
ページのスワップイン率	ページがディスクから読み込まれる速度を示します。
スワップアウトしたページ	ディスクへスワップされたページの合計数を示します。
ページのスワップアウト率	ページがディスクへスワップされる頻度を示します。
使用されているページ	使用されたキャッシュページ数。ディスクにスワップされたページが含まれるため、キャッシュサイズよりも大きくなる場合があります。
クライアントの接続率	現在接続されているクライアントの割合を示します。このメトリックは、 <a href="#">接続しているクライアント数の値をクライアントの最大数の値で除算し、その結果に 100 を乗算することによって得られます。</a>
接続の使用率	現在使用されている接続の割合を示します。
ストリームスタック内のページ (%)	ネットワークストリームで保持されているページの割合を示します。
ロックされているページ (%)	ロックされているページの割合を示します。
使用済みページ (%)	使用されているページの割合を示します。ディスクにスワップされたページが含まれるため、キャッシュサイズよりも大きくなる場合があります。
ping	現在接続されている、ping を実行するクライアント数を示します。
プロセッサのアーキテクチャ	プロセッサタイプを表す文字列を示します。
未加工の TCP ステージ長	ネットワークワークキューの長さを示します。
要求	処理されている有効な要求の数を示します。

メトリック名	説明
アップロード適用フェーズの要求	現在、アップロードの適用フェーズにある同期の数を示します。
ユーザ認証フェーズの要求	現在、ユーザ認証フェーズにある同期の数を示します。
同期開始フェーズの要求	現在、同期開始フェーズにある同期の数を示します。
ダウンロード確認の接続フェーズの要求	現在、ダウンロード確認の接続フェーズにある同期の数を示します。
接続フェーズの要求	現在、接続フェーズにある同期の数を示します。
同期終了フェーズの要求	現在、同期終了フェーズにある同期の数を示します。
ダウンロードフェッチフェーズの要求	現在、ダウンロードのフェッチフェーズにある同期の数を示します。
ダウンロード確認の DB ワークの取得フェーズの要求	現在、確認の DB ワークの取得フェーズにある同期の数を示します。
DB ワークの取得フェーズの要求	現在、DB ワークの取得フェーズにある同期の数を示します。
非ブロッキングダウンロード確認フェーズの要求	現在、非ブロッキングダウンロード確認フェーズにある同期の数を示します。
ダウンロード準備フェーズの要求	現在、ダウンロードの準備フェーズにある同期の数を示します。
アップロードの受信フェーズの要求	現在、アップロードの受信フェーズにある同期の数を示します。
ダウンロード送信フェーズの要求	現在、ダウンロードの送信フェーズにある同期の数を示します。
同期要求フェーズの要求	現在、同期要求フェーズにある同期の数を示します。
ダウンロード確認の待機フェーズの要求	現在、ダウンロード確認の待機フェーズにある同期の数を示します。
ロールバック率	ロールバックが発生する頻度を示します。
ロールバック	ロールバックの合計数を示します。
ローがダウンロードされました	リモートに送信されたローの合計数を示します。
ローのダウンロード率	ローがリモートに送信される頻度を示します。
ローが更新されました	リモートから受信したローの合計数を示します。
ローのアップロード率	リモートからローを受信する頻度を示します。
RTNotifier ライトウェイトポーリングヒット率	リモートタスク要求を最低 1 つ検出したリモートタスクエージェントのライトウェイトポーリング数の率を示します。
RTNotifier ライトウェイトポーリングのヒット	リモートタスク要求を最低 1 つ検出したリモートタスクエージェントのライトウェイトポーリングの数を示します。

メトリック名	説明
RTNotifier ライトウェイトポーリングミス率	リモートタスク要求を検出しなかったリモートタスクエージェントのライトウェイトポーリング数の率を示します。
RTNotifier ライトウェイトポーリングのミス	リモートタスクを検出しなかったリモートタスクエージェントのライトウェイトポーリングの数を示します。
RTNotifier ライトウェイトポーリング率	リモートタスクエージェントのライトウェイトポーリング数の率を示します。
RTNotifier ライトウェイトポーリング	リモートタスクエージェントのライトウェイトポーリング数を示します。
サーバキャッシュサイズ	サーバキャッシュのサイズを示します。
サーバ名	接続されているサーバの <code>-zs</code> オプションで指定された Mobile Link サーバの名前を示します。デフォルト値は <code>&lt;default&gt;</code> です。
サーバ時間オフセット	モニタが実行されているコンピュータの時刻と、モニタデータの表示に使用しているコンピュータの時刻の差異を示します。
サーバによって追跡されているメモリ使用率	サーバによって割り付けられたメモリの量を示します。このメトリックは、MEMORY_USED メトリックを使用できない、Windows 以外のシステムに対して使用します。Microsoft Windows システムでは、精度を上げるために MEMORY_USED メトリックを使用してください。
開始時刻	Mobile Link サーバが起動された時刻を示します。
同期の開始率	同期が開始される頻度を示します。
ストリームステージ長	高レベルのネットワーク処理キューの長さを示します。
成功した同期	成功した同期の合計数を示します。
同期エラー率	同期エラーの割合を示します。
同期警告率	サーバに対する同期警告の割合を示します。
同期	現在接続されているデータ同期数を示します。 NUM_CONNECTED_SYNCS を参照してください。
同期がブロックされました	データベース接続を現在待機している同期の数を示します。
同期が終了しました	終了した同期の合計数を示します。
同期を開始しました	開始された同期の合計数を示します。
読み取られた TCP バイト	これまでに読み込まれたバイトの合計数を示します。
TCP バイト読み取り率	ネットワークからバイトが読み込まれる速度を示します。

メトリック名	説明
書き込まれた TCP バイト	これまでに書き込まれたバイトの合計数を示します。
TCP バイト書き込み率	ネットワークにバイトが書き込まれる速度を示します。
TCP 接続が閉じられました	これまでに閉じられた接続の合計数を示します。
TCP 接続が開かれました	これまでに開かれた接続の合計数を示します。
TCP 接続が拒否されました	これまでに拒否された接続の合計数を示します。
時間が指定された作業のステージ長	動的キャッシュワークキューの長さを示します。
次の時刻から使用不可	リソースが使用不可になった時間を示します。
不明な接続済みクライアント	接続しているクライアントのうち、オリジンが不明なクライアントの数を示します。
未送信のエラーレポート	該当サーバについて未送信のエラーレポートの数を示します。ソフトウェアがクラッシュしたときに、エラーレポートが送信されます。
使用中のアップロード接続	現在使用中のアップロード接続の数を示します。
同期を更新中	アップロードフェーズにある同期の数を示します。
バージョン	実行中のソフトウェアのバージョンを示します。
VM メモリ使用率	付加された VM で使用されるメモリの量を示します。
データベース接続の待機中	データベース接続を待機している要求数。
警告	警告の合計数を示します。

## 1.8.2.1.11.4 Mobile Link サーバファームリソースのメトリックのリスト

モニタは Mobile Link サーバファームリソースの多くのメトリックを収集し格納します。

メトリック名	スコープ	説明
同期を認証中	平均	現在サーバで、ユーザ認証フェーズにある要求の数を示します。
バイト読み取り率	平均	ネットワークからバイトが読み込まれる頻度を示します。
バイト書き込み率	平均	ネットワークにバイトが書き込まれる速度を示します。

メトリック名	スコープ	説明
キャッシュされているリモートタスク要求	平均	キャッシュされたリモートタスク要求数を示します。
コマンドプロセスステージ長	平均	同期タスクのキューの長さを示します。
コミット率	平均	コミットが発生する頻度を示します。
コミット	平均	コミットの合計数を示します。
同期の完了率	合計	サーバの同期の正常完了した頻度を示します。
接続しているクライアント	平均	接続しているクライアント数を示します。
接続が閉じた割合	平均	TCP 接続が閉じた頻度を示します。
接続が開いた割合	平均	TCP 接続が開いた頻度を示します。
接続が拒否された割合	平均	接続がサーバによって拒否された頻度を示します。
CPU 合計時間	合計	Mobile Link サーバによって使用される CPU 時間の量 (マイクロ秒) を示します。
CPU 使用率	平均	Mobile Link サーバによって使用された CPU の平均量を示します。
現在の TCP 接続	合計	この Mobile Link サーバで現在使用中の TCP 接続の数を示します。
使用中のデータベース接続	合計	使用中のデータベース接続の数を示します。
同期をダウンロード中	平均	ダウンロードの準備、ダウンロードのフェッチ、ダウンロード確認の待機、同期の終了の各フェーズにある現在の同期数を示します。
エラー	平均	エラーの平均数を示します。
同期の失敗率	合計	サーバの失敗した同期の頻度を示します。
失敗した同期	平均	失敗した同期の合計数を示します。
ファイル転送	合計	現在接続されている mfiletransfer 数を示します。
終了した同期の割合	合計	完了した同期の頻度を示します。
Mobile Link キャッシュ用ディスク空き領域	平均	Mobile Link キャッシュ用の、テンポラリディスク上の空きディスク領域をバイト単位で示します。

メトリック名	スコープ	説明
ハートビートステージ長	平均	同期以外の定期的なタスクのキューの長さを示します。
リスナ	合計	現在接続されているリスナの数を示します。
最大同期時間	最大	同期が発生するまでにかかった最大時間を示します。最も古いアクティブな同期の経過時間 (マイクロ秒)。
接続の最大待ち時間	最大	アクティブな同期でデータベースを待機していた最長時間を示します。
使用メモリ	平均	使用中の RAM のバイト数を示します。サーバが Windows 以外のプラットフォームで実行されている場合は、追跡されたメモリの値を示します。
モニタ	合計	現在接続されているモニタおよび Mobile Link プロファイラの数を示します。
Notifier ステージ長	平均	Notifier ワークキューの長さを示します。
ストリームスタック内のページ	平均	ネットワークストリームで保持されているページの数を示します。
ロックされているページ	平均	メモリにロードされているキャッシュページ数を示します。
スワップインしたページ	平均	ディスクから読み込まれたページの合計数を示します。
ページのスワップイン率	平均	ページがディスクから読み込まれる速度を示します。
スワップアウトしたページ	平均	ディスクへスワップされたページの合計数を示します。
ページのスワップアウト率	平均	ページがディスクへスワップされる頻度を示します。
使用されているページ	平均	使用されたキャッシュページ数。ディスクにスワップされたページが含まれるため、キャッシュサイズよりも大きくなる場合があります。
クライアントの接続率	平均	現在接続されているクライアントの割合を示します。このメトリックは、 <b>接続しているクライアント数の値をクライアントの最大数の値で除算し、その結果に 100 を乗算することによって得られます。</b>

メトリック名	スコープ	説明
接続の使用率	平均	現在使用されている接続の割合を示します。
ストリームスタック内のページ (%)	平均	ネットワークストリームで保持されているページの割合を示します。
ロックされているページ (%)	平均	ロックされているページの割合を示します。
使用済みページ (%)	平均	使用されているページの割合を示します。ディスクにスワップされたページが含まれるため、キャッシュサイズよりも大きくなる場合があります。PAGES_USED を参照してください。
<i>ping</i>	合計	現在接続されている、ping を実行するクライアント数を示します。
未加工の TCP ステージ長	平均	ネットワークワークキューの長さを示します。
要求	合計	処理されている有効な要求の数を示します。
アップロード適用フェーズの要求	平均	現在、アップロードの適用フェーズにある同期の数を示します。
ユーザ認証フェーズの要求	平均	現在、ユーザ認証フェーズにある同期の数を示します。
同期開始フェーズの要求	平均	現在、同期開始フェーズにある同期の数を示します。
ダウンロード確認の接続フェーズの要求	平均	現在、ダウンロード確認の接続フェーズにある同期の数を示します。
接続フェーズの要求	平均	現在、接続フェーズにある同期の数を示します。
同期終了フェーズの要求	平均	現在、同期終了フェーズにある同期の数を示します。
ダウンロードフェッチフェーズの要求	平均	現在、ダウンロードのフェッチフェーズにある同期の数を示します。
ダウンロード確認の DB ワーカの取得フェーズの要求	平均	現在、確認の DB ワーカの取得フェーズにある同期の数を示します。
DB ワーカの取得フェーズの要求	平均	現在、DB ワーカの取得フェーズにある同期の数を示します。
非ブロッキングダウンロード確認フェーズの要求	平均	現在、非ブロッキングダウンロード確認フェーズにある同期の数を示します。

メトリック名	スコープ	説明
ダウンロード準備フェーズの要求	平均	現在、ダウンロードの準備フェーズにある同期の数を示します。
アップロードの受信フェーズの要求	平均	現在、アップロードの受信フェーズにある同期の数を示します。
ダウンロード送信フェーズの要求	平均	現在、ダウンロードの送信フェーズにある同期の数を示します。
同期要求フェーズの要求	平均	現在、同期要求フェーズにある同期の数を示します。
ダウンロード確認の待機フェーズの要求	平均	現在、ダウンロード確認の待機フェーズにある同期の数を示します。
ロールバック率	平均	ロールバックの合計数を示します。
ロールバック	平均	ロールバックの合計数を示します。
ローがダウンロードされました	平均	リモートに送信されたローの合計数を示します。
ローのダウンロード率	平均	リモートに送信されたローの合計数を示します。
ローが更新されました	平均	リモートから受信したローの合計数を示します。
ローのアップロード率	平均	リモートから受信したローの合計数を示します。
RTNotifier ライトウェイトポーリングヒット率	平均	リモートタスク要求を最低 1 つ検出したリモートタスクエージェントのライトウェイトポーリング数の率を示します。
RTNotifier ライトウェイトポーリングのヒット	平均	リモートタスク要求を最低 1 つ検出したリモートタスクエージェントのライトウェイトポーリングの数を示します。
RTNotifier ライトウェイトポーリングミス率	平均	リモートタスク要求を検出しなかったリモートタスクエージェントのライトウェイトポーリング数の率を示します。
RTNotifier ライトウェイトポーリングのミス	平均	リモートタスクを検出しなかったリモートタスクエージェントのライトウェイトポーリングの数を示します。
RTNotifier ライトウェイトポーリング率	平均	リモートタスクエージェントのライトウェイトポーリング数の率を示します。

メトリック名	スコープ	説明
RTNotifier ライトウェイトポーリング	平均	リモートタスクエージェントのライトウェイトポーリング数を示します。
サーバキャッシュサイズ	平均	サーバキャッシュのサイズを示します。
サーバによって追跡されているメモリ使用率	平均	サーバによって割り付けられたメモリの量を示します。このメトリックは、MEMORY_USED メトリックを使用できない、Windows 以外のシステムに対して使用しません。Microsoft Windows システムでは、精度を上げるために MEMORY_USED メトリックを使用してください。
同期の開始率	合計	同期が開始される頻度を示します。
ストリームステージ長	平均	高レベルのネットワーク処理キューの長さを示します。
成功した同期	平均	成功した同期の合計数を示します。
同期エラー率	合計	同期エラーの頻度を示します。
同期警告率	合計	サーバに対する同期警告の頻度を示します。
同期	合計	現在接続されているデータ同期数を示します。
同期がブロックされました	平均	データベース接続を現在待機している同期の数を示します。
同期が終了しました	合計	完了した同期の数が表示されます。
同期を開始しました	合計	開始された同期の合計数を示します。
読み取られた TCP バイト	平均	これまでに読み込まれたバイトの合計数を示します。
書き込まれた TCP バイト	平均	これまでに書き込まれたバイトの合計数を示します。
TCP 接続が閉じられました	平均	これまでに閉じられた接続の合計数を示します。
TCP 接続が開かれました	平均	これまでに開かれた接続の合計数を示します。
TCP 接続が拒否されました	平均	これまでに拒否された接続の合計数を示します。

メトリック名	スコープ	説明
時間が指定された作業のステージ長	平均	動的キャッシュワークキューの長さを示します。
不明な接続済みクライアント	平均	接続しているクライアントのうち、オリジンが不明なクライアントの数を示します。
使用中のアップロード接続	平均	現在使用中のアップロード接続の数を示します。
同期を更新中	平均	アップロードフェーズにある同期の数を示します。
VM メモリ使用率	平均	付加された VM で使用されるメモリの量を示します。
データベース接続の待機中	平均	データベース接続を待機している要求数。
警告	平均	警告の平均数を示します。

## 1.8.2.1.11.5 リソースの収集間隔の設定

メトリックがモニタによって収集される頻度を変更します。

### 前提条件

モニタの管理者である必要があります。

### コンテキスト

モニタに表示されるメトリックの精度は、収集間隔に左右されます。デフォルトの収集間隔は 30 秒です。

メトリックが再表示されるとき、リソースの収集間隔はモニタの再表示間隔とは異なります。

### 手順

1. モニタにログインします。
2. [管理](#)をクリックします。
3. [リソース](#)をクリックします。

4. 目的のリソースをクリックし、**設定**をクリックします。
5. **収集間隔**をクリックします。
6. メトリックが収集される間隔を指定します (`hours:minutes:seconds`)。収集間隔は 10 秒以上にします。
7. **保存**をクリックします。
8. 自分のユーザ ID とパスワードを入力して、**OK** をクリックします。
9. **閉じる**をクリックします。
10. **閉じる**をクリックします。

## 結果

リソースの収集間隔が設定されます。

## 関連情報

[メトリックの再表示 \[1265 ページ\]](#)

### 1.8.2.1.11.6 メトリックのエクスポート

グラフや表を関連付けられたメトリックを XML ファイルにエクスポートします。

## コンテキスト

たとえば、**主要なパフォーマンスのメトリック**ウィジェットのほとんどのメトリックをエクスポートできます。ファイルにエクスポートできるデータの最大量は 25 メトリックまたは 100 万ポイントです。

## 手順

1. モニタにログインします。
2. ダッシュボードを開きます。
3. メトリックを表示するウィジェットでドロップダウンメニューの矢印をクリックして、**エクスポート**をクリックし、**メトリックのエクスポート**ウィンドウの指示に従います。

### **i** 注記

25 を超えるメトリックまたは 100 万ポイント以上をエクスポートすることはできません。エクスポートした場合は、エラーメッセージを受け取り、エクスポート画面に戻ります。その場合は、より少ない数のメトリックまたは短い期間を選択します。

4. **エクスポート**をクリックします。
5. プロンプトが表示されたら、ファイル拡張子 `.xml` でファイル名を指定します。
6. **保存**をクリックします。

## 結果

指定したメトリックを含む XML ファイルが作成されます。

## 1.8.2.1.11.7 メトリックの再表示

再表示間隔を変更します。この間隔は、リソースの収集間隔とは異なります。

## コンテキスト

デフォルトでは、モニタの表示は 1 分ごとに自動的に再表示されます。

## 手順

1. 再表示間隔を設定します。
  - a. **ツール** > **ユーザ設定** をクリックします。
  - b. 必要に応じて設定を変更します。デフォルトは 1 分です。
  - c. **保存**をクリックします。
2. **データの再表示**をクリックします。

モニタが、最新のメトリックを取得して表示します。

## 結果

再表示間隔が変更されます。

## 次のステップ

F5 キーを押すと、モニタはブラウザを再ロードし、その時点までに収集されたメトリックを取得して表示します。

## 関連情報

[メトリック \[1245 ページ\]](#)

### 1.8.2.1.12 モニタのユーザ

モニタユーザがモニタにログインするには、ユーザ名とパスワードを使用します。

モニタにログインするためのユーザ ID とパスワードでは、大文字と小文字が区別されます。

モニタは次の 3 種類のユーザをサポートしています。

ユーザのタイプ	説明
読み込み専用ユーザ	モニタリソースに対する読み込み専用アクセスがあります。読み込み専用ユーザはメトリックを表示できますが、 <b>管理</b> ウィンドウにはアクセスできません。ログイン画面からユーザを作成した場合、このユーザは読み込み専用ユーザとなります。ユーザ名とパスワードが必要です。
オペレータ	モニタリソースに対する読み込み専用アクセスがあり、警告を受信できます。このユーザはメトリックの表示、電子メール警告の受信、および警告の解決と削除ができます。ただし、 <b>管理</b> ウィンドウにはアクセスできません。ユーザ名とパスワードが必要です。
管理者	オペレータと同じアクセス権を持ち、さらにリソースの設定とユーザの追加ができます。また、モニタ管理者は <b>管理</b> ウィンドウにもアクセスできます。デフォルトユーザである <i>admin</i> はモニタ管理者です。ユーザ名とパスワードが必要です。

自分のモニタユーザタイプは、モニタにログインし、**ツール** > **ユーザ設定** をクリックすると確認できます。また、ユーザ設定を変更することもできます。ユーザのタイプを変更できるのは、モニタ管理者のみです。

## モニタのデフォルトユーザ

デフォルトで、モニタの初回起動時には、ユーザ名が *admin* でパスワードが *admin* のモニタ管理者ユーザが 1 名、用意されています。デフォルトのモニタ管理者パスワードを変更して、モニタへのアクセスを制限することをお奨めします。

このセクションの内容:

#### [モニタユーザの作成 \[1267 ページ\]](#)

管理ウィンドウを使用して、管理者を含め、あらゆるタイプのユーザを作成します。

#### [読み込み専用モニタユーザの作成 \[1268 ページ\]](#)

自分の読み込み専用ユーザを作成して、モニタに読み込み専用でアクセスします。

#### [モニタユーザとリソースの関連付け \[1269 ページ\]](#)

モニタユーザがリソースに関する電子メール警告を受信するようになるときは、そのユーザをリソースに関連付けます。

#### [モニタユーザの編集 \[1270 ページ\]](#)

モニタユーザを編集します。すべてのユーザは自分自身のユーザ設定を変更できますが、ユーザのタイプを変更できるのはモニタ管理者のみです。

#### [モニタユーザの削除 \[1271 ページ\]](#)

ユーザをモニタから削除して、関連付けられていたリソースとの関連付けをすべて破棄します。

#### [ユーザの作成をモニタ管理者に限定 \[1272 ページ\]](#)

モニタの読み取り専用アクセスユーザを作成する、ユーザのデフォルト機能を無効化します。

## 関連情報

### [モニタの管理ウィンドウ \[1225 ページ\]](#)

## 1.8.2.1.12.1 モニタユーザの作成

管理ウィンドウを使用して、管理者を含め、あらゆるタイプのユーザを作成します。

## 前提条件

モニタの管理者である必要があります。

## 手順

1. モニタにログインします。
2. **ツール** > **管理** をクリックし、**ユーザ** をクリックします。
3. **新規** をクリックして、新しいユーザの情報を入力します。

電子メールアドレスは、モニタからの電子メール警告を受信する必要があるモニタの管理者とオペレータにのみ必要です。

指定した言語は、警告で使用される言語などの、モニタで使用される言語に設定されます。

4. [次へ](#)をクリックして、このユーザのリソースダッシュボードを指定します。
5. [保存](#)をクリックし、[閉じる](#)をクリックします。

## 結果

新しいユーザが作成されます。

## 次のステップ

モニタの管理者またはオペレータを作成した場合は、そのユーザをリソースに関連付けることで、警告の通知を電子メールで受信できるようになります。

## 関連情報

[モニタユーザとリソースの関連付け \[1269 ページ\]](#)

[ユーザの作成をモニタ管理者に限定 \[1272 ページ\]](#)

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

[モニタユーザの編集 \[1270 ページ\]](#)

## 1.8.2.1.12.2 読み込み専用モニタユーザの作成

自分の読み込み専用ユーザを作成して、モニタに読み込み専用でアクセスします。

## コンテキスト

デフォルトで、自分の読み込み専用ユーザを作成して、モニタに読み込み専用でアクセスすることができます。モニタ管理者は、モニタ管理者だけがユーザを作成できるように、このデフォルト動作を変更できます。

## 手順

1. ログイン画面で、[新しいユーザを作成](#)をクリックします。

### i 注記

新しいユーザを作成リンクを使用できない場合は、モニタ管理者だけが新しいユーザを作成できるように、モニタ管理者によってデフォルトの動作が変更されています。新しいユーザの作成については、モニタの管理者に作成を依頼してください。

#### 2. 新しいユーザの情報を入力します。

電子メールアドレスは、モニタからの電子メール警告を受信するためののみ必要です。電子メール警告を受信できるのは、モニタ管理者とオペレータのみです。自分のユーザタイプを変更して電子メール警告を受信するには、モニタ管理者に問い合わせてください。

## 結果

新しい読み込み専用ユーザが作成されます。

## 関連情報

[モニタユーザとリソースの関連付け \[1269 ページ\]](#)

[ユーザの作成をモニタ管理者に限定 \[1272 ページ\]](#)

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

[モニタユーザの編集 \[1270 ページ\]](#)

### 1.8.2.1.12.3 モニタユーザとリソースの関連付け

モニタユーザがリソースに関する電子メール警告を受信するようにするときは、そのユーザをリソースに関連付けます。

## 前提条件

モニタの管理者である必要があります。

モニタが警告の通知を電子メールで送信するよう設定されていることを確認します。

## コンテキスト

リソースに関する電子メール警告を受信できるのは、モニタ管理者とオペレータのみです。

## 手順

1. モニタにログインします。
2. **▶ ツール ▶ 管理 ▶**をクリックし、**ユーザ**をクリックします。
3. モニタ管理者またはオペレータを選択して、そのユーザアカウントに電子メールアドレスが指定されていることを確認します。
4. **電子メール警告通知設定**をクリックします。
5. **追加**および**削除**ボタンを使用して、ユーザをリソースに関連付けます。完了したら、**OK** をクリックします。

## 結果

ユーザが、指定したリソースに関する電子メール警告を受信します。

## 関連情報

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

### 1.8.2.1.12.4 モニタユーザの編集

モニタユーザを編集します。すべてのユーザは自分自身のユーザ設定を変更できますが、ユーザのタイプを変更できるのはモニタ管理者のみです。

## 前提条件

モニタの管理者である必要があります。

## 手順

1. モニタにログインします。
2. **▶ ツール ▶ 管理 ▶**をクリックします。
3. **ユーザ**をクリックします。
4. 編集するユーザをクリックし、**編集**をクリックします。

5. 必要に応じてユーザの設定を変更します。

電子メールアドレスは、モニタからの電子メール警告を受信する必要があるモニタの管理者とオペレータにのみ必要です。

6. [Save](#) をクリックします。
7. [閉じる](#) をクリックします。

## 結果

モニタユーザが編集されます。

## 次のステップ

モニタの管理者またはオペレータを作成した場合は、そのユーザをリソースに関連付けることで、警告の通知を電子メールで受信できるようになります。

## 関連情報

[モニタユーザとリソースの関連付け \[1269 ページ\]](#)

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

[モニタユーザの作成 \[1267 ページ\]](#)

[モニタユーザの削除 \[1271 ページ\]](#)

### 1.8.2.1.12.5 モニタユーザの削除

ユーザをモニタから削除して、関連付けられていたリソースとの関連付けをすべて破棄します。

## 前提条件

モニタの管理者である必要があります。

## 手順

1. モニタにログインします。

2. **ツール** > **管理** をクリックし、**ユーザ**をクリックします。
3. **ユーザ**を選択して、**削除**をクリックします。
4. **はい**をクリックして、選択したユーザを削除します。
5. **閉じる**をクリックします。

## 結果

ユーザがモニタから削除されます。

## 関連情報

[モニタユーザの作成 \[1267 ページ\]](#)

[モニタユーザの編集 \[1270 ページ\]](#)

## 1.8.2.1.12.6 ユーザの作成をモニタ管理者に限定

モニタの読み取り専用アクセスユーザを作成する、ユーザのデフォルト機能を無効化します。

## 前提条件

モニタの管理者である必要があります。

## 手順

1. モニタにログインします。
2. **ツール** > **管理** をクリックします。
3. **設定**をクリックします。
4. **編集**をクリックします。
5. **オプション**をクリックします。
6. **すべてのユーザに SQL Anywhere モニタの読み込み専用アクセスを許可しますオプション**をオフにします。
7. **Save** をクリックします。
8. **閉じる**をクリックします。

## 結果

モニタ管理者のみがユーザを作成できます。

## 関連情報

[モニタユーザの作成 \[1267 ページ\]](#)

[モニタユーザの編集 \[1270 ページ\]](#)

### 1.8.2.1.13 警告

警告とは、モニタ管理者またはオペレータによる注目を要する、リソースに関する特定の状況または状態のことです。

警告は、収集されたメトリックを基にモニタによって検出されます。モニタリングされているリソースでは検出されません。

空きディスク領域の減少、ログイン試行の失敗、高メモリ使用率などの状況については、事前に定義された警告があります。リソースを編集すると、デフォルトのスレッシュホールド値を変更できます。

警告の条件が満たされると、その警告が、指定したリソースの**警告リスト**ウィジェットにリストされます。また、警告が発生したときにモニタ管理者とオペレータに電子メールが送信されるよう、SQL Anywhere モニタを設定することもできます。

デフォルトで、警告は**警告リスト**ウィジェットに表示され、警告には、問題の原因に関する情報や、問題解決のアドバイスなどが含まれます。**リソースリスト**で、リソースのステータスは警告の存在と重大度を反映して変更されます。

1 リソースにつき最新の 50 個の警告のみがモニタによって格納され、**警告リスト**ウィジェットに表示されます。それよりも古い警告は削除されます。

このセクションの内容:

[警告の表示 \[1274 ページ\]](#)

モニタのリソースに関する警告を表示します。

[警告の解決および削除 \[1275 ページ\]](#)

警告の原因となった問題に対応した後に、その警告に解決済みのマークをつけるか警告を削除します。

[警告の定義とスレッシュホールド \[1276 ページ\]](#)

モニタ管理者は、警告のトリガに使用するスレッシュホールドを設定できます。

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

警告が発生したときにモニタ管理者とオペレータに電子メールが送信されるよう、モニタを設定します。

## 1.8.2.113.1 警告の表示

モニタのリソースに関する警告を表示します。

### コンテキスト

警告を解決して削除できるのは、モニタ管理者とオペレータだけです。

1 リソースにつき、最新の 100 個の警告だけがモニタによって格納され、[警告リスト](#)に表示されます。

### 手順

1. [警告](#) > [今日](#) をクリックします。
2. リストで警告をクリックし、[詳細](#)をクリックします。

### 結果

警告の詳細がウィンドウボックスに表示されます。

### 次のステップ

警告の詳細を閉じるには、[OK](#) をクリックします。

### 関連情報

[警告の解決および削除 \[1275 ページ\]](#)

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

## 1.8.2.1.13.2 警告の解決および削除

警告の原因となった問題に対応した後に、その警告に解決済みのマークをつけるか警告を削除します。

### 前提条件

モニタの管理者またはオペレータである必要があります。

### コンテキスト

警告を解決すると、警告のステータスが `user-name` によって解決済みに変更されますが、警告そのものは警告リストに残ります。

警告をリストから削除するには、その警告を削除します。

警告リストウィジェットから警告を削除しても、警告の電子メールには影響はありません。

### 手順

1. モニタにログインします。
2. 概要をクリックします。
3. 警告リストウィジェットで、警告をクリックして以下のオプションのいずれかを選択します。

アクション	説明
警告を解決済みとマークする	警告リストウィジェットで、リストで警告をクリックし、解決済みとマークするをクリックします。
警告を削除する	<ol style="list-style-type: none"><li>1. 警告リストウィジェットで、リストから警告をクリックし、削除をクリックします。</li><li>2. ウィンドウの指示に従います。</li></ol>

### 結果

- 解決済みの警告:
  - デフォルトでは警告リストウィジェットから警告が削除されます。
  - その警告が該当リソースの唯一の未解決警告だった場合は、リソースリストウィジェットのリソースのステータスが正常 (アイコンなし) に変わります。
- 削除した警告:
  - 警告リストウィジェットから警告が削除されます。

## 関連情報

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

[警告の表示 \[1274 ページ\]](#)

### 1.8.2.1.13.3 警告の定義とスレッシュホールド

モニタ管理者は、警告のトリガに使用するスレッシュホールドを設定できます。

警告とそのデフォルトのスレッシュホールドを次に示します。

同期システムはそれぞれ、動作と制約が異なるため、デフォルト設定が環境に不適切である場合があります。各警告スレッシュホールドを確認し、必要に応じて調整してください。

#### データベースリソースの警告スレッシュホールド

CPU 使用率が、指定した秒数の間に指定したスレッシュホールドを超えた場合に警告します

スレッシュホールド (%)

リソースの CPU 使用率が指定された割合に達すると、警告が発行されます。デフォルトは 95% です。

秒

デフォルトは 30 秒です。

メモリ使用率が最大キャッシュサイズの次の割合に達したときに警告します : X%

リソースのメモリ使用率が指定された割合に達すると、警告が発行されます。デフォルトは 90% です。

DB 領域あたりの空きディスク領域が次の値を下回ったときに警告します : X MB

DB 領域あたりの空きディスク領域が指定された容量より少なくなると、警告が発行されます。デフォルトは 100 MB です。

接続がブロックされた時間が次の時間を超えたときに警告します : X 秒

接続がブロックされた時間が指定された時間を超えると、警告が発行されます。デフォルトは 10 秒です。

使用接続数がライセンス制限の次の割合に達したときに警告します : X %

使用中の接続数がライセンス制限に対して指定された割合に達すると、警告が発行されます。デフォルトは 85% です。

クエリの実行時間が次の時間を超えたときに警告します : X 秒

クエリの実行時間が指定された時間を超えると、警告が発行されます。デフォルトは 10 秒です。

スケジュールされていない要求の数が次の値に達したときに警告します : X

スケジュールされていない要求の数が指定された数に達すると、警告が発行されます。デフォルトは 5 です。

最後のサーバ側のバックアップから指定日数を経過したときに警告します

最後のサーバ側のバックアップから指定日数を経過したときに警告します。リソースデータベースが最後にサーバ側でバックアップされてからの日数が指定した日数を超えると、警告を発行します。デフォルトは 14 日です。データベースがバックアップされないと、この警告チェックは実行されません。

切断されたスケールアウトノードの数が指定したスレッシュホールドを超えたときに警告します

読み込み専用スケールアウトシステムまたはミラーリングシステムにデータベースが含まれていて、システムで、切断されたノードが指定の率になっているときに、警告を発行します。

同じ状況が次の時間内で発生した場合は警告しません: X 分

このオプションにより、指定した時間内に警告を重複して受信することがなくなります。デフォルトは 30 分です。

## Mobile Link サーバリソースの警告スレッシュホールド

CPU 使用率が、指定した秒数の間に指定したスレッシュホールドを超えた場合に警告します

スレッシュホールドのデフォルトは 100% です。秒のデフォルトは 300 です。

使用されるキャッシュページの割合が次の値 (%) よりも大きい場合に警告します

デフォルトは 100 です。

ロックされているキャッシュページの割合が次の値 (%) よりも大きい場合に警告します

デフォルトは 80 です。

1 秒間にスワップインおよびスワップアウトされているページ数が、指定した秒数の間に指定したスレッシュホールドを超えた場合に警告します

スレッシュホールドのデフォルトは 256 です。秒のデフォルトは 120 です。

最も長いアクティブな同期の時間が次の値 (秒) よりも長い場合に警告します

デフォルトは 600 です。

失敗した同期の数が、指定した分数の間に指定したスレッシュホールドを超えた場合に警告します

失敗した同期のスレッシュホールドのデフォルトは 20 です。分のデフォルトは 60 です。

エラー数が、指定した分数の間に指定したスレッシュホールドを超えた場合に警告します

スレッシュホールド (エラー) のデフォルトは 50 です。分のデフォルトは 60 です。

Mobile Link キャッシュ用ディスク空き領域が次の値 (MB) を下回ったときに警告します

デフォルトは 100 です。

データベースワークスレッドの最も長いアクティブ待機が次の値 (秒) よりも長い場合に警告します

デフォルトは 300 です。

同じ状況が次の時間内で発生した場合は警告しません

このオプションにより、指定した時間内に警告を重複して受信することがなくなります。デフォルトは 30 分です。

## Web サービスリソースの警告スレッシュホールド

Web サービスが使用できないときに警告

Web サービスが特定期間使用できないときに警告を発行します。

同じ状況が次の時間内で発生した場合は警告しません

このオプションにより、指定した時間内に警告を重複して受信することがなくなります。デフォルトは 30 分です。

このセクションの内容:

#### [警告スレッシュホールドの指定 \[1278 ページ\]](#)

モニタで警告を発行するタイミングを設定するには、次の手順に従います。

#### [リソースからの未送信のエラーレポートに関する警告の抑制 \[1279 ページ\]](#)

リソースに未送信のエラーレポートがある場合に警告を送信するかどうかを設定します。デフォルトでは、このような警告は送信されません。

## 関連情報

### [警告スレッシュホールドの指定 \[1278 ページ\]](#)

## 1.8.2.1.13.3.1 警告スレッシュホールドの指定

モニタで警告を発行するタイミングを設定するには、次の手順に従います。

## 前提条件

モニタの管理者である必要があります。

バージョン 16 以降、データベースの警告スレッシュホールドを設定するには、SYS\_SAMONITOR\_ADMIN\_ROLE ロールを持つ、データベース内のユーザのユーザ ID とパスワードが必要です。それより前のバージョンのデータベースのスレッシュホールドを設定するには、DBA 権限を持つ、データベース内のユーザのユーザ ID とパスワードが必要です。

## 手順

1. **ツール > 管理** をクリックします。
2. **リソース** をクリックして、リストからリソースを選択し、**設定** をクリックします。
3. **警告スレッシュホールド** をクリックして、スレッシュホールドの設定を編集します。
4. **保存** をクリックします。
5. **閉じる** をクリックします。

## 関連情報

[メトリック \[1245 ページ\]](#)

[警告の定義とスレッシュホールド \[1276 ページ\]](#)

### 1.8.2.1.13.3.2 リソースからの未送信のエラーレポートに関する警告の抑制

リソースに未送信のエラーレポートがある場合に警告を送信するかどうかを設定します。デフォルトでは、このような警告は送信されません。

## 前提条件

モニタの管理者である必要があります。

## 手順

1. モニタにログインします。
2. **ツール > 管理** をクリックします。
3. **設定** をクリックします。
4. **編集** をクリックします。
5. **オプション** をクリックします。
6. **リソースの未送信のエラーレポートの警告を表示しません** をクリックします。
7. **保存** をクリックします。
8. **閉じる** をクリックします。

## 結果

リソースに未送信のエラーレポートがあると警告が送信されるよう、モニタが設定されます。

## 関連情報

[トラブルシューティング: エラーのレポート \[973 ページ\]](#)

## 1.8.2.113.4 モニタによる警告電子メールの送信の有効化

警告が発生したときにモニタ管理者とオペレータに電子メールが送信されるよう、モニタを設定します。

### 前提条件

これを行うには、モニタ管理者で電子メールプロトコルについての情報を知っている必要があります。

### コンテキスト

モニタでは、SMTP プロトコルと MAPI プロトコルを使用した電子メール送信をサポートしています。

### 手順

1. 電子メールの警告を受信するようユーザを設定します。
  - a. **ツール** > **管理** をクリックします。
  - b. **ユーザ** をクリックして、自分のユーザ ID を選択します。
  - c. **編集** をクリックして、**電子メールフィールド** に有効な電子メールアドレスを入力します。
  - d. **保存** をクリックします。管理ウィンドウは閉じないでください。
  - e. **ユーザタブ** で **電子メール警告通知設定** をクリックします。
  - f. **追加** および **削除** ボタンを使用して、目的のリソースを選択します。
  - g. **OK** をクリックします。管理ウィンドウは閉じないでください。
2. 電子メールの警告通知を送信するようにモニタを設定します。
  - a. **設定タブ** をクリックして、**編集** をクリックします。
  - b. **警告の通知タブ** で **警告の通知を電子メールで送信** をクリックします。
  - c. 電子メールプロトコルを指定します。
    - SMTP について、電子メールサーバとポート、および認証が必要かどうかを指定します。送信者の名前とアドレスを指定するときは、送信者名がユニークな識別子であることを確認してください。
    - MAPI について、ユーザ ID とパスワードを指定します。
  - d. **テスト電子メールを送信** をクリックして、電子メール通知を正しく設定しているかをテストします。
  - e. **保存** をクリックし、**閉じる** をクリックします。

### 関連情報

[モニタユーザの作成 \[1267 ページ\]](#)

[モニタユーザの編集 \[1270 ページ\]](#)

[モニタユーザとリソースの関連付け \[1269 ページ\]](#)

[モニタユーザの作成 \[1267 ページ\]](#)

[モニタユーザとリソースの関連付け \[1269 ページ\]](#)

[警告の解決および削除 \[1275 ページ\]](#)

[警告の表示 \[1274 ページ\]](#)

## 1.8.2.1.14 モニタのバックアップ

モニタでは、デフォルトで、毎日 1 回夜中の 12 時にメトリックのメンテナンスが実行されます。メンテナンスの対象はメトリックで、警告は対象ではありません。

モニタ管理者は、次の操作を実行できます。

- モニタにメトリックをバックアップするスケジュールを設定します。
- メトリックが保持される時間の長さを制御します。
- メンテナンスをオンデマンドで実行します。

このセクションの内容:

[メトリックのバックアップ \[1281 ページ\]](#)

メトリックをバックアップし、メトリックが保持される時間の長さを制御します。

[モニタデータベースのバックアップコピーのリストア \[1282 ページ\]](#)

モニタデータベースをバックアップコピーに置き換えます。

## 関連情報

[時刻の表示方法の概要 \[1224 ページ\]](#)

### 1.8.2.1.14.1 メトリックのバックアップ

メトリックをバックアップし、メトリックが保持される時間の長さを制御します。

## 前提条件

モニタの管理者である必要があります。

## 手順

1. モニタにログインします。
2. **管理**をクリックします。
3. **設定**をクリックし、**編集**をクリックします。
4. **メンテナンス**をクリックします。
5. メンテナンスの実行時刻 (24 時間表示) を指定します。デフォルトでは、夜中の 12 時に実行されます。この時刻は、モニタが実行されているコンピュータのローカル時間に対応します。
6. バックアップデータを保存するフォルダを指定します。フォルダは、モニタが実行されているコンピュータに存在する必要があります。
7. **データの削減**設定をカスタマイズします。

次よりも古い値を削除する

時間の長さを指定します。指定した時間より長く保持されているメトリックは削除されます。デフォルトは 5 日間です。
8. 省略可能です。**メンテナンスを今すぐ実行**をクリックして、すぐにバックアップを実行し、古くなったメトリックを削除します。プロンプトが表示されたら、**保存**をクリックします。
9. **OK**をクリックします。
10. **キャンセル**をクリックします。

## 結果

バックアップと古くなったメトリックの削除の、日単位のスケジュールが保存されます。

### 1.8.2.1.14.2 モニタデータベースのバックアップコピーのリストア

モニタデータベースをバックアップコピーに置き換えます。

## 前提条件

モニタデータベースの有効なバックアップコピーが必要です。

## 手順

1. モニタが実行されている場合は、停止します。
2. バックアップフォルダから `samonitor.db` データベースファイルと `samonitor.log` トランザクションログファイルをコピーします。

- 現在のモニタデータベースファイルとトランザクションログファイルが格納されているフォルダに、これらのファイルを貼り付けます。プロンプトが表示されたら、既存のファイルを上書きします。

バージョン 17.0.4 モニタデータベースファイルのデフォルトのロケーションを次の表に示します。

オペレーティングシステム	モニタのフォルダ
Windows 7 以降 (SQL Anywhere とともにインストール)	C:\Users\Public\Documents\SQL Anywhere 17\Monitor\samonitor.db
Windows 7 以降 (別のコンピュータにインストール)	C:\Users\Public\Documents\SQL Anywhere 17\Monitor\samonitor.db
Linux (SQL Anywhere とともにインストール)	/opt/sqlanywhere17/samonitor.db
Linux (別のコンピュータにインストール)	/opt/sqlanywhere17/samonitor.db

- モニタを再起動します。

## 結果

データベースがバックアップコピーからリストアされます。

### 1.8.2.1.15 モニタがインストールしたデータベースオブジェクトのリスト

データベースをモニタリング対象のリソースとして追加する場合にインストールされるオブジェクトがいくつかあります。

オブジェクト名	オブジェクトタイプ	説明
sa_monitor_user	データベースユーザ	これは読み込み専用ユーザで、メトリックを収集するためにデータベースに追加されます。このユーザはモニタリングされるデータベースに追加されるため、そのデータベースの外部のどこかにユーザクレデンシャルを格納する必要はありません。 sa_monitor_user に対して、パスワード検証の回避を許可する必要がある場合があります。sa_monitor_user には、モニタだけが知っているランダムなパスワードがあります。このユーザはデータベースで、CREATE EXTERNAL REFERENCE、MONITOR、MANAGE ANY DBSPACE、BACKUP DATABASE の各権限を持っています。

オブジェクト名	オブジェクトタイプ	説明
sa_monitor_connection_failure	テーブル	このテーブルには、失敗した接続試行に関するメトリックが含まれ、sa_monitor_connection_failed_event と共に使用されます。このテーブルに含まれるメトリックは、メトリックがモニタによって取得されると削除されます。
sa_monitor_connection_failed_event	イベント	このイベントは、ConnectFailed システムイベントを接続試行が失敗するたびに発行し、レコードを sa_monitor_connection_failure テーブルに挿入します。
sa_monitor_count_unsubmitted_crash_reports	関数	この関数は、xp_srvmon_count_unsubmitted_crash_reports プロシージャを呼び出し、未送信のクラッシュレポートの数を判断します。

## 関連情報

[モニタでモニタリングするデータベースリソースの修復 \[1243 ページ\]](#)

[リソースデータベースからのモニタリングオブジェクトの削除 \[1284 ページ\]](#)

### 1.8.2.1.16 リソースデータベースからのモニタリングオブジェクトの削除

リソースをモニタに追加したとき、インストールされたリソースデータベースからモニタリングオブジェクトを削除します。

#### 前提条件

バージョン 16 以降、データベースからモニタリングオブジェクトを削除するには、SYS\_SAMONITOR\_ADMIN\_ROLE ロールを持つ、データベース内のユーザのユーザ ID およびパスワードが必要です。16.0 より前のバージョンのデータベースからモニタリングオブジェクトを削除するには、DBA 権限を持つ、データベース内のユーザのユーザ ID とパスワードが必要です。

#### コンテキスト

データベースオブジェクトの所有者は 1 人なので、sa\_monitor\_user を削除することですべてのオブジェクトを削除できます。

## 手順

次の文を実行します。

```
DROP USER sa_monitor_user;
```

## 結果

モニタによってインストールされたデータベースオブジェクトが、データベースから削除されます。

### 1.8.2.1.17 モニタ通信のセキュリティ

モニタとブラウザ間、およびモニタとモニタリング対象のリソース間の通信を保護できます。

このセクションの内容:

[モニタの TLS セキュリティの設定 \[1285 ページ\]](#)

トランスポートレイヤセキュリティ (TLS) を使用すると、モニタとブラウザ間の通信が保護されます。

[モニタとリソースの間のセキュア接続 \[1286 ページ\]](#)

FIPS 認定の暗号化を使用すると、モニタとモニタリング対象のリソース間の通信を暗号化できます。

#### 1.8.2.1.17.1 モニタの TLS セキュリティの設定

トランスポートレイヤセキュリティ (TLS) を使用すると、モニタとブラウザ間の通信が保護されます。

## コンテキスト

モニタでは、TLS バージョン 1.0 および 1.1 を使用した HTTPS 接続をサポートする Web サーバが実行されます。

## 手順

1. 認証局からデジタル証明書を取得するか、証明書の作成ユーティリティ (createcert) を使用して自己署名証明書を作成します。
2. モニタの開始行の文字列を変更して、証明書を使用するようにします。
3. 必要に応じて、新しい証明書を受け入れるようにブラウザを設定します。

## 結果

モニタとブラウザの間の通信が保護されます。

## 関連情報

[Securing the SQL Anywhere Monitor With Transport Layer Security \(トランスポートレイヤセキュリティを使用する SQL Anywhere モニタの保護\)](#)

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

### 1.8.2.1.17.2 モニタとリソースの間のセキュア接続

FIPS 認定の暗号化を使用すると、モニタとモニタリング対象のリソース間の通信を暗号化できます。

FIPS 認定の暗号化には別途ライセンスが必要です。

### 1.8.2.1.18 モニタのトラブルシューティング

モニタ管理者がモニタのトラブルシューティングする方法は、いくつかあります。

たとえば、次に示す推奨事項に加えて、モニタ管理者はメッセージログ機能と例外レポート機能を使用して、モニタをトラブルシューティングできます。

問題	推奨事項
F5 キーを押してブラウザウィンドウを再表示すると、SQL Anywhere モニタへのログインが要求されます。	ブラウザで JavaScript を有効にします。
モニタにログインしようとすると、ネットワーク通信エラーになります。	モニタを起動します。
Adobe Flash Player を最新バージョンにアップグレードした後も Adobe Flash Player のアップグレードを求めるメッセージが表示されます。	インストールされている Adobe Flash Player のバージョンが、オペレーティングシステムでサポートされていることを確認します。モニタには、Adobe Flash Player のバージョン 10 との互換性があります。
モニタでデータベースリソースのモニタリングを開始できません。	該当リソースのパスワード検証関数とログインプロシージャで、user sa_monitor_user によるリソースへの接続が許可されていることを確認します。
警告の電子メールをまったく受信しません。	モニタが電子メールを送信するよう適切に設定されていることを確認し、テスト電子メールを送信します。  モニタが送信する警告の電子メールがウイルススキャナーによってブロックされていないことを確認します。

問題	推奨事項
<p>モニタからレポートされるスケジュールされていない要求の数が、実際より少ない数になっています。</p>	<p>スケジュールされていない要求の数に関するメトリックを収集する際、モニタはリソースに対してクエリを実行します。このクエリがスケジュールされていない要求になっている可能性が考えられます。</p> <p>スケジュールされていないクエリは、到着順に処理されます。そのため、モニタがクエリを実行しようとしたときにスケジュールされていない要求が存在すると、そのクエリは既存のスケジュールされていない要求が完了するのを待って処理されます。</p> <p>その結果、モニタがスケジュールされていない要求の数を収集するとき、その数には、モニタがクエリを発行してからそのクエリが実行されるまでに存在したスケジュールされていない要求が含まれません。</p>
<p>データベースのディスク領域が指定されたスレッシュホールドを超えたのに警告を受信しません。</p>	<p>データベースは、モニタによる収集間隔の間に、指定されたディスク領域の警告スレッシュホールドと使用可能な領域の上限を超える可能性があります。このような状況になると、モニタがディスク使用率メトリックを収集して警告を発行する前に、データベースが応答しなくなります。</p> <p>データベースのサイズが急激に増大する場合は、ディスク領域の警告スレッシュホールドの値を大きくして、データベースが領域不足になる前に警告を受信できるようにします。</p>
<p>モニタにログインしているとき、<b>管理</b>ウィンドウを表示できません。</p>	<p><b>管理</b>ウィンドウにアクセスするには、モニタに管理者としてログインする必要があります。</p>
<p>リソースを移行する前にモニタをアンインストールしたら、リソースを失いました。</p>	<p>モニタをアンインストールすると、アプリケーションだけでなく、リソース、収集したメトリックも削除されます。アップグレードするときは、モニタの新しいバージョンをインストールし、リソースと設定を移行してから、古いバージョンをアンインストールする必要があります。</p> <p>ただし、モニタを定期的にバックアップしている場合は、移行ユーティリティを使用して、バックアップファイルからモニタの新しいバージョンへリソースを移行できます。</p>

## 関連情報

[警告の定義とスレッシュホールド \[1276 ページ\]](#)

[モニタのユーザ \[1266 ページ\]](#)

[モニタのバックアップ \[1281 ページ\]](#)

[モニタの起動 \[1212 ページ\]](#)

[モニタによる警告電子メールの送信の有効化 \[1280 ページ\]](#)

[メッセージログの表示 \[1226 ページ\]](#)

[例外レポートの表示 \[1226 ページ\]](#)

## 1.8.2.119 チュートリアル: モニタによるリソースのモニタリング

データベース、または Mobile Link サーバのモニタリングを設定します。

### 前提条件

DROP CONNECTION システム権限を持っている、モニタリング対象のデータベース内のユーザのユーザ ID とパスワードを知っている必要があります。

チュートリアルでレッスンを実行するには、モニタに管理者としてログインしてください。読み込み専用ユーザとオペレータユーザが実行できるのは、管理者が実行できるタスクのサブセットのみです。自分のモニタユーザタイプを確認するには、**ツール** ▶ **ユーザ設定** をクリックし、**ユーザタイプ**設定を見直します。

バージョン 16 以降のデータベースを追加するには、SYS\_SAMONITOR\_ADMIN\_ROLE ロールを持つ、データベース内のユーザのユーザ ID とパスワードが必要です。16.0 より前のバージョンのデータベースを追加するには、DBA 権限を持つ、データベース内のユーザのユーザ ID とパスワードが必要です。これらのクレデンシャルは次の操作を行う場合に使用されます。

- データベースに接続します。
- **sa\_monitor\_user** という名前の新しいユーザを作成します。モニタは、sa\_monitor\_user を使用してデータベースに接続し、モニタリングします。
- データベースをモニタリングするために sa\_monitor\_user ユーザが必要とするデータベースオブジェクトをインストールします。
- 提供されたユーザ ID クレデンシャルをモニタから破棄します。sa\_monitor\_user は、モニタリングされるデータベースに追加されるため、SYS\_SAMONITOR\_ADMIN\_ROLE ロールを持つユーザのクレデンシャルと DBA クレデンシャルのいずれも、そのデータベースの外部のどこかに格納する必要はありません。

### コンテキスト

このチュートリアルでは、モニタ Developer Edition を使用しています。

このチュートリアルを使用して、データベース、または Mobile Link サーバファームのモニタリングを設定します。このチュートリアルでは、モニタ Developer Edition を使用しています。

このセクションの内容:

#### [レッスン 1: モニタへのログイン \[1289 ページ\]](#)

モニタにログインして、リソースに関する情報の追加、モニタリング、収集を行います。

#### [レッスン 2: リソースの追加 \[1290 ページ\]](#)

モニタリング対象のリソースとして、SQL Anywhere サンプルデータベースを追加します。

#### [レッスン 3: 警告のテスト \[1292 ページ\]](#)

警告をトリガし、警告の処理を練習することができます。

#### [レッスン 4: 新しいダッシュボードとウィジェットの追加 \[1293 ページ\]](#)

ダッシュボードとウィジェットをモニタに追加して、リソースに関する情報を表示します。

[レッスン 5: データベース接続の終了 \[1294 ページ\]](#)

リソースデータベースへの接続を終了します。

[レッスン 6: リソースのモニタリングの停止 \[1295 ページ\]](#)

モニタリングしているリソースを削除し、収集したメトリックを削除して、データ収集を停止します。

[レッスン 7: \(省略可\) データベースミラーリングシステムのモニタリング \[1297 ページ\]](#)

モニタを使用してデータベースミラーリングシステムをモニタリングします。

[レッスン 8: \(省略可\) SQL Anywhere モニタからの読み込み専用スケールアウトシステムのモニタリング \[1298 ページ\]](#)

読み込み専用スケールアウトシステムでルートノードであるデータベースを起動し、このデータベースをモニタのリソースとして追加します。

## 関連情報

[モニタのユーザ \[1266 ページ\]](#)

### 1.8.2.1.19.1 レッスン 1: モニタへのログイン

モニタにログインして、リソースに関する情報の追加、モニタリング、収集を行います。

## 前提条件

このチュートリアル の冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

1. モニタを起動します。モニタが現在バックグラウンドで実行されていないことが必要です。

モニタ Developer Edition を起動するには、次の手順に従います (Windows の場合)。

▶ スタート ▶ プログラム ▶ SQL Anywhere17 ▶ 管理ツール ▶ SQL Anywhere モニタ ▶ をクリックします。

モニタ Developer Edition を起動するには、次の手順に従います (Linux の場合)。

モニタのインストールディレクトリの bin32 または bin64 ディレクトリから samonitor.sh スクリプトを実行します。

```
./samonitor.sh launch
```

モニタによってメトリックの収集が開始され、モニタにログインするためのデフォルト URL `http://localhost:4950` がブラウザに表示されます。

### **i** 注記

ネットワークを経由してモニタにアクセスしている場合は、`http://computer-name:4950` をブラウズします。  
`computer-name` は、モニタが実行されているコンピュータの名前です。

2. デフォルトの管理者ユーザとしてモニタにログインします。
  - a. ユーザ名フィールドに、`admin` と入力します。
  - b. パスワードフィールドに、`admin` と入力します。
3. デフォルトでは、次の 2 つのウィジェットがある概要ダッシュボードが開きます。

#### リソースリスト

このウィジェットは、モニタリングされているリソースのリストを表示します。モニタを初めて開いたときは、*SQL Anywhere* モニタという名前のデフォルトのリソースを使用してモニタ自体のみがモニタリングされています。このリソースを変更したり、このリソースのモニタリングを停止することはできません。

#### 警告リスト

このウィジェットは、モニタリングされているリソースからのあらゆる警告のリストを表示します。

## 結果

モニタに管理者としてログインしています。

## 次のステップ

次のレッスンに進みます。

## 関連情報

[モニタの起動 \[1212 ページ\]](#)

## 1.8.2.1.19.2 レッスン 2: リソースの追加

モニタリング対象のリソースとして、SQL Anywhere サンプルデータベースを追加します。

## 前提条件

このチュートリアル of これまでのレッスンを完了している必要があります。

このチュートリアル冒頭の冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

1. モニタリング対象のリソースを起動します。たとえば、サンプルデータベースを起動するには、**▶ スタート ▶ プログラム ▶ SQL Anywhere17 ▶ SQL Anywhere ▶ ネットワークサーバのサンプル ▶**をクリックします。
2. モニタで、**▶ ツール ▶ 管理 ▶**をクリックします。
3. **リソース**をクリックして、**追加**をクリックします。
4. 次のオプションのうちの 1 つを選択してください。

### データベースリソースの追加

1. **SQL Anywhere サーバ**をクリックして**次へ**をクリックします。
2. **名前**フィールドに **demo17** と入力し、**次へ**をクリックします。
3. **ホスト**フィールドに **localhost** と入力し、**サーバ**フィールドに **demo17** と入力します。
4. **作成**をクリックします。
5. 必要な認証情報が要求されたら、**ユーザ ID**フィールドに **DBA** と入力し、**パスワード**フィールドに **sql** と入力します。**OK** をクリックします。  
リソースが追加され、モニタリングが自動的に開始されます。
5. **閉じる**を 2 回クリックします。
6. **▶ ダッシュボード ▶ 概要 ▶**をクリックします。  
**リソースリスト**にリソースが表示されます。
7. **リソース**をクリックすると、**ダッシュボード**が開き、収集されたメトリックが表示されます。

## 結果

リソースが追加され、モニタリングされます。

## 次のステップ

次のレッスンに進みます。

## 関連情報

[データベースリソースの追加 \[1228 ページ\]](#)

[Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

[モニタがインストールしたデータベースオブジェクトのリスト \[1283 ページ\]](#)

## 1.8.2.1.19.3 レッスン 3: 警告のテスト

警告をトリガし、警告の処理を練習することができます。

### 前提条件

このチュートリアルの前までのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

### 手順

1. データベースサーバをシャットダウンして警告をトリガします。

たとえば、Windows のデータベースサーバをシャットダウンするには、システムトレイのネットワークサーバアイコンをダブルクリックし、データベースサーバメッセージウィンドウで、**シャットダウン**をクリックします。

2. モニタで、**ダッシュボード** > **概要** をクリックします。

リソースリストでリソースの隣に、白い X の付いた赤い円が表示され、リソースが使用できないことを示します。

警告リストウィジェットで、データベースの場合は **"使用できません"** が **アクティブ** のステータスで表示されます。

状態やモニタの状態が変わるまで数秒かかることがあります。デフォルトでは、リソースの情報は 30 秒ごとに収集されます。

3. 警告リストウィジェットで、警告をクリックして説明を確認します。

4. **OK** をクリックして、警告を閉じます。

5. リソースを再起動します。たとえば、サンプルデータベースを起動するには、**スタート** > **プログラム** > **SQL Anywhere17** > **SQL Anywhere** > **ネットワークサーバのサンプル** をクリックします。

リソースリストウィジェットで、リソースのステータスが黄色の三角に変わります。このアイコンは、リソースがモニタリングされており、警告が出されていることを示します。警告リストで、警告のステータスが **非アクティブ** に変わります。非アクティブの警告は、その警告をトリガした問題は存在しませんが、警告が解決または削除されていないことを示しています。

6. 警告を解決します。解決するには、目的の警告を選択し、**解決済みとマークする** をクリックします。

リソースリストで、リソースの隣にアイコンがないので、リソースがモニタリングされており、警告がないことがわかります。

### 次のステップ

次のレッスンに進みます。

## 関連情報

[メトリック \[1245 ページ\]](#)

[モニタのユーザ \[1266 ページ\]](#)

## 1.8.2.1.19.4 レッスン 4: 新しいダッシュボードとウィジェットの追加

ダッシュボードとウィジェットをモニタに追加して、リソースに関する情報を表示します。

### 前提条件

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

### コンテキスト

ダッシュボードはユーザ固有です。どのユーザも、各自のダッシュボードとそこにあるウィジェットを追加、編集、または削除できます。

### 手順

1. **ダッシュボード** > **新規追加** をクリックします。
  - a. 次のリソースをモニタするダッシュボードをクリックし、リソースを選択します。
  - b. **ダッシュボード名**フィールドに、`user_joe` と入力します。
  - c. **カラムの数**フィールドに、`2` と入力します。
  - d. **OK** をクリックします。

新しいダッシュボードが、次のウィジェットとともに表示されます。

**データベースの場合**

**警告リスト、主要なパフォーマンスのメトリック、サーバ情報、および接続**が表示されます。

**Mobile Link の場合**

**警告リスト、主要なパフォーマンスのメトリック、サーバ情報、および接続**が表示されます。

2. デフォルトのスパークラインではなく、グラフを表示するために、メトリックウィジェットを追加します。
  - a. `user_joe` のダッシュボードの右上で、**カスタマイズ** > **ウィジェットの追加** をクリックします。
  - b. **メトリック** をクリックして、**次へ** をクリックします。

- c. このウィジェットの名前を指定してくださいフィールドに **Metrics display** と入力します。
- d. どのようなリソースに関心がありますか? フィールドで、データベースサーバの場合は *demo17* を、Mobile Link サーバの場合は *MobiLinkServerSample* をクリックします。
- e. どのような表示が必要ですか? フィールドで、**グラフ** をクリックします。
- f. どのようなメトリックを確認したいですか? で、次のいずれかをクリックします。

データベースの場合

**CPU 使用率、処理されたクエリ、および** **メモリのメトリック** **キャッシュサイズ** **】**。

**Mobile Link** の場合

**同期のメトリック**

- g. **作成** をクリックします。

ダッシュボードに**メトリック表示**ウィジェットが表示されます。

ウィジェットのサイズを最大にするには、ウィジェットウィンドウ枠でドロップダウンメニューの矢印をクリックし、**最大化** をクリックします。

グラフの詳細を確認するには、カーソルをグラフの特定の位置に置きます。

## 結果

ダッシュボードと、リソースのメトリックウィジェットを作成しました。

## 次のステップ

次のレッスンに進みます。

## 1.8.2.19.5 レッスン 5: データベース接続の終了

リソースデータベースへの接続を終了します。

## 前提条件

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

1. *demo17* ダッシュボードを開きます。
2. **接続**ウィジェットで、閉じる接続の横にある *x* をクリックします。
3. 必要な認証情報が要求されたら、**ユーザ ID** フィールドに **DBA** と入力し、**パスワード** フィールドに **sql** と入力します。

モニタは、ユーザ ID とパスワードを使用して、リソースデータベースに接続し、接続を削除します。このユーザ ID には DROP CONNECTION システム権限が必要です。ユーザ ID とパスワードはモニタによって保持されません。

4. **OK** をクリックします。

**接続**ウィジェットから接続が削除されます。

## 結果

接続が閉じられます。

## 次のステップ

次のレッスンに進みます。

## 1.8.2.1.19.6 レッスン 6: リソースのモニタリングの停止

モニタリングしているリソースを削除し、収集したメトリックを削除して、データ収集を停止します。

## 前提条件

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## コンテキスト

*SQL Anywhere* モニタリソースは削除できません。

## i 注記

データベースリソースを削除しても、データベースにインストールされたモニタリングオブジェクトは削除されません。これらのオブジェクトを削除するには、MANAGE ANY USER システム権限を持つユーザとしてデータベースに接続し、次の文を実行する必要があります。

```
DROP USER sa_monitor_user;
```

## 手順

1. このチュートリアルの一部として追加したリソースを削除します。
  - a. **ツール** > **管理** をクリックします。
  - b. **リソース** をクリックします。
  - c. *demo17*、または *MobiLinkServerSample* を選択して、**停止** をクリックします。
  - d. **削除** をクリックします。
  - e. **はい** をクリックして、リソースの削除を確定します。
  - f. **閉じる** をクリックします。
2. **ログアウト** をクリックしてモニタからログアウトし、モニタを表示しているブラウザウィンドウを閉じます。
3. モニタをシャットダウンしてモニタリングを停止します。
  - Windows では、システムトレイでモニタのアイコンをクリックし、**SQL Anywhere モニタの終了** をクリックします。
  - Linux では、モニタのインストールディレクトリの bin32 または bin64 ディレクトリから samonitor.sh スクリプトを実行します。

```
./samonitor.sh stop
```

モニタは、すべてのリソースについてメトリックの収集を停止します。

4. モニタリングされていたリソースをシャットダウンします。たとえば、データベースをシャットダウンするには、データベースサーバのシステムトレイにあるネットワークサーバのアイコンをダブルクリックし、データベースサーバメッセージウィンドウで、**シャットダウン** をクリックしてから、**はい** をクリックしてアクションを確認します。

## 結果

モニタがシャットダウンして、リソースのモニタリングが停止し、リソースもシャットダウンされます。

## 次のステップ

次のレッスンに進みます。

## 関連情報

[モニタのユーザ \[1266 ページ\]](#)

### 1.8.2.1.19.7 レッスン 7: (省略可) データベースミラーリングシステムのモニタリング

モニタを使用してデータベースミラーリングシステムをモニタリングします。

#### 前提条件

このチュートリアルは、データベースミラーリングシステムの作成に関するチュートリアルの説明に従って読み込み専用スケールアウトシステムを実行していることを前提としています。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

#### 手順

1. モニタのリソースリストにデータベースミラーリングシステムを追加します。
  - a. ツールサイドバーで、**管理**をクリックします。
  - b. **リソース**をクリックして、**追加**をクリックします。
  - c. **SQL Anywhere サーバ**をクリックして**次へ**をクリックします。
  - d. **名前**フィールドに **mirror\_demo** と入力し、**次へ**をクリックします。
  - e. **ホスト**フィールドでプライマリサーバとミラーサーバのホスト名とポート番号を、カンマ区切りのリストで指定します。つまり、**localhost:6871, localhost:6872** と入力します。
  - f. **サーバ**フィールドにプライマリサーバの代替サーバ名を入力します。つまり、**mirror\_demo\_primary** と入力します。
  - g. **作成**をクリックします。
  - h. 必要な認証情報が要求されたら、**ユーザ ID**フィールドに **DBA** と入力し、**パスワード**フィールドに **sql** と入力します。**OK** をクリックします。

リソースが追加され、モニタリングが自動的に開始されます。
  - i. **閉じる**を 2 回クリックします。
2. scaleoutdemo.db をモニタリングするためのウィジェットを追加します。
  - a. **ダッシュボード**サイドバーで、**mirror\_demo** をクリックします。
  - b. mirrorsystemdemo のダッシュボードの右上で、**カスタマイズ** > **ウィジェットの追加** をクリックします。
  - c. **SQL Anywhere スケールアウトポロジ**を選択して、**次へ**をクリックします。
  - d. このウィジェットの**名前**を指定してくださいフィールドに **mirror\_demowidget** と入力します。

- e. どのようなリソースに関心がありますか? フィールドで `mirror_demo` を選択します。
- f. 作成をクリックします。

ダッシュボードに新しいウィジェットが表示されます。

## 結果

データベースミラーリングシステムリソースが追加され、モニタリングされます。

## 次のステップ

次のレッスンに進みます。

## 関連情報

[SQL Anywhere モニタ \[1207 ページ\]](#)

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

[SQL Anywhere スケールアウトポロジウィジェット \[1221 ページ\]](#)

[チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

[データベースリソースの追加 \[1228 ページ\]](#)

[Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

[ウィジェットの管理 \[1220 ページ\]](#)

## 1.8.2.19.8 レッスン 8: (省略可) SQL Anywhere モニタからの読み込み専用スケールアウトシステムのモニタリング

読み込み専用スケールアウトシステムでルートノードであるデータベースを起動し、このデータベースをモニタのリソースとして追加します。

## 前提条件

実行中の読み込み専用スケールアウトシステムを設定するには、このチュートリアルのこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

1. モニタを起動します。▶ [スタート](#) ▶ [プログラム](#) ▶ [SQL Anywhere17](#) ▶ [管理ツール](#) ▶ [SQL Anywhere モニタ](#) をクリックします。
2. デフォルトの管理者ユーザとしてモニタにログインします。
  - a. ユーザ名フィールドに、**admin** と入力します。
  - b. パスワードフィールドに、**admin** と入力します。
3. `scaleoutdemo.db` をリソースとしてモニタに追加するには、次の手順に従います。
  - a. ツールサイドバーで、**管理** をクリックします。
  - b. **リソース** をクリックして、**追加** をクリックします。
  - c. [SQL Anywhere サーバ](#) をクリックして **次へ** をクリックします。
  - d. 名前フィールドに **scaleoutdemo** と入力し、**次へ** をクリックします。
  - e. ホストフィールドに、**localhost** と入力します。
  - f. サーバフィールドに、**scaleout\_root\_demo** と入力します。
  - g. **作成** をクリックします。
  - h. 必要な認証情報が要求されたら、**ユーザ ID** フィールドに **DBA** と入力し、**パスワード** フィールドに **sql** と入力します。**OK** をクリックします。

リソースが追加され、モニタリングが自動的に開始されます。
  - i. **閉じる** を 2 回クリックします。
4. `scaleoutdemo.db` をモニタリングするためのウィジェットを追加します。
  - a. **ダッシュボード** サイドバーで、**scaleoutdemo** をクリックします。
  - b. `scaleoutdemo` のダッシュボードの右上で、▶ [カスタマイズ](#) ▶ [ウィジェットの追加](#) をクリックします。
  - c. [SQL Anywhere スケールアウトポロジ](#) を選択して、**次へ** をクリックします。
  - d. このウィジェットの名前を指定してくださいフィールドに **scaleoutwidget** と入力します。
  - e. [どのようなリソースに関心がありますか?](#) フィールドで `scaleoutdemo` を選択します。
  - f. **作成** をクリックします。

ダッシュボードに新しいウィジェットが表示されます。

## 結果

読み込み専用スケールアウトシステムをリソースとしてモニタに追加し、この読み込み専用スケールアウトシステムのモニタリングとメトリックの収集を行うウィジェットを作成しました。

## 関連情報

[SQL Anywhere モニタ \[1207 ページ\]](#)

[ダッシュボードおよびウィジェット \[1218 ページ\]](#)

[SQL Anywhere スケールアウトポロジウィジェット \[1221 ページ\]](#)

[チュートリアル: 読み込み専用のスケールアウトシステムの作成 \[1753 ページ\]](#)

[データベースリソースの追加 \[1228 ページ\]](#)

[Mobile Link サーバリソースの追加 \[1230 ページ\]](#)

[ウィジェットの管理 \[1220 ページ\]](#)

## 1.8.2.2 Solution Manager

SAP Solution Manager は全体的な SAP ランドスケープの一部として、SQL Anywhere をモニタすることができるツールを提供しています。

### 設定ファイル

`ncs.conf` ファイルはモニタリングデータを SAP Solution Manager に送るために必要なすべての接続と設定情報を提供します。モニタリングを有効にするには、`-ncs` データベースサーバオプションを使用します。`-ncs` または `-ncsf` データベースサーバオプションを使用して `ncs.conf` 設定ファイルの場所を指定します。

### 関連情報

[-ncs データベースサーバオプション \[450 ページ\]](#)

[-ncsd データベースサーバオプション \[451 ページ\]](#)

[Maintenance of SAP Mobile Platform in the System Landscape \(システムランドスケープでの SAP Mobile Platform のメンテナンス\)](#)

[SAP Solution Manager Setup \(SAP Solution Manager セットアップ\)](#)

## 1.8.2.3 SQL Anywhere SNMP Extension Agent

SQL Anywhere SNMP Extension Agent を SNMP 管理アプリケーションとともに使用して、データベースを管理できます。

異なるコンピュータで動作している複数のデータベースサーバ上の各データベースを 1 つのエージェントでモニタリングできます。

SQL Anywhere SNMP Extension Agent を使用すると、次の操作を実行できます。

- 個々のサーバ統計とデータベース統計の値を検索します。
- 個々のサーバプロパティとデータベースプロパティの値を検索します。
- 個々の PUBLIC データベースオプションの値を検索します。
- いずれかの PUBLIC データベースオプションに値を設定します。
- ストアドプロシージャを実行します。

- プロパティ値または統計値に基づいてトラップを生成します。

## 提供されるファイル

SQL Anywhere インストール環境には、以下の SQL Anywhere SNMP Extension Agent 用ファイルが用意されています。

### **dbsnmp17.dll**

SQL Anywhere SNMP Extension Agent。このファイルは、`%SQLANY17%\bin32` にあります。

### **SQLAnywhere.mib**

SQL Anywhere MIB には、データベースサーバとデータベースのプロパティ、統計、オプションのうち SQL Anywhere SNMP Extension Agent を使用してアクセスできるものの OID が、すべて格納されています。

### **RDBMS-MIB.mib**

これはリレーショナルデータベース管理システムのための汎用 MIB であり、SQL Anywhere SNMP Extension Agent を使用してアクセスできる OID が格納されています。

### **SNMPv2-SMI.mib**

この MIB は、SQL Anywhere と RDBMS MIB から参照されます。

### **SNMPv2-TC.mib**

この MIB は、SQL Anywhere と RDBMS MIB から参照されます。

### **SYBASE-MIB.mib**

Sybase MIB です。この MIB は、ASQL Anywhere MIB から参照されます。

### **sasnmp.ini**

このファイルには、SQL Anywhere SNMP Extension Agent によるモニタリングの対象となるデータベースが列記されています。デフォルトで、このファイルは `%SQLANY17%\bin32` にあります。

SQL Anywhere SNMP Extension Agent を使用するには、コンピュータに SNMP がインストールされている必要があります。また、SQL Anywhere SNMP Extension Agent を使用してモニタリングするデータベースについての情報が保存されている `sasnmp.ini` ファイルを作成する必要があります。

このセクションの内容:

#### [SNMP \[1302 ページ\]](#)

Simple Network Management Protocol (SNMP) は、ネットワーク管理に使用されている標準的なプロトコルです。

#### [SNMP のインストール \[1305 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用するには、コンピュータに SNMP をインストールする必要があります。

#### [SQL Anywhere SNMP Extension Agent の設定 \[1308 ページ\]](#)

SQL Anywhere SNMP Extension Agent では、1 つ以上のデータベースをモニタリングできます。

#### [SQL Anywhere SNMP Extension Agent が返す値 \[1310 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用して、データベースサーバプロパティ、データベースサーバ統計、データベースオプション、データベースプロパティ、およびデータベース統計を取得します。

#### [SQL Anywhere SNMP Extension Agent による値の設定 \[1311 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用すると、任意のデータベースオプション、一部のサーバプロパティ、1つのデータベースプロパティを設定できます。

#### [SQL Anywhere SNMP Extension Agent によるスタアドプロシージャの実行 \[1312 ページ\]](#)

SQL Anywhere MIB には、SQL Anywhere SNMP Extension Agent を使用してスタアドプロシージャを実行するための OID が格納されています。

#### [トラップ \[1312 ページ\]](#)

トラップは、特定のイベントが発生した場合に SNMP エージェントから送信される OID です。

#### [SQL Anywhere MIB リファレンス \[1316 ページ\]](#)

SNMP エージェントがサポートするオブジェクト識別子 (OID) のリストは、"Management Information Base (MIB)" と呼ばれるファイルに保存されています。このリストには、OID の名前や型などの情報も含まれています。

#### [RDBMS MIB リファレンス \[1343 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用して、多くの値の OID を検索することができます。

## 1.8.2.3.1 SNMP

Simple Network Management Protocol (SNMP) は、ネットワーク管理に使用されている標準的なプロトコルです。

SNMP では、マネージャとエージェントが通信できます。マネージャはエージェントに要求を送り、エージェントはマネージャからのクエリに回答します。エージェントは、特定のイベントが発生した場合にトラップと呼ばれる通知を使用してマネージャに通知することもできます。

SNMP エージェントは、管理対象オブジェクトに対する変数値の取得要求と設定要求を処理します。各変数には値が 1 つあります。この値は一般に文字列型または整数型ですが、他のデータ型の場合もあります。

変数は、1 つのグローバルな階層に保持されます。各変数には、その親の下でユニークとなる番号が割り当てられます。変数の完全名 (すべての親を含めた名前) は、オブジェクト識別子 (OID) と呼ばれます。SAP が所有するすべての OID は 1.3.6.1.4.1.897 で始まります。

エージェントがサポートする OID のリストは、**Management Information Base (MIB)** と呼ばれるファイルに保存されています。このリストには、OID の名前や型などの情報も含まれています。

MIB は、管理対象オブジェクトについてのネットワーク管理情報が格納されているデータベースです。MIB は、SQL Anywhere SNMP Extension Agent を使用してモニタリングする SQL Anywhere データベースとは別のものです。MIB オブジェクトの値は、SNMP を使用して変更または検索できます。MIB オブジェクトは 1 つの階層にまとめられており、この階層の最上位には、ネットワークについての最も一般的な情報が置かれています。SQL Anywhere SNMP Extension Agent は SQL Anywhere MIB および RDBMS MIB をサポートします。

このセクションの内容:

#### [SQL Anywhere MIB \[1303 ページ\]](#)

SQL Anywhere MIB は SQL Anywhere SNMP Extension Agent のために作成され、すべてのデータベースサーバの統計、プロパティ、およびすべてのデータベースの統計、プロパティ、およびオプションを含みます。

#### [RDBMS MIB \[1305 ページ\]](#)

RDBMS MIB は、特定のベンダーに依存しない、リレーショナルデータベース管理システム製品のための汎用 MIB (RFC 1697) です。

## 1.8.2.3.1.1 SQL Anywhere MIB

SQL Anywhere MIB は SQL Anywhere SNMP Extension Agent のために作成され、すべてのデータベースサーバの統計、プロパティ、およびすべてのデータベースの統計、プロパティ、およびオプションを含みます。

統計とプロパティはいくつかの例外を除いてすべて読み込み専用ですが、データベースのオプションはいずれも読み込みと書き込みの両方が可能です。

デフォルトでは、SQL Anywhere MIB は `%SQLANY17%\snmp\iAnywhere.mib` にあります。

SQL Anywhere MIB の階層は、以下の説明のとおりです。

OID	名前	説明
1.3.6.1.4.897.2.1.1.n.db	saServer.saSrvStat	データベース db のサーバ統計 n の値を返します。
1.3.6.1.4.897.2.1.2.n.db	saServer.saSrvProp	データベース db のサーバプロパティ n の値を返します。
1.3.6.1.4.897.2.2.1.n.db	saDb.saDbStat	データベース db のデータベース統計 n の値を返します。
1.3.6.1.4.897.2.2.2.n.db	saDb.saDbProp	データベース db のデータベースプロパティ n の値を返します。
1.3.6.1.4.897.2.2.3.n.db	saDb.saDbOpt	データベース db のデータベースオプション n の値を返す
1.3.6.1.4.897.2.3.1	saAgent.saVersion	SQL Anywhere SNMP Extension Agent のバージョンを返します。
1.3.6.1.4.897.2.3.2.db	saAgent.saDbConnStr	データベース db 用の接続文字列を返します。
1.3.6.1.4.897.2.3.3.db	saAgent.saConnected	SQL Anywhere SNMP Extension Agent がデータベース db に接続されているかどうかを示す値を返します。0 を設定すると、SQL Anywhere SNMP Extension Agent はデータベースとの接続を切断します。1 を設定すると、SQL Anywhere SNMP Extension Agent はデータベースに接続します。
1.3.6.1.4.897.2.3.4.db	saAgent.saStarted	データベース db が動作しているかどうかを返します。0 を設定すると、SQL Anywhere SNMP Extension Agent はデータベースを停止します。 <sup>1</sup> 1 を設定すると、データベースを起動しようとします。 <sup>2</sup>

OID	名前	説明
1.3.6.1.4.897.2.3.5.db	saAgent.saProc	文字列 <code>proc_name</code> を設定すると、SQL Anywhere Extension Agent はデータベース内のプロシージャ <code>proc_name</code> を実行します。  引数は <code>proc_name('string', 4)</code> などの形式で指定できます。引数を指定しなかった場合には、名前の後に空のカッコ ( ) が追加されます。値の取得を行った場合には、空の引用符 ("" ) が返されます。
1.3.6.1.4.897.2.3.6	saAgent.saRestart	この変数の値を 1 に設定すると、エージェントが自動的に再起動します (すべてのデータベースとの接続を切断し、.ini ファイルを再ロードします)。値の取得を行った場合には、0 が返されます。
1.3.6.1.4.897.2.3.7	saAgent.salnifile	SQL Anywhere Extension Agent が使用している <code>sasnmplib.ini</code> ファイルのフルパスを返します。
1.3.6.1.4.897.2.4	saMetaData	複数の仮想テーブル。各ローが SQL Anywhere MIB でサポートされる変数の 1 つを示します。

<sup>1</sup> この変数の値を設定することによってデータベースを停止すると、無条件で停止が実行されます。つまり、アクティブな接続がある場合でもデータベースは停止します。

<sup>2</sup> この変数の値を設定することによってデータベースを起動する場合は、接続文字列内に DBF パラメータを指定しておく必要があります (DBN と、必要な場合は DBKEY も含みます)。また、`sasnmplib.ini` ファイル内に `UtilDbPwd` フィールドを設定するか、サーバのデータベース起動権限 (-gd サーバオプションを使用して指定します) を all に設定することが必要です。

## asaMetaData テーブル

SQL Anywhere MIB には、サポートされている変数を調べるときに SQL Anywhere SNMP Extension Agent への問い合わせの手段として利用できるメタデータテーブルが含まれています。

### saSrvMetaData.saSrvStatMetaDataTable

データベースサーバ統計 (sa.saServer.saSrvStat に属する変数) を示します。

### saSrvMetaData.saSrvPropMetaDataTable

データベースサーバプロパティ (sa.saServer.saSrv.Prop に属する変数) を示します。

### saDbMetaData.saDbStatMetaDataTable

データベース統計 (sa.saDb.saDbStat に属する変数) を示します。

### saDbMetaData.saDbpropMetaDataTable

データベースプロパティ (sa.saDb.saDbProp に属する変数) を示します。

### saDbMetaData.saDbOptMetaDataTable

データベースオプション (sa.saDb.saDbOpt に属する変数) を示します。

## 関連情報

[SQL Anywhere MIB リファレンス \[1316 ページ\]](#)

[SQL Anywhere SNMP Extension Agent による値の設定 \[1311 ページ\]](#)

[asaMetaData テーブル \[1317 ページ\]](#)

### 1.8.2.3.1.2 RDBMS MIB

RDBMS MIB は、特定のベンダーに依存しない、リレーショナルデータベース管理システム製品のための汎用 MIB (RFC 1697) です。

RDBMS MIB は、仮想テーブルを使用して、サーバとデータベースについての情報を返します。ベース OID は 1.3.6.1.2.1.39 です。この MIB 内には 9 つの仮想テーブルがあります。SQL Anywhere SNMP Extension Agent は、これらの仮想テーブルのうちの 8 つをサポートします。

SQL Anywhere SNMP Extension Agent からは、RDBMS MIB 内のサポートされる変数に読み込み専用でアクセスできません。RDBMS MIB 内の変数を SQL Anywhere SNMP Extension Agent で書き込むことはできません。

仮想テーブルは、一定数の属性と、不特定の数のローで構成されています。テーブル内の要素は、GET 要求を使用して検索します。このとき、テーブルの OID にカラム番号とロー番号を追加したものを指定します。テーブル OID の後には、次のように 1 を追加する必要があります。

```
table.1.column.rownum
```

デフォルトでは、RDBMS MIB は `%SQLANY17%\snmp\RDBMS-MIB.mib` にあります。

### 1.8.2.3.2 SNMP のインストール

SQL Anywhere SNMP Extension Agent を使用するには、コンピュータに SNMP をインストールする必要があります。

デフォルトでは、SNMP は Windows にはインストールされません。

SNMP のインストールについては、オペレーティングシステムのマニュアルを参照してください。

コンピュータに SNMP をインストールした後は、コンピュータ上で SNMP サービスと SNMP トラップサービスが実行中であることが必要です。

SQL Anywhere をインストールする前に SNMP をインストールした場合は、SNMP サービスが SQL Anywhere SNMP Extension Agent を検出できるようにするため、SNMP サービスをいったん停止し、再起動する必要があります。SQL Anywhere をインストールした後で SNMP をインストールした場合は、SNMP サービスが自動的に SQL Anywhere SNMP Extension Agent を検出します。

このセクションの内容:

[SNMP サービスの再開 \[1306 ページ\]](#)

SQL Anywhere SNMP Extension Agent を検出できるように、SNMP サービスを再起動します。

[SNMP Extension Agent の再起動 \[1307 ページ\]](#)

SNMP Extension Agent を再起動

## 1.8.2.3.2.1 SNMP サービスの再開

SQL Anywhere SNMP Extension Agent を検出できるように、SNMP サービスを再起動します。

### 前提条件

コンピュータの管理者である必要があります。

### コンテキスト

SQL Anywhere をインストールするまえに SNMP をインストールした場合は、サービスを再起動する必要があります。

### 手順

1. 次のコマンドを実行して、SNMP サービスを停止します。

```
net stop snmp
```

2. 次のコマンドを実行して、SNMP サービスを開始します。

```
net start snmp
```

### 結果

SNMP サービスが再開されます。

## 1.8.2.3.2 SNMP Extension Agent の再起動

SNMP Extension Agent を再起動

### 前提条件

コンピュータの管理者である必要があります。

### コンテキスト

`sasnmplib.ini` ファイルの内容が変更されたときには、エージェントを再起動する必要があります。

### 手順

SNMP 管理ツールを使用して、`saAgent.saRestart` プロパティ (1.3.6.1.4.1.897.2.3.6) の値を 1 に変更します。

### 結果

SNMP Extension Agent への変更が有効になります。

### 次のステップ

ファイル非表示ユーティリティ (`dbfhide`) を使用して、`sasnmplib.ini` ファイルの内容を暗号化できます。その場合、ファイルの非暗号化コピーを保持し、暗号化ファイルを更新できるようにします。

### 関連情報

[.ini ファイルの内容を非表示にする \[566 ページ\]](#)

## 1.8.2.3.3 SQL Anywhere SNMP Extension Agent の設定

SQL Anywhere SNMP Extension Agent では、1 つ以上のデータベースをモニタリングできます。

モニタリング対象のデータベース情報は、次に示すフォーマットで `sasnmplib.ini` に保存されています。

```
[SAAgent]
TrapPollTime=time-in-seconds
[DBn]
ConnStr=connection-string
UtilDbPwd=utility-database-password
CacheTime=time-in-seconds
DBSpaceCacheTime=time-in-seconds
Trapt=trap-information
Disabled=1 or 0
```

`sasnmplib.ini` ファイルは、デフォルトで SQL Anywhere のインストール時に `%SQLANY17%\bin32` ディレクトリに格納されます。

ファイル非表示ユーティリティ (`dbfhide`) を使用して、`sasnmplib.ini` ファイルの内容を暗号化できます。

### SAAgent セクション

`sasnmplib.ini` ファイルの SAAgent セクションには、SQL Anywhere SNMP Extension Agent に関する情報があります。TrapPollTime プロパティが不要な場合は、セクション全体を省略できます。

#### TrapPollTime

この値は、動的トラップのポーリング頻度を指定します (動的トラップが指定されている場合)。SQL Anywhere SNMP Extension Agent は、デフォルトでは 5 秒おきに値をポーリングします。この値を 0 に設定すると、動的トラップが無効になります。このプロパティは省略可能です。

### DBn セクション

`sasnmplib.ini` ファイルの各 DBn セクションには、特定のデータベースの情報、そのデータベースへの接続方法、そのデータベース用の動的トラップを記述します。ファイルのプロパティ名は大文字と小文字を区別します。

n の値は、データベースを識別する番号です。この番号は、1 から開始する連番にする必要があります。不連続にすることはできません。たとえば、`sasnmplib.ini` ファイルにエントリ [DB1]、[DB2]、[DB4] がある場合、エントリ [DB3] が欠如しているためエントリ [DB4] は無視されます。

#### ConnStr

データベースへの接続に使用される接続文字列です。データベースに接続するには、情報を不足なく指定する必要があります。プロパティが必要です。

- ODBC データソースを使用してデータベースに接続するには、使用するデータソースが、ユーザデータソースではなく、システムデータソースである必要があります。
- 統合化ログインを使用するには、SNMP エージェントがサービスとして動作するため、SYSTEM アカウントにマッピングする必要があります。ただし、サービスとして動作するものはすべて、パスワードなしでデータベースに接続できま

す。別の方法として、サービスを実行するためのアカウントを変更し、その後でそのアカウントの統合化ログインを作成することもできます。

- 接続文字列の先頭には、文字列 `ASTART=NO; IDLE=0; CON=SNMP; ASTOP=NO` を付けます。この文字列に基づいて次の設定が行われます。
  - SQL Anywhere SNMP Extension Agent がデータベースサーバを自動的に起動するのを防ぐ
  - アイドルタイムアウトを無効にする (SQL Anywhere SNMP Extension Agent は一定時間アイドル状態になる可能性があるため)
  - 識別が可能なように接続に名前を付ける
  - SQL Anywhere SNMP Extension Agent が接続を停止するときにデータベースが停止しないようにする`sasnmplib.ini` ファイル内の接続文字列にこれらの値を指定すると、デフォルトの設定が無効にされ、`sasnmplib.ini` に指定した値が使用されます。

### UtilDbPwd

データベースを起動するように `sa.agent.saStarted` を設定すると、SQL Anywhere SNMP Extension Agent は、DBF パラメータ (データベースファイルの場所をデータベースサーバに指定するためのパラメータ) を使用してデータベースに接続しようとします。しかし、データベースの起動に必要な権限が DBA (ネットワークサーバのデフォルト。これは、パーソナルサーバとネットワークサーバの両方で、`-gd dba` オプションを使用して設定することもできます) の場合は、サーバで接続が許可されません。

このようなサーバでデータベースを起動するには、SNMP Extension Agent が SERVER OPERATOR システム権限でユーザとして、同じサーバですでに動作中のデータベースに接続する必要があります。この条件は、ユーティリティデータベースに接続することで満たすことができます。ユーティリティデータベースのパスワード (`-su` サーバオプションで指定します) を `sasnmplib.ini` ファイル内で指定した場合、SQL Anywhere Extension Agent は、データベースを起動するために、そのデータベースと同じサーバ上のユーティリティデータベースに接続し、START DATABASE 文を実行してから接続を切断します。このプロパティは省略可能です。

### CacheTime

データベースからデータを検索するときに、検索したデータを SQL Anywhere SNMP Extension Agent 内にキャッシュすることができます。これによって、それ以後に同じタイプのデータ (サーバプロパティやデータベース統計など) を検索するときには、データベースとのやりとりが不要になります。データをキャッシュすると、それ以降の検索ではデータをすばやく取得できますが、そのデータは最新の状態でない可能性があります。CacheTime プロパティは、キャッシュ時間を変更するために使用できます。値を 0 に設定すれば、キャッシュを無効にすることもできます。デフォルトでは、キャッシュ時間は 0 秒です。CacheTime パラメータを 0 に設定すると、取り出されるデータは常に最新となります。これは、要求ごとにデータがデータベースから検索されるためです。このプロパティは省略可能です。

### DBSpaceCacheTime

RDBMS MIB 内の `rdbmsDbLimitedResourceTable` には、DB 領域についての情報が格納されています。この情報も、データベースから検索されたときに SQL Anywhere SNMP Extension Agent 内にキャッシュすることができます。DB 領域情報のデフォルトのキャッシュ時間は 600 秒 (10 分) です。このプロパティは、キャッシュ時間を変更するために使用できます。値を 0 に設定すれば、キャッシュを無効にすることもできます。このプロパティは省略可能です。

### トラップ t

動的トラップを作成します。値 `t` は 1 から始まる正の整数である必要があります。数字を省略することはできません。このプロパティは省略可能です。

### 無効

SQL Anywhere SNMP Extension Agent は、このフィールドが 1 に設定されているデータベースエントリを無視します。このことを利用すると、SQL Anywhere SNMP Extension Agent の管理対象データベースのリストから特定のデータベ

ースを一時的に削除する場合、他のデータベースの番号を変更する必要がありません。このプロパティは省略可能です。

このファイルを編集した場合は、SNMP サービスを再起動するか、SQL Anywhere SNMP Extension Agent をリセットして、新しい設定が使用されるようにする必要があります。

## サンプル sasnmp.ini ファイル

SQL Anywhere SNMP Extension Agent の sasnmp.ini ファイルの例を以下に示します。

```
[SAAgent]
[DB1]
ConnStr=UID=DBA;PWD=passwd;DBN=sales;DBF=sales.db
Trap1=1.1.5 > 50000
UtilDbPwd=test
[DB2]
ConnStr=UID=DBA;PWD=passwd;DBN=field;DBF=field.db
UtilDbPwd=test
Disabled=1
[DB3]
ConnStr=UID=DBA;PWD=passwd;Host=host2;DBN=hq;DBF=hq.db
UtilDbPwd=test
```

SAAgent セクションにパラメータが指定されていないので、SQL Anywhere SNMP Extension Agent は 5 秒おきに値をポーリングします。

SQL Anywhere SNMP Extension Agent は、2 台のサーバで動作している 3 つの異なるデータベースをモニタリングします。データベース 3 は別のコンピュータで動作しているため、Host 接続パラメータが必要です。DB1 には、データベースサーバから送信されたデータが 50000 バイトを超えている場合に起動するトラップが指定されています。

## 関連情報

[動的トラップ \[1314 ページ\]](#)

[rdbmsDbLimitedResourceTable \[1346 ページ\]](#)

### 1.8.2.3.4 SQL Anywhere SNMP Extension Agent が返す値

SQL Anywhere SNMP Extension Agent を使用して、データベースサーバプロパティ、データベースサーバ統計、データベースオプション、データベースプロパティ、およびデータベース統計を取得します。

これらの値を取得する方法は、使用する SNMP 管理ソフトウェアによって異なります。

#### 例

次の表は、いくつかの OID の説明と、返される値 (それぞれの OID の値) の例を示します。

OID	説明	値の例
1.3.6.1.4.1.897.2.1.1.1.1	データベース 1 のサーバ統計 ActiveReq	1
1.3.6.1.4.1.897.2.2.1.4.1	データベース 1 のデータベース統計 CacheRead	11397
1.3.6.1.4.1.897.2.3.1	エージェントのバージョン	17.0.4(2459)
1.3.6.1.4.1.897.2.3.2.1	データベース 1 用の接続文字列	UID=DBA、PWD=passwd、Host=host1、DBN=sales、DBF=sales.db

## 関連情報

[SQL Anywhere MIB データベースサーバのプロパティ \[1323 ページ\]](#)

[SQL Anywhere MIB データベースサーバの統計 \[1320 ページ\]](#)

[SQL Anywhere MIB データベースオプション \[1337 ページ\]](#)

[SQL Anywhere MIB データベースプロパティ \[1332 ページ\]](#)

[SQL Anywhere MIB データベース統計 \[1329 ページ\]](#)

### 1.8.2.3.5 SQL Anywhere SNMP Extension Agent による値の設定

SQL Anywhere SNMP Extension Agent を使用すると、任意のデータベースオプション、一部のサーバプロパティ、1つのデータベースプロパティを設定できます。

SQL Anywhere SNMP Extension Agent は、SNMP の get クエリ、get-next クエリ、set クエリに応答します。

SQL Anywhere SNMP Extension Agent は、データベースオプションの設定時に、次の文を実行します。

```
SET OPTION PUBLIC.option-name = 'value'
```

データベースプロパティとサーバプロパティの設定には、sa\_server\_option システムプロシージャを使用します。

これらの値の設定方法は、使用する SNMP 管理ソフトウェアによって異なります。

SQL Anywhere SNMP Extension Agent に付属する SQL Anywhere MIB ファイルを使用して、多くのオプションやプロパティを設定することができます。

## 関連情報

[データベースオプション \[622 ページ\]](#)

[SQL Anywhere MIB リファレンス \[1316 ページ\]](#)

## 1.8.2.3.6 SQL Anywhere SNMP Extension Agent によるストアードプロシージャの実行

SQL Anywhere MIB には、SQL Anywhere SNMP Extension Agent を使用してストアードプロシージャを実行するための OID が格納されています。

ストアードプロシージャを実行するには、SQL Anywhere SNMP Extension Agent が接続に使用するユーザが以下に示す要件のいずれか 1 つを満たしている必要があります。

- プロシージャに対する EXECUTE 権限
- プロシージャの所有者である
- EXECUTE ANY PROCEDURE システム権限がある

プロシージャによって生成された結果セットや戻り値は無視されます。

SQL Anywhere SNMP Extension Agent を使用してストアードプロシージャを実行するには、`saAgent.saProc` (OID 1.3.6.1.4.1.897.2.3.5.db、db は `sasnmplib.ini` ファイル内のデータベース番号) の値としてストアードプロシージャの名前を示す文字列を設定します。必要に応じて、プロシージャに引数を指定できます。引数を指定しなかった場合には、プロシージャ名の後に空のカッコが追加されます。

たとえば、`saAgent.saProc` の値として文字列 `"pchin.updatesales('param1', 2)"` を設定すると、ユーザ `pchin` が所有するストアードプロシージャ `updatesales` が呼び出されます。

この OID の値としてプロシージャ名を設定する方法は、使用する SNMP 管理ソフトウェアによって異なります。

### 関連情報

[SQL Anywhere MIB \[1303 ページ\]](#)

## 1.8.2.3.7 トラップ

トラップは、特定のイベントが発生した場合に SNMP エージェントから送信される OID です。

トラップは SNMP エージェントで開始され、SNMP 管理ソフトウェアによって検出できます。検出後、SNMP 管理ソフトウェアは、イベントを直接処理することも、SNMP エージェントに詳細を問い合わせることもできます。

トラップを受信するには、SNMP サービスを設定する必要があります。SNMP サービスはトラップ情報を受信してこれを転送しますが、デフォルトではどこにも転送されず、動作中のトラップリスナは何も検出しません。

### SQL Anywhere SNMP Extension Agent トラップ

SQL Anywhere SNMP Extension Agent は、データベースサーバによって接続が停止されると、必ずトラップを送信します。このトラップの OID は 1.3.6.1.2.1.39.2.1 です。

データベースミラーリングを使用している場合に、SQL Anywhere SNMP Extension Agent とデータベースサーバとの接続が切断されると、SQL Anywhere SNMP Extension Agent は同じデータベースサーバへの再接続を 30 秒おきに試行します。再接続が成功しても、接続先が以前とは別のデータベースサーバであった場合 (ServerName プロパティによって確認できます)、SQL Anywhere SNMP Extension Agent は OID 1.3.6.1.4.1.897.2.6.3 のトラップと `sasnmplib.ini` ファイルから取得したデータベース ID を送信します。この場合、SQL Anywhere SNMP Extension Agent が接続していたプライマリデータベースサーバが停止したため、現在はミラーサーバがプライマリサーバとして動作しています。

SQL Anywhere SNMP Extension Agent が送信する他のトラップは、すべて動的トラップです。

このセクションの内容:

[トラップをコンピュータに送信するための SNMP サービスの設定 \[1313 ページ\]](#)

SNMP サービスを設定して、イベントに関する通知をトラップを使用してコンピュータに送信するように設定します。

[動的トラップ \[1314 ページ\]](#)

動的トラップは、特定のプロパティ、統計、またはオプションの値を含む単純な式の評価が true である場合に SQL Anywhere SNMP Extension Agent によって送信されるトラップです。

## 関連情報

[データベースミラーリング \[1660 ページ\]](#)

### 1.8.2.3.7.1 トラップをコンピュータに送信するための SNMP サービスの設定

SNMP サービスを設定して、イベントに関する通知をトラップを使用してコンピュータに送信するように設定します。

## 前提条件

コンピュータの管理者である必要があります。

## 手順

1. [マイコンピュータ](#)を右クリックし、[管理](#)をクリックします。
2. 左ウィンドウ枠で、[サービスとアプリケーション](#)をダブルクリックします。
3. 左ウィンドウ枠で、[サービス](#)をダブルクリックします。
4. 右ウィンドウ枠のサービスのリストで "SNMP Service" を見つけ、右クリックして、[プロパティ](#)をクリックします。
5. [トラップ](#)タブをクリックします。

6. 追加をクリックします。
7. SNMP サービスの構成ウィンドウで、テキストボックスに `localhost` と入力し、追加をクリックします。
8. OK をクリックします。

## 結果

コンピュータに通知が送信されるよう、トラップが SNMP サービスに追加されます。

### 1.8.2.3.7.2 動的トラップ

動的トラップは、特定のプロパティ、統計、またはオプションの値を含む単純な式の評価が true である場合に SQL Anywhere SNMP Extension Agent によって送信されるトラップです。

動的トラップは、`sasnmplib.ini` ファイル内に作成されます。`sasnmplib.ini` ファイルエントリ内のトラップ情報のフォーマットは次のとおりです。

```
Traptrapnum=[1.3.6.1.4.1.897.2.]oid[.dbnum] op value
```

#### trapnum

動的トラップの番号です。開始値が 1 の連番になっている必要があります。

#### oid

プロパティ、統計、またはオプションの OID です。SQL Anywhere MIB 内または RDBMS MIB 内の OID がサポートされます。指定された OID が有効な SQL Anywhere または RDBMS OID ではない場合は、SQL Anywhere MIB プレフィクス `1.3.6.1.4.1.897.2.` が先頭に追加されます。

#### i 注記

動的トラップでは、データベースサーバまたはデータベースのプロパティ、統計、またはオプションに対応する OID のみ使用できます。

#### dbnum

データベース番号です。このフィールドはオプションですが、指定する場合は `sasnmplib.ini` ファイルの `[DBn]` セクションのデータベース番号と一致する番号を指定する必要があります。

次のいずれかの値を指定します。

- `=` または `==` (等しい)
- `!=`、`<>`、または `><` (等しくない)
- `<=` または `=<` (以下)
- `>=` または `=>` (以上)
- `<` (より小さい)
- `>` (より大きい)

#### op

## i 注記

値が文字列の場合、等しいか等しくないかの評価のみがサポートされます。

### value

式内で使用する値です。文字列値は、一重引用符または二重引用符で囲む必要があります。これらの引用符は値の一部として扱われません。開始引用符または終了引用符を文字列に含める必要がある場合は、それらを二重に指定する必要があります。文字列内に出現する一重引用符は二重引用符にしません。

動的トラップを設定するとき、単位をキロバイト、メガバイト、ギガバイト、またはテラバイトで指定するには、それぞれ k、m、g、または t を使用します。たとえば、次のように指定することで、現在のキャッシュサイズが 200 MB を超えた場合にトラップがトリガするように、動的トラップを設定できます。

```
Trap1=1.3.6.1.4.1.897.2.1.1.11.1 > 200M
```

sasnmp.ini ファイル内には、任意の数の Trap フィールドを指定できます。トラップに使用される OID は 1.3.6.1.4.1.897.2.4.1 です。トラップとともに送信されるデータには次のものがあります。

- トラップ番号 (SQL Anywhere SNMP Agent によって送信される動的トラップの番号は開始値が 1 の連番)
- データベースインデックス
- データベース名トラップインデックス (sasnmp.ini ファイル内の値)
- 変数名
- 変数値 (変数の現在値。閾値になるとはかぎらない。)

## 動的トラップの動作

動的トラップは、いったんトリガされると、トリガを引き起こした条件が FALSE に変わり、その後再び TRUE になるまで、再送信されません。

たとえば、1.1.11.1 >= 51200K という式を使用して動的トラップを設定した場合、サーバのキャッシュサイズが 50 MB (51200 KB) に達すると、トラップがトリガされます。この時点で、この動的トラップは無効になり、それ以上送信されません。このトラップが再び有効になるのは、キャッシュサイズがその後 50 MB 未満になった場合だけです。この後でキャッシュサイズが 50 MB に戻ると、そのことが通知されます。

## トラップの例

トラップ情報	説明
Trap1=1.1.5 > 10000	サーバから送信されたデータが 10000 バイトを超えると、トラップが送信されます。
Trap2=1.3.6.1.2.1.39.1.4.1.4.14.1 = 10485760	トランザクションログファイルのサイズが 10 MB を超えると、トラップが送信されます。

## 関連情報

[SQL Anywhere MIB リファレンス \[1316 ページ\]](#)

[RDBMS MIB リファレンス \[1343 ページ\]](#)

### 1.8.2.3.8 SQL Anywhere MIB リファレンス

SNMP エージェントがサポートするオブジェクト識別子 (OID) のリストは、"Management Information Base (MIB)" と呼ばれるファイルに保存されています。このリストには、OID の名前や型などの情報も含まれています。

SQL Anywhere SNMP Extension Agent を使用して、多くの統計、プロパティ、オプションを検索および設定することができます。

このセクションの内容:

[Agent \[1317 ページ\]](#)

Agent テーブルには、SQL Anywhere SNMP Extension Agent の情報が含まれています。

[asaMetaData テーブル \[1317 ページ\]](#)

SQL Anywhere MIB には、複数のメタデータテーブルがあります。

[SQL Anywhere MIB データベースサーバの統計 \[1320 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用してデータベースサーバの統計を取得します。

[SQL Anywhere MIB データベースサーバのプロパティ \[1323 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用してデータベースサーバのプロパティを取得します。

[SQL Anywhere MIB データベース統計 \[1329 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用してデータベース統計を取得します。

[SQL Anywhere MIB データベースプロパティ \[1332 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用してデータベースプロパティを取得します。

[SQL Anywhere MIB データベースオプション \[1337 ページ\]](#)

SQL Anywhere SNMP Extension Agent を使用してデータベースオプションを取得します。

## 関連情報

[SNMP \[1302 ページ\]](#)

## 1.8.2.3.8.1 Agent

Agent テーブルには、SQL Anywhere SNMP Extension Agent の情報が含まれています。

書き込み可能なプロパティには、アスタリスク記号 (\*) が付いています。値 n は `sasnmplib.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.4.1.897.2.3.1	String	saVersion	エージェントのバージョン
1.3.6.1.4.1.897.2.3.2.n	String	saDBConnStr	接続文字列
1.3.6.1.4.1.897.2.3.3.n	Integer32	saConnected*	エージェントが接続されている場合は 1。それ以外の場合は 0。
1.3.6.1.4.1.897.2.3.4.n	Integer32	saStarted*	データベースが起動されている場合は 1。それ以外の場合は 0。
1.3.6.1.4.1.897.2.3.5.n	String	saProc*	" "
1.3.6.1.4.1.897.2.3.6	String	saRestart*	0

## 1.8.2.3.8.2 asaMetaData テーブル

SQL Anywhere MIB には、複数のメタデータテーブルがあります。

- `saSrvMetaData.saSrvStatMetaDataTable`
- `saSrvMetaData.saSrvPropMetaDataTable`
- `saSrvMetaData.saDbStatMetaDataTable`
- `saSrvMetaData.saDbPropMetaDataTable`
- `saSrvMetaData.saDbOptMetaDataTable`

このセクションの内容:

[saSrvMetaData.saSrvStatMetaDataTable \[1318 ページ\]](#)

データベースサーバ統計についてのメタデータが格納されています。

[saSrvMetaData.saSrvPropMetaDataTable \[1318 ページ\]](#)

データベースサーバプロパティについてのメタデータが格納されています。

[saDbMetaData.saDbStatMetaDataTable \[1319 ページ\]](#)

データベース統計についてのメタデータが格納されています。

[saDbMetaData.saDbPropMetaDataTable \[1319 ページ\]](#)

データベースプロパティについてのメタデータが格納されています。

[saDbMetaData.saDbOptMetaDataTable \[1320 ページ\]](#)

データベースオプションについてのメタデータが格納されています。

## 1.8.2.3.8.2.1 saSrvMetaData.saSrvStatMetaDataTable

データベースサーバ統計についてのメタデータが格納されています。

値 `db` は、`sasnmplib.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.4.1.897.2.4.1.1.1.1.db	Integer32	saSrvStatIndex	db
1.3.6.1.4.1.897.2.4.1.1.1.2.db	Integer32	saSrvStatObjType	1 <sup>1</sup>
1.3.6.1.4.1.897.2.4.1.1.1.3.db	Integer32	saSrvStatType	1 <sup>2</sup>
1.3.6.1.4.1.897.2.4.1.1.1.4.db	OID	saSrvStatOID	SQL Anywhere MIB エントリの OID <sup>3</sup>
1.3.6.1.4.1.897.2.4.1.1.1.5.db	文字列	saSrvStatName	統計名

<sup>1</sup> 値: 1=Server、2=Database

<sup>2</sup> 値: 1=Statistic、2=Property、3=Option

<sup>3</sup> 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## 1.8.2.3.8.2.2 saSrvMetaData.saSrvPropMetaDataTable

データベースサーバプロパティについてのメタデータが格納されています。

値 `db` は `sasnmplib.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.4.1.897.2.4.1.2.1.1.db	Integer32	saSrvPropIndex	db
1.3.6.1.4.1.897.2.4.1.2.1.2.db	Integer32	saSrvPropObjType	1 <sup>1</sup>
1.3.6.1.4.1.897.2.4.1.2.1.3.db	Integer32	saSrvPropType	2 <sup>2</sup>
1.3.6.1.4.1.897.2.4.1.2.1.4.db	OID	saSrvPropOID	SQL Anywhere MIB エントリの OID <sup>3</sup>
1.3.6.1.4.1.897.2.4.1.2.1.5.db	文字列	saSrvPropName	プロパティ名

<sup>1</sup> 値: 1=Server、2=Database

<sup>2</sup> 値: 1=Statistic、2=Property、3=Option

<sup>3</sup> 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## 1.8.2.3.8.2.3 saDbMetaData.saDbStatMetaDataTable

データベース統計についてのメタデータが格納されています。

値 `db` は `sasnmpr.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.4.1.897.2.4.2.1.1.1.db	Integer32	saDbStatIndex	db
1.3.6.1.4.1.897.2.4.2.1.1.2.db	Integer32	saDbStatObjType	2 <sup>1</sup>
1.3.6.1.4.1.897.2.4.2.1.1.3.db	Integer32	saDbStatType	1 <sup>2</sup>
1.3.6.1.4.1.897.2.4.2.1.1.4.db	OID	saDbStatOID	SQL Anywhere MIB エントリの OID <sup>3</sup>
1.3.6.1.4.1.897.2.4.2.1.1.5.db	文字列	saDbStatName	統計名

<sup>1</sup> 値: 1=Server、2=Database

<sup>2</sup> 値: 1=Statistic、2=Property、3=Option

<sup>3</sup> 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## 1.8.2.3.8.2.4 saDbMetaData.saDbPropMetaDataTable

データベースプロパティについてのメタデータが格納されています。

値 `db` は `sasnmpr.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.4.1.897.2.4.2.2.1.1.db	Integer32	saDbPropIndex	db
1.3.6.1.4.1.897.2.4.2.2.1.2.db	Integer32	saDbPropObjType	2 <sup>1</sup>
1.3.6.1.4.1.897.2.4.2.2.1.3.db	Integer32	saDbPropType	2 <sup>2</sup>
1.3.6.1.4.1.897.2.4.2.2.1.4.db	OID	saDbPropOID	SQL Anywhere MIB エントリの OID <sup>3</sup>
1.3.6.1.4.1.897.2.4.2.2.1.5.db	文字列	saDbPropName	プロパティ名

<sup>1</sup> 値: 1=Server、2=Database

<sup>2</sup> 値: 1=Statistic、2=Property、3=Option

<sup>3</sup> 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## 1.8.2.3.8.2.5 saDbMetaData.saDbOptMetaDataTable

データベースオプションについてのメタデータが格納されています。

値 `db` は `sasnmp.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.4.1.897.2.4.2.1.1.1.db	Integer32	saDbOptIndex	db
1.3.6.1.4.1.897.2.4.2.1.1.2.db	Integer32	saDbOptObjType	2 <sup>1</sup>
1.3.6.1.4.1.897.2.4.2.1.1.3.db	Integer32	saDbOptType	3 <sup>2</sup>
1.3.6.1.4.1.897.2.4.2.1.1.4.db	OID	saDbOptOID	SQL Anywhere MIB エントリの OID <sup>3</sup>
1.3.6.1.4.1.897.2.4.2.1.1.5.db	文字列	saDbOptName	オプション名

<sup>1</sup> 値: 1=Server、2=Database

<sup>2</sup> 値: 1=Statistic、2=Property、3=Option

<sup>3</sup> 返される OID にデータベース番号は含まれません。クエリ内でデータベース番号を使用するためには、あらかじめ OID にそのデータベース番号を追加しておく必要があります。

## 1.8.2.3.8.3 SQL Anywhere MIB データベースサーバの統計

SQL Anywhere SNMP Extension Agent を使用してデータベースサーバの統計を取得します。

書き込み可能なオプションには、アスタリスク記号 (\*) が付いています。値 `n` は、`sasnmp.ini` ファイル内のデータベース番号です。

### 説明

OID	型	名前	統計情報
1.3.6.1.4.1.897.2.1.1.1.n	Integer32	srvStatActiveReq	ActiveReq
1.3.6.1.4.1.897.2.1.1.2.n	Integer32	srvStatAvailIO	AvailIO
1.3.6.1.4.1.897.2.1.1.3.n	Counter64	srvStatBytesReceived	BytesReceived
1.3.6.1.4.1.897.2.1.1.4.n	Counter64	srvStatBytesReceivedUncomp	BytesReceivedUncomp
1.3.6.1.4.1.897.2.1.1.5.n	Counter64	srvStatBytesSent	BytesSent
1.3.6.1.4.1.897.2.1.1.6.n	Counter64	srvStatBytesSentUncomp	BytesSentUncomp
1.3.6.1.4.1.897.2.1.1.7.n	Counter64	srvStatCacheHits	CacheHits
1.3.6.1.4.1.897.2.1.1.8.n	Integer32	srvStatCachePinned	CachePinned

OID	型	名前	統計情報
1.3.6.1.4.1.897.2.1.1.9.n	Counter64	srvStatCacheRead	CacheRead
1.3.6.1.4.1.897.2.1.1.10.n	Counter64	srvStatCacheReplacements	CacheReplacements
1.3.6.1.4.1.897.2.1.1.11.n	Integer32	srvStatCurrentCacheSize	CurrentCacheSize
1.3.6.1.4.1.897.2.1.1.12.n	Counter64	srvStatDiskRead	DiskRead
1.3.6.1.4.1.897.2.1.1.13.n	Integer32	srvStatFreeBuffers	FreeBuffers
1.3.6.1.4.1.897.2.1.1.14.n	Integer32	srvStatInternal	Internal
1.3.6.1.4.1.897.2.1.1.15.n	Integer32	srvStatUniqueClientAddresses	UniqueClientAddresses
1.3.6.1.4.1.897.2.1.1.16.n	Integer32	srvStatLockedHeapPages	LockedHeapPages
1.3.6.1.4.1.897.2.1.1.17.n	Counter64	srvStatMainHeapBytes	MainHeapBytes
1.3.6.1.4.1.897.2.1.1.18.n	Integer32	srvStatMainHeapPages	MainHeapPages
1.3.6.1.4.1.897.2.1.1.19.n	Integer32	srvStatMapPhysicalMemoryEng	MapPhysicalMemoryEng
1.3.6.1.4.1.897.2.1.1.20.n	Integer32	srvStatMaxCacheSize	MaxCacheSize
1.3.6.1.4.1.897.2.1.1.21.n	Integer32	srvStatMinCacheSize	MinCacheSize
1.3.6.1.4.1.897.2.1.1.22.n	Counter64	srvStatMultiPacketsReceived	MultiPacketsReceived
1.3.6.1.4.1.897.2.1.1.23.n	Counter64	srvStatMultiPacketsSent	MultiPacketsSent
1.3.6.1.4.1.897.2.1.1.24.n	Counter64	srvStatPacketsReceived	PacketsReceived
1.3.6.1.4.1.897.2.1.1.25.n	Counter64	srvStatPacketsReceivedUncomp	PacketsReceivedUncomp
1.3.6.1.4.1.897.2.1.1.26.n	Counter64	srvStatPacketsSent	PacketsSent
1.3.6.1.4.1.897.2.1.1.27.n	Counter64	srvStatPacketsSentUncomp	PacketsSentUncomp
1.3.6.1.4.1.897.2.1.1.28.n	Integer32	srvStatPeakCacheSize	PeakCacheSize
1.3.6.1.4.1.897.2.1.1.29.n	Integer32	srvStatRemoteputWait	RemoteputWait
1.3.6.1.4.1.897.2.1.1.30.n	Integer32	srvStatReq	Req
1.3.6.1.4.1.897.2.1.1.31.n	Counter64	srvStatSendFail	SendFail
1.3.6.1.4.1.897.2.1.1.32.n	Integer32	srvStatTotalBuffers	TotalBuffers
1.3.6.1.4.1.897.2.1.1.33.n	Integer32	srvStatUnschReq	UnschReq
1.3.6.1.4.1.897.2.1.1.34.n	Integer32	srvStatInternal	Internal
1.3.6.1.4.1.897.2.1.1.35.n	Integer32	srvStatCacheFile	CacheFile
1.3.6.1.4.1.897.2.1.1.36.n	Integer32	srvStatCacheFileDirty	CacheFileDirty
1.3.6.1.4.1.897.2.1.1.37.n	Integer32	srvStatCacheAllocated	CacheAllocated
1.3.6.1.4.1.897.2.1.1.38.n	Integer32	srvStatCachePanics	CachePanics
1.3.6.1.4.1.897.2.1.1.39.n	Integer32	srvStatCacheFree	CacheFree
1.3.6.1.4.1.897.2.1.1.40.n	Integer32	srvStatCacheScavenges	CacheScavenges

OID	型	名前	統計情報
1.3.6.1.4.1.897.2.1.1.41.n	Integer32	srvStatCacheScavengeVisited	CacheScavengeVisited
1.3.6.1.4.1.897.2.1.1.42.n	Integer32	srvStatLockedCursorPages	LockedCursorPages
1.3.6.1.4.1.897.2.1.1.43.n	Integer32	srvStatQueryHeapPages	QueryHeapPages
1.3.6.1.4.1.897.2.1.1.44.n	Integer32	srvStatCarverHeapPages	CarverHeapPages
1.3.6.1.4.1.897.2.1.1.45.n	Integer32	srvStatHeapsRelocatable	HeapsRelocatable
1.3.6.1.4.1.897.2.1.1.46.n	Integer32	srvStatHeapsLocked	HeapsLocked
1.3.6.1.4.1.897.2.1.1.47.n	Integer32	srvStatHeapsQuery	HeapsQuery
1.3.6.1.4.1.897.2.1.1.48.n	Integer32	srvStatHeapsCarver	HeapsCarver
1.3.6.1.4.1.897.2.1.1.49.n	Integer32	srvStatMultiPageAllocs	MultiPageAllocs
1.3.6.1.4.1.897.2.1.1.50.n	Integer32	srvStatRequestsReceived	RequestsReceived
1.3.6.1.4.1.897.2.1.1.51.n	Integer32	srvStatExchangeTasks	ExchangeTasks
1.3.6.1.4.1.897.2.1.1.52.n	Integer32	srvStatClientStmtCacheHits	ClientStmtCacheHits
1.3.6.1.4.1.897.2.1.1.53.n	Integer32	srvStatClientStmtCacheMisses	ClientStmtCacheMisses
1.3.6.1.4.1.897.2.1.1.54.n	Integer32	srvStatQueryMemActiveCurr	QueryMemActiveCurr
1.3.6.1.4.1.897.2.1.1.55.n	Integer32	srvStatQueryMemActiveEst	QueryMemActiveEst
1.3.6.1.4.1.897.2.1.1.56.n	Integer32	srvStatQueryMemGrantWaiting	QueryMemGrantWaiting
1.3.6.1.4.1.897.2.1.1.57.n	Integer32	srvStatQueryMemGrantRequested	QueryMemGrantRequested
1.3.6.1.4.1.897.2.1.1.58.n	Integer32	srvStatQueryMemGrantWaited	QueryMemGrantWaited
1.3.6.1.4.1.897.2.1.1.59.n	Integer32	srvStatQueryMemGrantFailed	QueryMemGrantFailed
1.3.6.1.4.1.897.2.1.1.60.n	Integer32	srvStatQueryMemGrantGranted	QueryMemGrantGranted
1.3.6.1.4.1.897.2.1.1.61.n	Integer32	srvStatQueryMemExtraAvail	QueryMemExtraAvail
1.3.6.1.4.1.897.2.1.1.62.n	Integer32	srvStatThreadDeadlocksAvoided	ThreadDeadlocksAvoided
1.3.6.1.4.1.897.2.1.1.63.n	Integer32	srvStatThreadDeadlocksReported	ThreadDeadlocksReported
1.3.6.1.4.1.897.2.1.1.64.n	Integer32	srvStatCommit	Commit
1.3.6.1.4.1.897.2.1.1.65.n	Integer32	srvStatCursor	Cursor
1.3.6.1.4.1.897.2.1.1.66.n	Integer32	srvStatCursorOpen	CursorOpen
1.3.6.1.4.1.897.2.1.1.67.n	Integer32	srvStatRlbc	Rlbc
1.3.6.1.4.1.897.2.1.1.68.n	Integer32	srvStatPrepStmt	PrepStmt

OID	型	名前	統計情報
1.3.6.1.4.1.897.2.1.1.69.n	Integer32	srvStatParameterizationPrepareCount	ParameterizationPrepareCount
1.3.6.1.4.1.897.2.1.1.70.n	Integer32	srvStatConnCount	ConnCount
1.3.6.1.4.1.897.2.1.1.71.n	Integer32	srvStatCompletedReq	CompletedReq
1.3.6.1.4.1.897.2.1.1.72.n	Integer32	srvStatCurrRead	CurrRead
1.3.6.1.4.1.897.2.1.1.73.n	Integer32	srvStatCurrWrite	CurrWrite
1.3.6.1.4.1.897.2.1.1.74.n	Counter64	srvStatDiskWrite	DiskWrite
1.3.6.1.4.1.897.2.1.1.75.n	Integer32	srvStatCacheMisses	CacheMisses

## 関連情報

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

## 1.8.2.3.8.4 SQL Anywhere MIB データベースサーバのプロパティ

SQL Anywhere SNMP Extension Agent を使用してデータベースサーバのプロパティを取得します。

書き込み可能なオプションには、アスタリスク記号 (\*) が付いています。値 n は sasnmp.ini ファイル内のデータベース番号です。

## 説明

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.1.2.1.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.2.n	String	srvPropCharSet	CharSet
1.3.6.1.4.1.897.2.1.2.3.n	String	srvPropCommandLine	CommandLine
1.3.6.1.4.1.897.2.1.2.4.n	String	srvPropCompactPlatformVer	CompactPlatformVer
1.3.6.1.4.1.897.2.1.2.5.n	String	srvPropCompanyName	CompanyName
1.3.6.1.4.1.897.2.1.2.6.n	String	srvPropConnsDisabled*	ConnsDisabled
1.3.6.1.4.1.897.2.1.2.7.n	String	srvPropConsoleLogFile	ConsoleLogFile
1.3.6.1.4.1.897.2.1.2.8.n	String	srvPropDefaultCollation	DefaultCollation
1.3.6.1.4.1.897.2.1.2.9.n	String	srvPropIdleTimeout	IdleTimeout

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.1.2.10.n	String	srvPropIsIQ	IsIQ
1.3.6.1.4.1.897.2.1.2.11.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.12.n	String	srvPropIsNetworkServer	IsNetworkServer
1.3.6.1.4.1.897.2.1.2.13.n	String	srvPropIsRuntimeServer	IsRuntimeServer
1.3.6.1.4.1.897.2.1.2.14.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.15.n	String	srvPropLanguage	Language
1.3.6.1.4.1.897.2.1.2.16.n	String	srvPropLegalCopyright	LegalCopyright
1.3.6.1.4.1.897.2.1.2.17.n	String	srvPropLegalTrademarks	LegalTrademarks
1.3.6.1.4.1.897.2.1.2.18.n	String	srvPropLicenseCount	LicenseCount
1.3.6.1.4.1.897.2.1.2.19.n	String	srvPropLicensedCompany	LicensedCompany
1.3.6.1.4.1.897.2.1.2.20.n	String	srvPropLicensedUser	LicensedUser
1.3.6.1.4.1.897.2.1.2.21.n	String	srvPropLicenseType	LicenseType
1.3.6.1.4.1.897.2.1.2.22.n	String	srvPropLivenessTimeout*	LivenessTimeout
1.3.6.1.4.1.897.2.1.2.23.n	String	srvPropMachineName	MachineName
1.3.6.1.4.1.897.2.1.2.24.n	String	srvPropMaxMessage	MaxMessage
1.3.6.1.4.1.897.2.1.2.25.n	String	srvPropMessageWindowSize	MessageWindowSize
1.3.6.1.4.1.897.2.1.2.26.n	String	srvPropName	Name
1.3.6.1.4.1.897.2.1.2.27.n	String	srvPropNativeProcessorArchitecture	NativeProcessorArchitecture
1.3.6.1.4.1.897.2.1.2.28.n	String	srvPropNumPhysicalProcessors	NumPhysicalProcessors
1.3.6.1.4.1.897.2.1.2.29.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.30.n	String	srvPropOmnidIdentifier	OmnidIdentifier
1.3.6.1.4.1.897.2.1.2.31.n	String	srvPropPageSize	PageSize
1.3.6.1.4.1.897.2.1.2.32.n	String	srvPropPlatform	Platform
1.3.6.1.4.1.897.2.1.2.33.n	String	srvPropPlatformVer	PlatformVer
1.3.6.1.4.1.897.2.1.2.34.n	String	srvPropProcessCPU	ProcessCPU
1.3.6.1.4.1.897.2.1.2.35.n	String	srvPropProcessCPUSystem	ProcessCPUSystem
1.3.6.1.4.1.897.2.1.2.36.n	String	srvPropProcessCPUUser	ProcessCPUUser
1.3.6.1.4.1.897.2.1.2.37.n	String	srvPropProcessorArchitecture	ProcessorArchitecture
1.3.6.1.4.1.897.2.1.2.38.n	String	srvPropProductName	ProductName
1.3.6.1.4.1.897.2.1.2.39.n	String	srvPropProductVersion	ProductVersion
1.3.6.1.4.1.897.2.1.2.40.n	String	srvPropQuittingTime*	QuittingTime

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.1.2.41.n	String	srvPropRememberLastStatement*	RememberLastStatement
1.3.6.1.4.1.897.2.1.2.42.n	String	srvPropRequestFilterConn	RequestFilterConn
1.3.6.1.4.1.897.2.1.2.43.n	String	srvPropRequestFilterDB	RequestFilterDB
1.3.6.1.4.1.897.2.1.2.44.n	String	srvPropRequestLogFile*	RequestLogFile
1.3.6.1.4.1.897.2.1.2.45.n	String	srvPropRequestLogging*	RequestLogging
1.3.6.1.4.1.897.2.1.2.46.n	String	srvPropRequestLogMaxSize	RequestLogMaxSize
1.3.6.1.4.1.897.2.1.2.47.n	String	srvPropStartTime	StartTime
1.3.6.1.4.1.897.2.1.2.48.n	String	srvPropTempDir	TempDir
1.3.6.1.4.1.897.2.1.2.49.n	String	srvPropMultiProgrammingLevel	MultiProgrammingLevel
1.3.6.1.4.1.897.2.1.2.50.n	String	srvPropTimeZoneAdjustment	TimeZoneAdjustment
1.3.6.1.4.1.897.2.1.2.51.n	String	srvPropHttpPorts	HttpPorts
1.3.6.1.4.1.897.2.1.2.52.n	String	srvPropHttpsPorts	HttpsPorts
1.3.6.1.4.1.897.2.1.2.53.n	String	srvPropProfileFilterConn	ProfileFilterConn
1.3.6.1.4.1.897.2.1.2.54.n	String	srvPropProfileFilterUser	ProfileFilterUser
1.3.6.1.4.1.897.2.1.2.55.n	String	srvPropRequestLogNumFiles	RequestLogNumFiles
1.3.6.1.4.1.897.2.1.2.56.n	String	srvPropIsFipsAvailable	IsFipsAvailable
1.3.6.1.4.1.897.2.1.2.57.n	String	srvPropFipsMode	FipsMode
1.3.6.1.4.1.897.2.1.2.58.n	String	srvPropStartDBPermission	StartDBPermission
1.3.6.1.4.1.897.2.1.2.59.n	String	srvPropServerName	ServerName
1.3.6.1.4.1.897.2.1.2.60.n	String	srvPropRememberLastPlan	RememberLastPlan
1.3.6.1.4.1.897.2.1.2.61.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.62.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.63.n	String	srvPropRequestTiming	RequestTiming
1.3.6.1.4.1.897.2.1.2.64.n	String	srvPropCacheSizingStatistics	CacheSizingStatistics
1.3.6.1.4.1.897.2.1.2.65.n	String	srvPropConsoleLogMaxSize	ConsoleLogMaxSize
1.3.6.1.4.1.897.2.1.2.66.n	String	srvPropDebuggingInformation	DebuggingInformation
1.3.6.1.4.1.897.2.1.2.67.n	String	srvPropMessage	Message
1.3.6.1.4.1.897.2.1.2.68.n	String	srvPropMessageText	MessageText
1.3.6.1.4.1.897.2.1.2.69.n	String	srvPropMessageTime	MessageTime
1.3.6.1.4.1.897.2.1.2.70.n	String	srvPropIsRsaAvailable	IsRsaAvailable
1.3.6.1.4.1.897.2.1.2.71.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.72.n	String	srvPropMaxConnections	MaxConnections

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.1.2.73.n	String	srvPropNumLogicalProcessors	NumLogicalProcessors
1.3.6.1.4.1.897.2.1.2.74.n	String	srvPropNumLogicalProcessorsUsed	NumLogicalProcessorsUsed
1.3.6.1.4.1.897.2.1.2.75.n	String	srvPropNumPhysicalProcessorsUsed	NumPhysicalProcessorsUsed
1.3.6.1.4.1.897.2.1.2.76.n	String	srvPropDefaultNcharCollation	DefaultNcharCollation
1.3.6.1.4.1.897.2.1.2.77.n	String	srvPropCollectStatistics	CollectStatistics
1.3.6.1.4.1.897.2.1.2.78.n	String	srvPropFirstOption	FirstOption
1.3.6.1.4.1.897.2.1.2.79.n	String	srvPropLastOption	LastOption
1.3.6.1.4.1.897.2.1.2.80.n	String	srvPropLastConnectionProperty	LastConnectionProperty
1.3.6.1.4.1.897.2.1.2.81.n	String	srvPropLastDatabaseProperty	LastDatabaseProperty
1.3.6.1.4.1.897.2.1.2.82.n	String	srvPropLastServerProperty	LastServerProperty
1.3.6.1.4.1.897.2.1.2.83.n	String	srvPropWebClientLogging	WebClientLogging
1.3.6.1.4.1.897.2.1.2.84.n	String	srvPropWebClientLogFile	WebClientLogFile
1.3.6.1.4.1.897.2.1.2.85.n	String	srvPropHttpNumConnections	HttpNumConnections
1.3.6.1.4.1.897.2.1.2.86.n	String	srvPropHttpsNumConnections	HttpsNumConnections
1.3.6.1.4.1.897.2.1.2.87.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.88.n	String	srvPropHttpNumActiveReq	HttpNumActiveReq
1.3.6.1.4.1.897.2.1.2.89.n	String	srvPropHttpsNumActiveReq	HttpsNumActiveReq
1.3.6.1.4.1.897.2.1.2.90.n	String	srvPropHttpNumSessions	HttpNumSessions
1.3.6.1.4.1.897.2.1.2.91.n	String	srvPropQueryMemPages	QueryMemPages
1.3.6.1.4.1.897.2.1.2.92.n	String	srvPropQueryMemGrantBase	QueryMemGrantBase
1.3.6.1.4.1.897.2.1.2.93.n	String	srvPropQueryMemGrantBaseMI	QueryMemGrantBaseMI
1.3.6.1.4.1.897.2.1.2.94.n	String	srvPropQueryMemGrantExtra	QueryMemGrantExtra
1.3.6.1.4.1.897.2.1.2.95.n	String	srvPropQueryMemActiveMax	QueryMemActiveMax
1.3.6.1.4.1.897.2.1.2.96.n	String	srvPropQueryMemPercentOfCache	QueryMemPercentOfCache
1.3.6.1.4.1.897.2.1.2.97.n	String	srvPropMessageCategoryLimit	MessageCategoryLimit
1.3.6.1.4.1.897.2.1.2.98.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.99.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.100.n	String	srvProplsService	IsService

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.1.2.101.n	String	srvPropTcplpAddresses	TcplpAddresses
1.3.6.1.4.1.897.2.1.2.102.n	String	srvPropHttpAddresses	HttpAddresses
1.3.6.1.4.1.897.2.1.2.103.n	String	srvPropHttpsAddresses	HttpsAddresses
1.3.6.1.4.1.897.2.1.2.104.n	String	srvPropProcessID	ProcessID
1.3.6.1.4.1.897.2.1.2.105.n	String	srvPropRemoteCapability	RemoteCapability
1.3.6.1.4.1.897.2.1.2.106.n	String	srvPropMaxRemoteCapability	MaxRemoteCapability
1.3.6.1.4.1.897.2.1.2.107.n	String	srvPropEventTypeName	EventTypeName
1.3.6.1.4.1.897.2.1.2.108.n	String	srvPropEventTypeDef	EventTypeDef
1.3.6.1.4.1.897.2.1.2.109.n	String	srvPropMaxEventType	MaxEventType
1.3.6.1.4.1.897.2.1.2.110.n	String	srvPropIsPortableDevice	IsPortableDevice
1.3.6.1.4.1.897.2.1.2.111.n	String	srvPropIPAddressMonitorPeriod	IPAddressMonitorPeriod
1.3.6.1.4.1.897.2.1.2.112.n	String	srvPropServerEdition	ServerEdition
1.3.6.1.4.1.897.2.1.2.113.n	String	srvPropObjectType	ObjectType
1.3.6.1.4.1.897.2.1.2.114.n	String	srvPropCurrentMultiProgrammingLevel	CurrentMultiProgrammingLevel
1.3.6.1.4.1.897.2.1.2.115.n	String	srvPropMinMultiProgrammingLevel	MinMultiProgrammingLevel
1.3.6.1.4.1.897.2.1.2.116.n	String	srvPropMaxMultiProgrammingLevel	MaxMultiProgrammingLevel
1.3.6.1.4.1.897.2.1.2.117.n	String	srvPropAutoMultiProgrammingLevel	AutoMultiProgrammingLevel
1.3.6.1.4.1.897.2.1.2.118.n	String	srvPropAutoMultiProgrammingLevelStatistics	AutoMultiProgrammingLevelStatistics
1.3.6.1.4.1.897.2.1.2.119.n	String	srvPropApproximateCPUTime	ApproximateCPUTime
1.3.6.1.4.1.897.2.1.2.120.n	String	srvPropConnectedTime	ConnectedTime
1.3.6.1.4.1.897.2.1.2.121.n	String	srvPropReqCountUnscheduled	ReqCountUnscheduled
1.3.6.1.4.1.897.2.1.2.122.n	String	srvPropReqCountActive	ReqCountActive
1.3.6.1.4.1.897.2.1.2.123.n	String	srvPropReqCountBlockIO	ReqCountBlockIO
1.3.6.1.4.1.897.2.1.2.124.n	String	srvPropReqCountBlockLock	ReqCountBlockLock
1.3.6.1.4.1.897.2.1.2.125.n	String	srvPropReqCountBlockContention	ReqCountBlockContention
1.3.6.1.4.1.897.2.1.2.126.n	String	srvPropReqTimeUnscheduled	ReqTimeUnscheduled
1.3.6.1.4.1.897.2.1.2.127.n	String	srvPropReqTimeActive	ReqTimeActive
1.3.6.1.4.1.897.2.1.2.128.n	String	srvPropReqTimeBlockIO	ReqTimeBlockIO

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.1.2.129.n	String	srvPropReqTimeBlockLock	ReqTimeBlockLock
1.3.6.1.4.1.897.2.1.2.130.n	String	srvPropReqTimeBlockContent tion	ReqTimeBlockContent tion
1.3.6.1.4.1.897.2.1.2.131.n	String	srvPropLicenseKey	LicenseKey
1.3.6.1.4.1.897.2.1.2.132.n	String	srvPropDiskSandbox	DiskSandbox
1.3.6.1.4.1.897.2.1.2.133.n	String	srvPropProcessorAffinity	ProcessorAffinity
1.3.6.1.4.1.897.2.1.2.134.n	String	srvPropIsAesniAvailable	IsAesniAvailable
1.3.6.1.4.1.897.2.1.2.135.n	String	srvPropCurrentMirrorBackgr oundWorkers	CurrentMirrorBackgroundWo rkers
1.3.6.1.4.1.897.2.1.2.136.n	String	srvPropMaxMirrorBackgrou ndWorkers	MaxMirrorBackgroundWorker s
1.3.6.1.4.1.897.2.1.2.137.n	String	srvPropHasSecuredFeature	HasSecuredFeature
1.3.6.1.4.1.897.2.1.2.138.n	String	srvPropHasSecureFeatureKe y	HasSecureFeatureKey
1.3.6.1.4.1.897.2.1.2.139.n	String	srvPropTcpIpListeners	TcpIpListeners
1.3.6.1.4.1.897.2.1.2.140.n	String	srvPropHttpListeners	HttpListeners
1.3.6.1.4.1.897.2.1.2.141.n	String	srvPropHttpsListeners	HttpsListeners
1.3.6.1.4.1.897.2.1.2.142.n	String	srvPropODataAddresses	ODataAddresses
1.3.6.1.4.1.897.2.1.2.143.n	String	srvPropODataSecureAddress es	ODataSecureAddresses
1.3.6.1.4.1.897.2.1.2.144.n	String	srvPropTopologyAwareSched uling	TopologyAwareScheduling
1.3.6.1.4.1.897.2.1.2.145.n	String	srvPropSharedMemoryListen er	SharedMemoryListener
1.3.6.1.4.1.897.2.1.2.146.n	String	srvPropCockpitDB	CockpitDB
1.3.6.1.4.1.897.2.1.2.147.n	String	srvPropSingleJVM	SingleJVM
1.3.6.1.4.1.897.2.1.2.148.n	String	srvPropSingleCLR	SingleCLR
1.3.6.1.4.1.897.2.1.2.149.n	String	srvPropJavaVM	JavaVM
1.3.6.1.4.1.897.2.1.2.150.n	String	srvPropPlanStatisticsStored	PlanStatisticsStored
1.3.6.1.4.1.897.2.1.2.151.n	String	srvPropPlanStatisticsStored Max	PlanStatisticsStoredMax
1.3.6.1.4.1.897.2.1.2.152.n	String	srvPropPropertyHistoryList	PropertyHistoryList
1.3.6.1.4.1.897.2.1.2.153.n	String	srvPropPropertyHistoryListA ctual	PropertyHistoryListActual
1.3.6.1.4.1.897.2.1.2.154.n	String	srvPropPropertyHistorySize	PropertyHistorySize
1.3.6.1.4.1.897.2.1.2.155.n	String	srvPropPropertyHistorySizeB ytes	PropertyHistorySizeBytes
1.3.6.1.4.1.897.2.1.2.156.n	String	srvPropCockpitURL	CockpitURL

## 関連情報

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

### 1.8.2.3.8.5 SQL Anywhere MIB データベース統計

SQL Anywhere SNMP Extension Agent を使用してデータベース統計を取得します。

書き込み可能なオプションには、アスタリスク記号 (\*) が付いています。値 *n* は、`sasnmplib.ini` ファイル内のデータベース番号です。

## 説明

OID	型	名前	統計情報
1.3.6.1.4.1.897.2.2.1.1.n	Counter64	dbStatCacheHits	CacheHits
1.3.6.1.4.1.897.2.2.1.2.n	Integer32	dbStatCacheReadIndInt	CacheReadIndInt
1.3.6.1.4.1.897.2.2.1.3.n	Integer32	dbStatCacheReadIndLeaf	CacheReadIndLeaf
1.3.6.1.4.1.897.2.2.1.4.n	Counter64	dbStatCacheRead	CacheRead
1.3.6.1.4.1.897.2.2.1.5.n	Integer32	dbStatCacheReadTable	CacheReadTable
1.3.6.1.4.1.897.2.2.1.6.n	Integer32	dbStatChkpt	Chkpt
1.3.6.1.4.1.897.2.2.1.7.n	Integer32	dbStatChkptFlush	ChkptFlush
1.3.6.1.4.1.897.2.2.1.8.n	Integer32	dbStatChkptPage	ChkptPage
1.3.6.1.4.1.897.2.2.1.9.n	Integer32	dbStatCheckpointUrgency	CheckpointUrgency
1.3.6.1.4.1.897.2.2.1.10.n	Integer32	dbStatInternal	Internal
1.3.6.1.4.1.897.2.2.1.11.n	Integer32	dbStatInternal	Internal
1.3.6.1.4.1.897.2.2.1.12.n	Integer32	dbStatCheckpointLogCommitToDisk	CheckpointLogCommitToDisk
1.3.6.1.4.1.897.2.2.1.13.n	Integer32	dbStatCheckpointLogPagesInUse	CheckpointLogPagesInUse
1.3.6.1.4.1.897.2.2.1.14.n	Integer32	dbStatCheckpointLogPagesRelocated	CheckpointLogPagesRelocated
1.3.6.1.4.1.897.2.2.1.15.n	Integer32	dbStatCheckpointLogSavePreimage	CheckpointLogSavePreimage
1.3.6.1.4.1.897.2.2.1.16.n	Integer32	dbStatCheckpointLogSize	CheckpointLogSize
1.3.6.1.4.1.897.2.2.1.17.n	Integer32	dbStatCheckpointLogWrites	CheckpointLogWrites
1.3.6.1.4.1.897.2.2.1.18.n	Integer32	dbStatCheckpointLogPagesWritten	CheckpointLogPagesWritten

OID	型	名前	統計情報
1.3.6.1.4.1.897.2.2.1.19.n	Integer32	dbStatCommitFile	CommitFile
1.3.6.1.4.1.897.2.2.1.20.n	Integer32	dbStatConnCount	ConnCount
1.3.6.1.4.1.897.2.2.1.21.n	Integer32	dbStatCurrIO	CurrIO
1.3.6.1.4.1.897.2.2.1.22.n	Integer32	dbStatCurrRead	CurrRead
1.3.6.1.4.1.897.2.2.1.23.n	Integer32	dbStatCurrWrite	CurrWrite
1.3.6.1.4.1.897.2.2.1.24.n	Integer32	dbStatDiskReadIndInt	DiskReadIndInt
1.3.6.1.4.1.897.2.2.1.25.n	Integer32	dbStatDiskReadIndLeaf	DiskReadIndLeaf
1.3.6.1.4.1.897.2.2.1.26.n	Counter64	dbStatDiskRead	DiskRead
1.3.6.1.4.1.897.2.2.1.27.n	Integer32	dbStatDiskReadTable	DiskReadTable
1.3.6.1.4.1.897.2.2.1.28.n	Counter64	dbStatDiskWrite	DiskWrite
1.3.6.1.4.1.897.2.2.1.29.n	Integer32	dbStatExtendDB	ExtendDB
1.3.6.1.4.1.897.2.2.1.30.n	Integer32	dbStatExtendTempWrite	ExtendTempWrite
1.3.6.1.4.1.897.2.2.1.31.n	Integer32	dbStatFullCompare	FullCompare
1.3.6.1.4.1.897.2.2.1.32.n	Integer32	dbStatGetData	GetData
1.3.6.1.4.1.897.2.2.1.33.n	Integer32	dbStatIdleCheck	IdleCheck
1.3.6.1.4.1.897.2.2.1.34.n	Integer32	dbStatIdleChkpt	IdleChkpt
1.3.6.1.4.1.897.2.2.1.35.n	Integer32	dbStatIdleChkTime	IdleChkTime
1.3.6.1.4.1.897.2.2.1.36.n	Integer32	dbStatIdleWrite	IdleWrite
1.3.6.1.4.1.897.2.2.1.37.n	Integer32	dbStatIndAdd	IndAdd
1.3.6.1.4.1.897.2.2.1.38.n	Integer32	dbStatIndLookup	IndLookup
1.3.6.1.4.1.897.2.2.1.39.n	Integer32	dbStatIOToRecover	IOToRecover
1.3.6.1.4.1.897.2.2.1.40.n	Integer32	dbStatInternal	Internal
1.3.6.1.4.1.897.2.2.1.41.n	Integer32	dbStatInternal	Internal
1.3.6.1.4.1.897.2.2.1.42.n	Integer32	dbStatLockTablePages	LockTablePages
1.3.6.1.4.1.897.2.2.1.43.n	Integer32	dbStatInternal	Internal
1.3.6.1.4.1.897.2.2.1.44.n	Integer32	dbStatMaxIO	MaxIO
1.3.6.1.4.1.897.2.2.1.45.n	Integer32	dbStatMaxRead	MaxRead
1.3.6.1.4.1.897.2.2.1.46.n	Integer32	dbStatMaxWrite	MaxWrite
1.3.6.1.4.1.897.2.2.1.47.n	Counter64	dbStatPageRelocations	PageRelocations
1.3.6.1.4.1.897.2.2.1.48.n	Integer32	dbStatProcedurePages	ProcedurePages
1.3.6.1.4.1.897.2.2.1.49.n	Integer32	dbStatQueryCachePages	QueryCachePages
1.3.6.1.4.1.897.2.2.1.50.n	Integer32	dbStatQueryLowMemoryStrategy	QueryLowMemoryStrategy
1.3.6.1.4.1.897.2.2.1.51.n	Counter64	dbStatQueryRowsMaterialized	QueryRowsMaterialized

OID	型	名前	統計情報
1.3.6.1.4.1.897.2.2.1.52.n	Integer32	dbStatRecoveryUrgency	RecoveryUrgency
1.3.6.1.4.1.897.2.2.1.53.n	Integer32	dbStatLogFreeCommit	LogFreeCommit
1.3.6.1.4.1.897.2.2.1.54.n	Integer32	dbStatLogWrite	LogWrite
1.3.6.1.4.1.897.2.2.1.55.n	Integer32	dbStatRelocatableHeapPages	RelocatableHeapPages
1.3.6.1.4.1.897.2.2.1.56.n	Integer32	dbStatRollbackLogPages	RollbackLogPages
1.3.6.1.4.1.897.2.2.1.57.n	Integer32	dbStatTempTablePages	TempTablePages
1.3.6.1.4.1.897.2.2.1.58.n	Integer32	dbStatTriggerPages	TriggerPages
1.3.6.1.4.1.897.2.2.1.59.n	Integer32	dbStatViewPages	ViewPages
1.3.6.1.4.1.897.2.2.1.60.n	Integer32	dbStatVersionStorePages	VersionStorePages
1.3.6.1.4.1.897.2.2.1.61.n	Integer32	dbStatSnapshotCount	SnapshotCount
1.3.6.1.4.1.897.2.2.1.62.n	Integer32	dbStatLockCount	LockCount
1.3.6.1.4.1.897.2.2.1.63.n	Integer32	dbStatCacheReadWorkTable	CacheReadWorkTable
1.3.6.1.4.1.897.2.2.1.64.n	Integer32	dbStatDiskReadWorkTable	DiskReadWorkTable
1.3.6.1.4.1.897.2.2.1.65.n	Integer32	dbStatPrepares	Prepares
1.3.6.1.4.1.897.2.2.1.66.n	Integer32	dbStatConnPoolCachedCount	ConnPoolCachedCount
1.3.6.1.4.1.897.2.2.1.67.n	Integer32	dbStatConnPoolHits	ConnPoolHits
1.3.6.1.4.1.897.2.2.1.68.n	Integer32	dbStatConnPoolMisses	ConnPoolMisses
1.3.6.1.4.1.897.2.2.1.69.n	Integer32	dbStatHttpConnPoolCachedCount	HttpConnPoolCachedCount
1.3.6.1.4.1.897.2.2.1.70.n	Integer32	dbStatHttpConnPoolHits	HttpConnPoolHits
1.3.6.1.4.1.897.2.2.1.71.n	Integer32	dbStatHttpConnPoolMisses	HttpConnPoolMisses
1.3.6.1.4.1.897.2.2.1.72.n	Integer32	dbStatHttpConnPoolSteals	HttpConnPoolSteals
1.3.6.1.4.1.897.2.2.1.73.n	Integer32	dbStatMirrorServerWaits	MirrorServerWaits
1.3.6.1.4.1.897.2.2.1.74.n	Counter64	dbStatBytesReceived	BytesReceived
1.3.6.1.4.1.897.2.2.1.75.n	Counter64	dbStatBytesReceivedUncomp	BytesReceivedUncomp
1.3.6.1.4.1.897.2.2.1.76.n	Counter64	dbStatBytesSent	BytesSent
1.3.6.1.4.1.897.2.2.1.77.n	Counter64	dbStatBytesSentUncomp	BytesSentUncomp
1.3.6.1.4.1.897.2.2.1.78.n	Integer32	dbStatCarverHeapPages	CarverHeapPages
1.3.6.1.4.1.897.2.2.1.79.n	Integer32	dbStatClientStmtCacheHits	ClientStmtCacheHits
1.3.6.1.4.1.897.2.2.1.80.n	Integer32	dbStatClientStmtCacheMisses	ClientStmtCacheMisses
1.3.6.1.4.1.897.2.2.1.81.n	Integer32	dbStatCommit	Commit
1.3.6.1.4.1.897.2.2.1.82.n	Integer32	dbStatCursor	Cursor

OID	型	名前	統計情報
1.3.6.1.4.1.897.2.2.1.83.n	Integer32	dbStatHeapsCarver	HeapsCarver
1.3.6.1.4.1.897.2.2.1.84.n	Integer32	dbStatHeapsLocked	HeapsLocked
1.3.6.1.4.1.897.2.2.1.85.n	Integer32	dbStatHeapsQuery	HeapsQuery
1.3.6.1.4.1.897.2.2.1.86.n	Integer32	dbStatHeapsRelocatable	HeapsRelocatable
1.3.6.1.4.1.897.2.2.1.87.n	Integer32	dbStatCursorOpen	CursorOpen
1.3.6.1.4.1.897.2.2.1.88.n	Counter64	dbStatPacketsReceived	PacketsReceived
1.3.6.1.4.1.897.2.2.1.89.n	Counter64	dbStatPacketsReceivedUncomp	PacketsReceivedUncomp
1.3.6.1.4.1.897.2.2.1.90.n	Counter64	dbStatPacketsSent	PacketsSent
1.3.6.1.4.1.897.2.2.1.91.n	Counter64	dbStatPacketsSentUncomp	PacketsSentUncomp
1.3.6.1.4.1.897.2.2.1.92.n	Integer32	dbStatQueryHeapPages	QueryHeapPages
1.3.6.1.4.1.897.2.2.1.93.n	Integer32	dbStatQueryMemActiveCurr	QueryMemActiveCurr
1.3.6.1.4.1.897.2.2.1.94.n	Integer32	dbStatQueryMemGrantFailed	QueryMemGrantFailed
1.3.6.1.4.1.897.2.2.1.95.n	Integer32	dbStatQueryMemGrantGranted	QueryMemGrantGranted
1.3.6.1.4.1.897.2.2.1.96.n	Integer32	dbStatQueryMemGrantRequested	QueryMemGrantRequested
1.3.6.1.4.1.897.2.2.1.97.n	Integer32	dbStatQueryMemGrantWaited	QueryMemGrantWaited
1.3.6.1.4.1.897.2.2.1.98.n	Integer32	dbStatQueryMemGrantWaiting	QueryMemGrantWaiting
1.3.6.1.4.1.897.2.2.1.99.n	Integer32	dbStatRequestsReceived	RequestsReceived
1.3.6.1.4.1.897.2.2.1.100.n	Integer32	dbStatRlbc	Rlbc
1.3.6.1.4.1.897.2.2.1.101.n	Integer32	dbStatPrepStmt	PrepStmt
1.3.6.1.4.1.897.2.2.1.102.n	Integer32	dbStatParameterizationPrepareCount	ParameterizationPrepareCount

## 関連情報

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

### 1.8.2.3.8.6 SQL Anywhere MIB データベースプロパティ

SQL Anywhere SNMP Extension Agent を使用してデータベースプロパティを取得します。

書き込み可能なオプションには、アスタリスク記号 (\*) が付いています。値 n は、sasnmplib.ini ファイル内のデータベース番号です。

## 説明

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.2.2.1.n	String	dbPropAlias	Alias
1.3.6.1.4.1.897.2.2.2.2.n	String	dbPropAuditingTypes	AuditingTypes
1.3.6.1.4.1.897.2.2.2.3.n	String	dbPropBlankPadding	BlankPadding
1.3.6.1.4.1.897.2.2.2.4.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.5.n	String	dbPropCapabilities	Capabilities
1.3.6.1.4.1.897.2.2.2.6.n	String	dbPropCaseSensitive	CaseSensitive
1.3.6.1.4.1.897.2.2.2.7.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.8.n	String	dbPropCharSet	CharSet
1.3.6.1.4.1.897.2.2.2.9.n	String	dbPropChecksum	Checksum
1.3.6.1.4.1.897.2.2.2.10.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.11.n	String	dbPropCollation	Collation
1.3.6.1.4.1.897.2.2.2.12.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.13.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.14.n	String	dbPropCurrentRedoPos	CurrentRedoPos
1.3.6.1.4.1.897.2.2.2.15.n	String	dbPropDBFileFragments	DBFileFragments
1.3.6.1.4.1.897.2.2.2.16.n	String	dbPropDriveType	DriveType
1.3.6.1.4.1.897.2.2.2.17.n	String	dbPropEncryption	Encryption
1.3.6.1.4.1.897.2.2.2.18.n	String	dbPropFile	File
1.3.6.1.4.1.897.2.2.2.19.n	String	dbPropFileSize	FileSize
1.3.6.1.4.1.897.2.2.2.20.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.21.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.22.n	String	dbPropFreePages	FreePages
1.3.6.1.4.1.897.2.2.2.23.n	String	dbPropGlobalDBID	GlobalDBID
1.3.6.1.4.1.897.2.2.2.24.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.25.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.26.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.27.n	String	dbPropIQStore	IQStore
1.3.6.1.4.1.897.2.2.2.28.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.29.n	String	dbPropLanguage	Language
1.3.6.1.4.1.897.2.2.2.30.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.31.n	String	dbPropLogFileFragments	LogFileFragments
1.3.6.1.4.1.897.2.2.2.32.n	String	dbPropLogMirrorName	LogMirrorName

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.2.2.33.n	String	dbPropLogName	LogName
1.3.6.1.4.1.897.2.2.2.34.n	String	dbPropLTMGeneration	LTMGeneration
1.3.6.1.4.1.897.2.2.2.35.n	String	dbPropLTMTrunc	LTMTrunc
1.3.6.1.4.1.897.2.2.2.36.n	String	dbPropMultiByteCharSet	MultiByteCharSet
1.3.6.1.4.1.897.2.2.2.37.n	String	dbPropName	Name
1.3.6.1.4.1.897.2.2.2.38.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.39.n	String	dbPropPageSize	PageSize
1.3.6.1.4.1.897.2.2.2.40.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.41.n	String	dbPropProcedureProfiling*	ProcedureProfiling
1.3.6.1.4.1.897.2.2.2.42.n	String	dbPropReadOnly	ReadOnly
1.3.6.1.4.1.897.2.2.2.43.n	String	dbPropRemoteTrunc	RemoteTrunc
1.3.6.1.4.1.897.2.2.2.44.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.45.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.46.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.47.n	String	dbPropSyncTrunc	SyncTrunc
1.3.6.1.4.1.897.2.2.2.48.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.49.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.50.n	String	dbPropTempFileName	TempFileName
1.3.6.1.4.1.897.2.2.2.51.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.52.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.53.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.54.n	String	dbPropNextScheduleTime	NextScheduleTime
1.3.6.1.4.1.897.2.2.2.55.n	String	dbPropIdentitySignature	IdentitySignature
1.3.6.1.4.1.897.2.2.2.56.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.57.n	String	dbPropSnapshotIsolationState	SnapshotIsolationState
1.3.6.1.4.1.897.2.2.2.58.n	String	dbPropConnsDisabled	ConnsDisabled
1.3.6.1.4.1.897.2.2.2.59.n	String	dbPropPartnerState	PartnerState
1.3.6.1.4.1.897.2.2.2.60.n	String	dbPropArbiterState	ArbiterState
1.3.6.1.4.1.897.2.2.2.61.n	String	dbPropMirrorState	MirrorState
1.3.6.1.4.1.897.2.2.2.62.n	String	dbPropAlternateServerName	AlternateServerName
1.3.6.1.4.1.897.2.2.2.63.n	String	dbPropEncryptionScope	EncryptionScope
1.3.6.1.4.1.897.2.2.2.64.n	String	dbPropNcharCharSet	NcharCharSet
1.3.6.1.4.1.897.2.2.2.65.n	String	dbPropNcharCollation	NcharCollation

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.2.2.66.n	String	dbPropAccentSensitive	AccentSensitive
1.3.6.1.4.1.897.2.2.2.67.n	String	dbPropSendingTracingTo	SendingTracingTo
1.3.6.1.4.1.897.2.2.2.68.n	String	dbPropReceivingTracingFrom	ReceivingTracingFrom
1.3.6.1.4.1.897.2.2.2.69.n	String	dbPropIOParallelism	IOParallelism
1.3.6.1.4.1.897.2.2.2.70.n	String	dbPropJavaVM	JavaVM
1.3.6.1.4.1.897.2.2.2.71.n	String	dbPropDatabaseCleaner	DatabaseCleaner
1.3.6.1.4.1.897.2.2.2.72.n	String	dbPropHasCollationTailoring	HasCollationTailoring
1.3.6.1.4.1.897.2.2.2.73.n	String	dbPropCatalogCollation	CatalogCollation
1.3.6.1.4.1.897.2.2.2.74.n	String	dbPropHasEndianSwapFix	HasEndianSwapFix
1.3.6.1.4.1.897.2.2.2.75.n	String	dbPropAlternateMirrorServerName	AlternateMirrorServerName
1.3.6.1.4.1.897.2.2.2.76.n	String	dbPropOptionWatchList	OptionWatchList
1.3.6.1.4.1.897.2.2.2.77.n	String	dbPropOptionWatchAction	OptionWatchAction
1.3.6.1.4.1.897.2.2.2.78.n	String	dbPropMirrorMode	MirrorMode
1.3.6.1.4.1.897.2.2.2.79.n	String	dbPropHasNCHARLegacyCollationFix	HasNCHARLegacyCollationFix
1.3.6.1.4.1.897.2.2.2.80.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.81.n	String	dbPropAuthenticated	Authenticated
1.3.6.1.4.1.897.2.2.2.82.n	String	dbPropSynchronizationSchemaChangeActive	SynchronizationSchemaChangeActive
1.3.6.1.4.1.897.2.2.2.83.n	String	dbPropLastCheckpointTime	LastCheckpointTime
1.3.6.1.4.1.897.2.2.2.84.n	String	dbPropMirrorServerState	MirrorServerState
1.3.6.1.4.1.897.2.2.2.85.n	String	dbPropDriveBus	DriveBus
1.3.6.1.4.1.897.2.2.2.86.n	String	dbPropDriveModel	DriveModel
1.3.6.1.4.1.897.2.2.2.87.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.88.n	String	dbPropMirrorRole	MirrorRole
1.3.6.1.4.1.897.2.2.2.89.n	String	dbPropWriteChecksum	WriteChecksum
1.3.6.1.4.1.897.2.2.2.90.n	String	dbPropApproximateCPUTime	ApproximateCPUTime
1.3.6.1.4.1.897.2.2.2.91.n	String	dbPropConnectedTime	ConnectedTime
1.3.6.1.4.1.897.2.2.2.92.n	String	dbPropReqCountUnscheduled	ReqCountUnscheduled
1.3.6.1.4.1.897.2.2.2.93.n	String	dbPropReqCountActive	ReqCountActive
1.3.6.1.4.1.897.2.2.2.94.n	String	dbPropReqCountBlockIO	ReqCountBlockIO
1.3.6.1.4.1.897.2.2.2.95.n	String	dbPropReqCountBlockLock	ReqCountBlockLock
1.3.6.1.4.1.897.2.2.2.96.n	String	dbPropReqCountBlockContention	ReqCountBlockContention

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.2.2.97.n	String	dbPropReqTimeUnscheduled	ReqTimeUnscheduled
1.3.6.1.4.1.897.2.2.2.98.n	String	dbPropReqTimeActive	ReqTimeActive
1.3.6.1.4.1.897.2.2.2.99.n	String	dbPropReqTimeBlockIO	ReqTimeBlockIO
1.3.6.1.4.1.897.2.2.2.100.n	String	dbPropReqTimeBlockLock	ReqTimeBlockLock
1.3.6.1.4.1.897.2.2.2.101.n	String	dbPropReqTimeBlockContention	ReqTimeBlockContention
1.3.6.1.4.1.897.2.2.2.102.n	String	dbPropCopyNodeParent	CopyNodeParent
1.3.6.1.4.1.897.2.2.2.103.n	String	dbPropLastCommitRedoPos	LastCommitRedoPos
1.3.6.1.4.1.897.2.2.2.104.n	String	dbPropLastWrittenRedoPos	LastWrittenRedoPos
1.3.6.1.4.1.897.2.2.2.105.n	String	dbPropLastSyncedRedoPos	LastSyncedRedoPos
1.3.6.1.4.1.897.2.2.2.106.n	String	dbPropUTCimestampCatalog	UTCimestampCatalog
1.3.6.1.4.1.897.2.2.2.107.n	String	dbPropDiskSandbox	DiskSandbox
1.3.6.1.4.1.897.2.2.2.108.n	String	dbPropTimeWithoutClientConnection	TimeWithoutClientConnection
1.3.6.1.4.1.897.2.2.2.109.n	String	dbPropBackupInProgress	BackupInProgress
1.3.6.1.4.1.897.2.2.2.110.n	String	dbPropIQPointInTimeRecoveryLogStatus	IQPointInTimeRecoveryLogStatus
1.3.6.1.4.1.897.2.2.2.111.n	String	dbPropPointInTimeRecovery	PointInTimeRecovery
1.3.6.1.4.1.897.2.2.2.112.n	String	dbPropHasQDADTTFeature	HasQDADTTFeature
1.3.6.1.4.1.897.2.2.2.113.n	String	dbPropIQPointInTimeRecoveryLogFile	IQPointInTimeRecoveryLogFilename
1.3.6.1.4.1.897.2.2.2.114.n	String	dbPropMaxConnections	MaxConnections
1.3.6.1.4.1.897.2.2.2.115.n	String	dbPropTimeZone	TimeZone
1.3.6.1.4.1.897.2.2.2.116.n	String	dbPropCurrentTimeZoneOffset	CurrentTimeZoneOffset
1.3.6.1.4.1.897.2.2.2.117.n	String	dbPropCurrentTimelineID	CurrentTimelineID
1.3.6.1.4.1.897.2.2.2.118.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.119.n	String	dbPropPreviousTimelineID	PreviousTimelineID
1.3.6.1.4.1.897.2.2.2.120.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.121.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.122.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.123.n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.124.n	String	dbPropIdentitySignatureUUID	IdentitySignatureUUID
1.3.6.1.4.1.897.2.2.2.125.n	String	dbPropPlanStatisticsStored	PlanStatisticsStored
1.3.6.1.4.1.897.2.2.2.126.n	String	dbPropCurrentTimelineSignature	CurrentTimelineSignature

OID	型	名前	プロパティ
1.3.6.1.4.1.897.2.2.2.127.n	String	dbPropPropertyHistoryList	PropertyHistoryList

## 関連情報

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

### 1.8.2.3.8.7 SQL Anywhere MIB データベースオプション

SQL Anywhere SNMP Extension Agent を使用してデータベースオプションを取得します。

書き込み可能なオプションには、アスタリスク記号 (\*) が付いています。値 n は sasnmp.ini ファイル内のデータベース番号です。

## 説明

OID	型	名前	オプション
1.3.6.1.4.1.897.2.2.3.1.n	String	dbOptAllowNullsByDefault*	allow_nulls_by_default
1.3.6.1.4.1.897.2.2.3.2.n	String	dbOptAnsiNull*	ansi_null
1.3.6.1.4.1.897.2.2.3.3.n	String	dbOptAnsiBlanks*	ansi_blanks
1.3.6.1.4.1.897.2.2.3.4.n	String	dbOptAnsiCloseCursorsOnRollback*	ansi_close_cursors_on_rollback
1.3.6.1.4.1.897.2.2.3.5.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.6.n	String	dbOptAnsiPermissions*	ansi_permissions
1.3.6.1.4.1.897.2.2.3.7.n	String	dbOptAnsiUpdateConstraints*	ansi_update_constraints
1.3.6.1.4.1.897.2.2.3.8.n	String	dbOptAuditing*	auditing
1.3.6.1.4.1.897.2.2.3.9.n	String	dbOptAuditingOptions*	auditing_options
1.3.6.1.4.1.897.2.2.3.10.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.11.n	String	dbOptBackgroundPriority*	background_priority
1.3.6.1.4.1.897.2.2.3.12.n	String	dbOptBlocking*	blocking
1.3.6.1.4.1.897.2.2.3.13.n	Integer32	dbOptBlockingTimeout*	blocking_timeout
1.3.6.1.4.1.897.2.2.3.14.n	String	dbOptChained*	chained
1.3.6.1.4.1.897.2.2.3.15.n	Integer32	dbOptCheckpointTime*	checkpoint_time

OID	型	名前	オプション
1.3.6.1.4.1.897.2.2.3.16.n	Integer32	dbOptCisOption*	cis_option
1.3.6.1.4.1.897.2.2.3.17.n	Integer32	dbOptCisRowsetSize*	cis_rowset_size
1.3.6.1.4.1.897.2.2.3.18.n	String	dbOptCloseOnEndtrans*	close_on_endtrans
1.3.6.1.4.1.897.2.2.3.19.n	String	dbOptConnectionAuthentication*	connection_authentication
1.3.6.1.4.1.897.2.2.3.20.n	String	dbOptContinueAfterRaiserror*	continue_after_raiserror
1.3.6.1.4.1.897.2.2.3.21.n	String	dbOptConversionError*	conversion_error
1.3.6.1.4.1.897.2.2.3.22.n	String	dbOptCooperativeCommits*	cooperative_commits
1.3.6.1.4.1.897.2.2.3.23.n	Integer32	dbOptCooperativeCommitTimeout*	cooperative_commit_timeout
1.3.6.1.4.1.897.2.2.3.24.n	String	dbOptDatabaseAuthentication*	database_authentication
1.3.6.1.4.1.897.2.2.3.25.n	String	dbOptDateFormat*	date_format
1.3.6.1.4.1.897.2.2.3.26.n	String	dbOptDateOrder*	date_order
1.3.6.1.4.1.897.2.2.3.27.n	String	dbOptDebugMessages*	debug_messages
1.3.6.1.4.1.897.2.2.3.28.n	String	dbOptDedicatedTask*	dedicated_task
1.3.6.1.4.1.897.2.2.3.29.n	Integer32	dbOptDefaultTimestampIncrement*	default_timestamp_increment
1.3.6.1.4.1.897.2.2.3.30.n	String	dbOptDelayedCommits*	delayed_commits
1.3.6.1.4.1.897.2.2.3.31.n	Integer32	dbOptDelayedCommitTimeout*	delayed_commit_timeout
1.3.6.1.4.1.897.2.2.3.32.n	String	dbOptDivideByZeroError*	divide_by_zero_error
1.3.6.1.4.1.897.2.2.3.33.n	String	dbOptEscapeCharacter*	escape_character
1.3.6.1.4.1.897.2.2.3.34.n	String	dbOptExcludeOperators*	exclude_operators
1.3.6.1.4.1.897.2.2.3.35.n	String	dbOptExtendedJoinSyntax*	extended_join_syntax
1.3.6.1.4.1.897.2.2.3.36.n	String	dbOptFireTriggers*	fire_triggers
1.3.6.1.4.1.897.2.2.3.37.n	Integer32	dbOptFirstDayOfWeek*	first_day_of_week
1.3.6.1.4.1.897.2.2.3.38.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.39.n	String	dbOptForceViewCreation*	force_view_creation
1.3.6.1.4.1.897.2.2.3.40.n	String	dbOptForXmlNullTreatment*	for_xml_null_treatment
1.3.6.1.4.1.897.2.2.3.41.n	Integer32	dbOptGlobalDatabaseId*	global_database_id
1.3.6.1.4.1.897.2.2.3.42.n	Integer32	dbOptIsolationLevel*	isolation_level
1.3.6.1.4.1.897.2.2.3.43.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.44.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.45.n	Integer32	dbOptInternal	Internal

OID	型	名前	オプション
1.3.6.1.4.1.897.2.2.3.46.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.47.n	String	dbOptLockRejectedRows*	lock_rejected_rows
1.3.6.1.4.1.897.2.2.3.48.n	String	dbOptLoginMode*	login_mode
1.3.6.1.4.1.897.2.2.3.49.n	String	dbOptLoginProcedure*	login_procedure
1.3.6.1.4.1.897.2.2.3.50.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.51.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.52.n	Integer32	dbOptMaxCursorCount*	max_cursor_count
1.3.6.1.4.1.897.2.2.3.53.n	Integer32	dbOptMaxHashSize*	max_hash_size
1.3.6.1.4.1.897.2.2.3.54.n	Integer32	dbOptMaxPlansCached*	max_plans_cached
1.3.6.1.4.1.897.2.2.3.55.n	Integer32	dbOptMaxRecursivelterations*	max_recursive_iterations
1.3.6.1.4.1.897.2.2.3.56.n	Integer32	dbOptMaxStatementCount*	max_statement_count
1.3.6.1.4.1.897.2.2.3.57.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.58.n	Integer32	dbOptMinPasswordLength*	min_password_length
1.3.6.1.4.1.897.2.2.3.59.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.60.n	Integer32	dbOptNearestCentury*	nearest_century
1.3.6.1.4.1.897.2.2.3.61.n	String	dbOptNonKeywords*	non_keywords
1.3.6.1.4.1.897.2.2.3.62.n	String	dbOptOdbcDistinguishCharA ndVarchar*	odbc_distinguish_char_and_v archar
1.3.6.1.4.1.897.2.2.3.63.n	String	dbOptOnCharsetConversionF ailure*	on_charset_conversion_failur e
1.3.6.1.4.1.897.2.2.3.64.n	String	dbOptOnTsqlError*	on_tsql_error
1.3.6.1.4.1.897.2.2.3.65.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.66.n	String	dbOptOptimizationGoal*	optimization_goal
1.3.6.1.4.1.897.2.2.3.67.n	Integer32	dbOptOptimizationLevel*	optimization_level
1.3.6.1.4.1.897.2.2.3.68.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.69.n	String	dbOptOptimizationWorkload*	optimization_workload
1.3.6.1.4.1.897.2.2.3.70.n	Integer32	dbOptPinnedCursorPercentO fCache*	pinned_cursor_percent_of_ca che
1.3.6.1.4.1.897.2.2.3.71.n	Integer32	dbOptPrecision*	precision
1.3.6.1.4.1.897.2.2.3.72.n	String	dbOptPrefetch*	prefetch
1.3.6.1.4.1.897.2.2.3.73.n	String	dbOptPreserveSourceFormat *	preserve_source_format
1.3.6.1.4.1.897.2.2.3.74.n	String	dbOptPreventArticlePkeyUpd ate*	prevent_article_pkey_update
1.3.6.1.4.1.897.2.2.3.75.n	Integer32	dbOptInternal	Internal

OID	型	名前	オプション
1.3.6.1.4.1.897.2.2.3.76.n	String	dbOptQuotedIdentifier*	quoted_identifier
1.3.6.1.4.1.897.2.2.3.77.n	String	dbOptReadPastDeleted*	read_past_deleted
1.3.6.1.4.1.897.2.2.3.78.n	Integer32	dbOptRecoveryTime*	recovery_time
1.3.6.1.4.1.897.2.2.3.79.n	String	dbOptReplicateAll*	replicate_all
1.3.6.1.4.1.897.2.2.3.80.n	String	dbOptReturnDateTimeAsString*	return_date_time_as_string
1.3.6.1.4.1.897.2.2.3.81.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.82.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.83.n	String	dbOptRowCounts*	row_counts
1.3.6.1.4.1.897.2.2.3.84.n	Integer32	dbOptScale*	scale
1.3.6.1.4.1.897.2.2.3.85.n	String	dbOptSortCollation*	sort_collation
1.3.6.1.4.1.897.2.2.3.86.n	String	dbOptSqlFlaggerErrorLevel*	sql_flagger_error_level
1.3.6.1.4.1.897.2.2.3.87.n	String	dbOptSqlFlaggerWarningLevel*	sql_flagger_warning_level
1.3.6.1.4.1.897.2.2.3.88.n	String	dbOptStringRtruncation*	string_rtruncation
1.3.6.1.4.1.897.2.2.3.89.n	String	dbOptSubsumeRowLocks*	subsume_row_locks
1.3.6.1.4.1.897.2.2.3.90.n	String	dbOptSuppressTdsDebugging*	suppress_tds_debugging
1.3.6.1.4.1.897.2.2.3.91.n	String	dbOptTdsEmptyStringsIsNull*	tds_empty_string_is_null
1.3.6.1.4.1.897.2.2.3.92.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.93.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.94.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.95.n	String	dbOptTimestampFormat*	timestamp_format
1.3.6.1.4.1.897.2.2.3.96.n	String	dbOptTimeFormat*	time_format
1.3.6.1.4.1.897.2.2.3.97.n	Integer32	dbOptTimeZoneAdjustment*	time_zone_adjustment
1.3.6.1.4.1.897.2.2.3.98.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.99.n	String	dbOptTruncateTimestampValues*	truncate_timestamp_values
1.3.6.1.4.1.897.2.2.3.100.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.101.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.102.n	String	dbOptTsqlVariables*	tsql_variables
1.3.6.1.4.1.897.2.2.3.103.n	String	dbOptUpdateStatistics*	update_statistics
1.3.6.1.4.1.897.2.2.3.104.n	String	dbOptUpgradeDatabaseCapability*	upgrade_database_capability
1.3.6.1.4.1.897.2.2.3.105.n	String	dbOptUserEstimates*	user_estimates
1.3.6.1.4.1.897.2.2.3.106.n	String	dbOptWaitForCommit*	wait_for_commit

OID	型	名前	オプション
1.3.6.1.4.1.897.2.2.3.107.n	String	dbOptTempSpaceLimitCheck*	temp_space_limit_check
1.3.6.1.4.1.897.2.2.3.108.n	Integer32	dbOptRemotIdleTimeout*	remote_idle_timeout
1.3.6.1.4.1.897.2.2.3.109.n	String	dbOptAnsiSubstring*	ansi_substring
1.3.6.1.4.1.897.2.2.3.110.n	String	dbOptOdbcDescribeBinaryAsVarbinary*	odbc_describe_binary_as_varbinary
1.3.6.1.4.1.897.2.2.3.111.n	String	dbOptRollbackOnDeadlock*	rollback_on_deadlock
1.3.6.1.4.1.897.2.2.3.112.n	String	dbOptIntegratedServerName*	integrated_server_name
1.3.6.1.4.1.897.2.2.3.113.n	String	dbOptLogDeadlocks*	log_deadlocks
1.3.6.1.4.1.897.2.2.3.114.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.115.n	String	dbOptWebserviceNamespaceHost*	webservice_namespace_host
1.3.6.1.4.1.897.2.2.3.116.n	Integer32	dbOptMaxQueryTasks*	max_query_tasks
1.3.6.1.4.1.897.2.2.3.117.n	Integer32	dbOptRequestTimeout*	request_timeout
1.3.6.1.4.1.897.2.2.3.118.n	String	dbOptSynchronizeMirrorOnCommit*	synchronize_mirror_on_commit
1.3.6.1.4.1.897.2.2.3.119.n	Integer32	dbOptHttpSessionTimeout*	http_session_timeout
1.3.6.1.4.1.897.2.2.3.120.n	String	dbOptUuidHasHyphens*	uuid_has_hyphens
1.3.6.1.4.1.897.2.2.3.121.n	String	dbOptAllowSnapshotIsolation*	allow_snapshot_isolation
1.3.6.1.4.1.897.2.2.3.122.n	String	dbOptVerifyPasswordFunction*	verify_password_function
1.3.6.1.4.1.897.2.2.3.123.n	String	dbOptDefaultDbSpace*	default_db_space
1.3.6.1.4.1.897.2.2.3.124.n	String	dbOptCollectStatisticsOnDmlUpdates*	collect_statistics_on_dml_updates
1.3.6.1.4.1.897.2.2.3.125.n	String	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.126.n	String	dbOptJavaLocation*	java_location
1.3.6.1.4.1.897.2.2.3.127.n	String	dbOptOemString*	oem_string
1.3.6.1.4.1.897.2.2.3.128.n	Integer32	dbOptMaxTempSpace*	max_temp_space
1.3.6.1.4.1.897.2.2.3.129.n	String	dbOptSecureFeatureKey*	secure_feature_key
1.3.6.1.4.1.897.2.2.3.130.n	String	dbOptMaterializedViewOptimization*	materialized_view_optimization
1.3.6.1.4.1.897.2.2.3.131.n	Integer32	dbOptUpdatableStatementIsolation*	updatable_statement_isolation
1.3.6.1.4.1.897.2.2.3.132.n	String	dbOptTsqlOuterJoins*	tsql_outer_joins
1.3.6.1.4.1.897.2.2.3.133.n	String	dbOptPostLoginProcedure*	post_login_procedure
1.3.6.1.4.1.897.2.2.3.134.n	String	dbOptConnAuditing*	conn_auditing

OID	型	名前	オプション
1.3.6.1.4.1.897.2.2.3.135.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.136.n	String	dbOptJavaVmOptions*	java_vm_options
1.3.6.1.4.1.897.2.2.3.137.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.138.n	Integer32	dbOptMaxClientStatementsC ached*	max_client_statements_cach ed
1.3.6.1.4.1.897.2.2.3.139.n	String	dbOptQueryMemTimeout*	query_mem_timeout
1.3.6.1.4.1.897.2.2.3.140.n	String	dbOptAllowReadClientFile*	allow_read_client_file
1.3.6.1.4.1.897.2.2.3.141.n	String	dbOptAllowWriteClientFile*	allow_write_client_file
1.3.6.1.4.1.897.2.2.3.142.n	String	dbOptPriority*	priority
1.3.6.1.4.1.897.2.2.3.143.n	String	dbOptMaxPriority*	max_priority
1.3.6.1.4.1.897.2.2.3.144.n	String	dbOptProgressMessages*	progress_messages
1.3.6.1.4.1.897.2.2.3.145.n	Integer32	dbOptBlockingOthersTimeou t*	blocking_others_timeout
1.3.6.1.4.1.897.2.2.3.146.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.147.n	String	dbOptTimestampWithTimeZo neFormat*	timestamp_with_time_zone_f ormat
1.3.6.1.4.1.897.2.2.3.148.n	Integer32	dbOptHttpConnectionPoolTi meout*	http_connection_pool_timeou t
1.3.6.1.4.1.897.2.2.3.149.n	Integer32	dbOptHttpConnectionPoolBa sesize*	http_connection_pool_basesi ze
1.3.6.1.4.1.897.2.2.3.150.n	String	dbOptStGeometryDescribeTy pe*	st_geometry_describe_type
1.3.6.1.4.1.897.2.2.3.151.n	String	dbOptStGeometryAstextFor mat*	st_geometry_astext_format
1.3.6.1.4.1.897.2.2.3.152.n	String	dbOptStGeometryAsbinaryFo rmat*	st_geometry_asbinary_forma t
1.3.6.1.4.1.897.2.2.3.153.n	String	dbOptStGeometryAsxmlForm at*	st_geometry_asxml_format
1.3.6.1.4.1.897.2.2.3.154.n	String	dbOptReservedKeywords*	reserved_keywords
1.3.6.1.4.1.897.2.2.3.155.n	String	dbOptStGeometryOnInvalid*	st_geometry_on_invalid
1.3.6.1.4.1.897.2.2.3.156.n	Integer32	dbOptMinRoleAdmins*	min_role_admins
1.3.6.1.4.1.897.2.2.3.157.n	String	dbOptTrustedCertificatesFile *	trusted_certificates_file
1.3.6.1.4.1.897.2.2.3.158.n	String	dbOptJavaClassPath*	java_class_path
1.3.6.1.4.1.897.2.2.3.159.n	String	dbOptStGeometryInterpolatio n*	st_geometry_interpolation
1.3.6.1.4.1.897.2.2.3.160.n	String	dbOptExternLoginCredentials *	extern_login_credentials

OID	型	名前	オプション
1.3.6.1.4.1.897.2.2.3.161.n	String	dbOptWebserviceSessionidName*	webservice_sessionid_name
1.3.6.1.4.1.897.2.2.3.162.n	String	dbOptDbPublisher*	db_publisher
1.3.6.1.4.1.897.2.2.3.163.n	String	dbOptAutoCommitOnCreateLocalTempIndex*	auto_commit_on_create_local_temp_index
1.3.6.1.4.1.897.2.2.3.164.n	String	dbOptDiskSandbox*	disk_sandbox
1.3.6.1.4.1.897.2.2.3.165.n	String	dbOptBaseTablesInRlvStore*	base_tables_in_rlv_store
1.3.6.1.4.1.897.2.2.3.166.n	String	dbOptAuditLog*	audit_log
1.3.6.1.4.1.897.2.2.3.167.n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.168.n	Integer32	dbOptMaxConnections*	max_connections
1.3.6.1.4.1.897.2.2.3.169.n	String	dbOptTimeZone*	time_zone
1.3.6.1.4.1.897.2.2.3.170.n	String	dbOptConnectionType*	connection_type
1.3.6.1.4.1.897.2.2.3.171.n	String	dbOptParameterizationLevel*	parameterization_level
1.3.6.1.4.1.897.2.2.3.172.n	String	dbOptAutoCommit*	auto_commit

## 関連情報

[アルファベット順のデータベースオプションリスト \[632 ページ\]](#)

### 1.8.2.3.9 RDBMS MIB リファレンス

SQL Anywhere SNMP Extension Agent を使用して、多くの値の OID を検索することができます。

デフォルトでは、RDBMS MIB は `C:\Program Files\SQL Anywhere 17\snmp\RDBMS-MIB.mib` にあります。

このセクションの内容:

[rdbmsDbTable \[1344 ページ\]](#)

システムにインストールされているデータベースについての情報を示します。

[rdbmsDbInfoTable \[1344 ページ\]](#)

システム上のデータベースについての追加情報を示します。

[rdbmsDbParamTable \[1345 ページ\]](#)

システム上のデータベースの設定パラメータを示します。

[rdbmsDbLimitedResourceTable \[1346 ページ\]](#)

各 DB 領域に関する空き領域情報のリストです。

[rdbmsSrvTable \[1346 ページ\]](#)

システム上の動作中またはインストール済みのデータベースサーバを示します。

[rdbmsSrvInfoTable \[1347 ページ\]](#)

システム内のデータベースサーバについての追加情報を示します。

[rdbmsSrvParamTable \[1347 ページ\]](#)

SQL Anywhere SNMP Extension Agent で SQL Anywhere MIB を使用して設定できるサーバオプションを示します。

[rdbmsSrvLimitedResourceTable \[1348 ページ\]](#)

サーバ設定パラメータについての情報が格納されています。

## 1.8.2.3.9.1 rdbmsDbTable

システムにインストールされているデータベースについての情報を示します。

値 `db` は、`sasnmplib.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
<a href="#">1.3.6.1.2.1.39.1.1.1.1.db</a>	Integer	rdbmsDbIndex	<code>db</code>
<a href="#">1.3.6.1.2.1.39.1.1.1.2.db</a>	OID	rdbmsDbPrivateMibOID	1.3.6.1.4.1.897.2
<a href="#">1.3.6.1.2.1.39.1.1.1.3.db</a>	String	rdbmsDbVendorName	PROPERTY('CompanyName')
<a href="#">1.3.6.1.2.1.39.1.1.1.4.db</a>	String	rdbmsDbName	DB_PROPERTY('Name')
<a href="#">1.3.6.1.2.1.39.1.1.1.5.db</a>	String	rdbmsDbContact	PROPERTY('LicensedUser')

## 1.8.2.3.9.2 rdbmsDbInfoTable

システム上のデータベースについての追加情報を示します。

値 `db` は `sasnmplib.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
<a href="#">1.3.6.1.2.1.39.1.2.1.1.db</a>	String	rdbmsDbInfoProductName	PROPERTY('ProductName')
<a href="#">1.3.6.1.2.1.39.1.2.1.2.db</a>	String	rdbmsDbInfoVersion	PROPERTY('ProductVersion')

OID	型	名前	戻り値
1.3.6.1.2.1.39.1.2.1.3.db	Integer	rdbmsDbInfoSizeUnits	dbInfoSizeAllocatedと dbInfoSizeUsedに基づいて計 算される <ul style="list-style-type: none"> <li>• 1=バイト</li> <li>• 2=KB</li> <li>• 3=MB</li> <li>• 4=GB</li> <li>• 5=TB</li> </ul> (各単位はその1つ前の単位の 1024倍)
1.3.6.1.2.1.39.1.2.1.4.db	Integer	rdbmsDbInfoSizeAllocated	DB_PROPERTY('PageSize') * DB_PROPERTY('FileSize')
1.3.6.1.2.1.39.1.2.1.5.db	Integer	rdbmsDbInfoSizeUsed	DB_PROPERTY('PageSize') * (DB_PROPERTY('FileSize') - DB_PROPERTY('FreePages') )
1.3.6.1.2.1.39.1.2.1.6.db	String	rdbmsDbInfoLastBackup	NULL <sup>1</sup>

<sup>1</sup>この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

### 1.8.2.3.9.3 rdbmsDbParamTable

システム上のデータベースの設定パラメータを示します。

値 db は sasnmplib.ini ファイル内のデータベース番号です。また、n は sa.2.3 サブツリー内のオプションのインデックスです。

OID	型	名前	戻り値
1.3.6.1.2.1.39.1.3.1.1.db	String	rdbmsDbParamName	オプション名
1.3.6.1.2.1.39.1.3.1.2.db	Integer	rdbmsDbParamSubIndex	n
1.3.6.1.2.1.39.1.3.1.3.db	OID	rdbmsDbParamID	このオプションに対応する SQL Anywhere MIB 内の OID
1.3.6.1.2.1.39.1.3.1.4.db	String	rdbmsDbParamCurrValue	オプション値
1.3.6.1.2.1.39.1.3.1.5.db	String	rdbmsDbParamComment	NULL <sup>1</sup>

<sup>1</sup>この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

## 1.8.2.3.9.4 rdbmsDbLimitedResourceTable

各 DB 領域に関する空き領域情報のリストです。

表中の *n* は、以下に示すとおり、いずれかの DB 領域を表す値になります。

- 1 ~ 13: 通常の DB 領域 (データベース内では番号 0 ~ 12)
- 14: トランザクションログファイル
- 15: トランザクションログミラーファイル
- 16: テンポラリファイル

値 *db* は、`sasnmplib.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.2.1.39.1.4.1.1.n.db	String	rdbmsDbLimitedResourceName	DB 領域の名前、トランザクションログ、トランザクションログミラー、またはテンポラリファイル
1.3.6.1.2.1.39.1.4.1.2.n.db	OID	rdbmsDbLimitedResourceID	1.3.6.1.4.1.897.2
1.3.6.1.2.1.39.1.4.1.3.n.db	Integer	rdbmsDbLimitedResourceLimit	ディスク上の使用可能な空き領域 + 現在のファイルサイズ
1.3.6.1.2.1.39.1.4.1.4.n.db	Integer	rdbmsDbLimitedResourceCurrent	現在のファイルサイズ
1.3.6.1.2.1.39.1.4.1.5.n.db	Integer	rdbmsDbLimitedResourceHighwater	現在のサイズ
1.3.6.1.2.1.39.1.4.1.6.n.db	Integer	rdbmsDbLimitedResourceFailure	0 <sup>1</sup>
1.3.6.1.2.1.39.1.4.1.7.n.db	String	rdbmsDbLimitedResourceDescription	バイト、KB、MB、GB、TB のいずれか

<sup>1</sup>この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

## 1.8.2.3.9.5 rdbmsSrvTable

システム上の動作中またはインストール済みのデータベースサーバを示します。

値 *db* は `sasnmplib.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.2.1.39.1.5.1.1.db	OID	rdbmsSrvPrivateMibOID	1.3.6.1.4.1.897.2
1.3.6.1.2.1.39.1.5.1.2.db	String	rdbmsSrvVendorName	PROPERTY( 'CompanyName' )
1.3.6.1.2.1.39.1.5.1.3.db	String	rdbmsSrvProductName	PROPERTY( 'ProductName' )
1.3.6.1.2.1.39.1.5.1.4.db	String	rdbmsSrvContact	PROPERTY( 'LicensedCompany' )

## 1.8.2.3.9.6 rdbmsSrvInfoTable

システム内のデータベースサーバについての追加情報を示します。

値 `db` は、`sasnmp.ini` ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.2.1.39.1.6.1.1.db	Integer	rdbmsSrvInfoStartupTime	PROPERTY( 'StartTime' )
1.3.6.1.2.1.39.1.6.1.2.db	Integer	rdbmsSrvInfoFinishedTransactions	0 <sup>1</sup>
1.3.6.1.2.1.39.1.6.1.3.db	Integer	rdbmsSrvInfoDiskReads	PROPERTY( 'DiskReadEng' )
1.3.6.1.2.1.39.1.6.1.4.db	Integer	rdbmsSrvInfoLogicalReads	0 <sup>1</sup>
1.3.6.1.2.1.39.1.6.1.5.db	Integer	rdbmsSrvInfoDiskWrites	PROPERTY( 'DiskWriteEng' )
1.3.6.1.2.1.39.1.6.1.6.db	Integer	rdbmsSrvInfoLogicalWrites	0 <sup>1</sup>
1.3.6.1.2.1.39.1.6.1.7.db	Integer	rdbmsSrvInfoPageReads	0 <sup>1</sup>
1.3.6.1.2.1.39.1.6.1.8.db	Integer	rdbmsSrvInfoPageDiskOutOfWrites	0 <sup>1</sup>
1.3.6.1.2.1.39.1.6.1.9.db	Integer	rdbmsSrvInfoSpaces	0 <sup>1</sup>
1.3.6.1.2.1.39.1.6.1.10.db	Integer	rdbmsSrvInfoHandledRequests	PROPERTY( 'Req' )
1.3.6.1.2.1.39.1.6.1.11.db	Integer	rdbmsSrvInfoRequestRecvs	PROPERTY( 'PacketsReceivedUncomp' )
1.3.6.1.2.1.39.1.6.1.12.db	Integer	rdbmsSrvInfoRequestSends	PROPERTY( 'PacketsSentUncomp' )
1.3.6.1.2.1.39.1.6.1.13.db	Integer	rdbmsSrvInfoHighwaterInboundAssociations	0 <sup>1</sup>
1.3.6.1.2.1.39.1.6.1.14.db	Integer	rdbmsSrvInfoMaxInboundAssociations	0 <sup>1</sup>

<sup>1</sup>この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

## 1.8.2.3.9.7 rdbmsSrvParamTable

SQL Anywhere SNMP Extension Agent で SQL Anywhere MIB を使用して設定できるサーバオプションを示します。

`n` はインデックスです。このインデックスの詳細は次のとおりです。

n	サーバオプション
1	ConnsDisabled
2	LivenessTimeout (デフォルト)
3	QuittingTime

n	サーバオプション
4	RememberLastStatement
5	RequestLogFile
6	RequestLogging

値 db は、sasmp.ini ファイル内のデータベース番号です。

OID	型	名前	戻り値
1.3.6.1.2.1.39.1.7.1.1.n.db	String	rdbmsDbSrvParamName	オプション n の名前
1.3.6.1.2.1.39.1.7.1.2.n.db	Integer	rdbmsDbSrvParamSubIndex	n
1.3.6.1.2.1.39.1.7.1.3.n.db	OID	rdbmsDbSrvParamID	1.3.6.1.4.1.897.2
1.3.6.1.2.1.39.1.7.1.4.n.db	String	rdbmsDbSrvParamCurrValue	オプション n の現在の値
1.3.6.1.2.1.39.1.7.1.5.n.db	String	rdbmsDbSrvParamComment	オプション n の完全名

## 1.8.2.3.9.8 rdbmsSrvLimitedResourceTable

サーバ設定パラメータについての情報が格納されています。

値 db は、sasmp.ini ファイル内のデータベース番号です。また、n はリソースのインデックスです。このインデックスの詳細は次のとおりです。

n	名前	リソース	リソースの制限
1	接続	PROPERTY('UniqueClientAddresses')	PROPERTY('LicenseCount')
2	プロセッサ	PROPERTY('NumLogicalProcessorsUsed')	PROPERTY('NumLogicalProcessorsUsed')

OID	型	名前	戻り値
1.3.6.1.2.1.39.1.8.1.1.db	String	rdbmsSrvLimitedResourceName	リソース n の名前
1.3.6.1.2.1.39.1.8.1.2.db	OID	rdbmsSrvLimitedResourceID	このオプションに対応する SQL Anywhere MIB 内の OID
1.3.6.1.2.1.39.1.8.1.3.db	Integer	rdbmsSrvLimitedResourceLimit	リソース n の上限
1.3.6.1.2.1.39.1.8.1.4.db	Integer	rdbmsSrvLimitedResourceCurrent	リソース n の現在の値
1.3.6.1.2.1.39.1.8.1.5.db	Integer	rdbmsSrvLimitedResourceHighwater	リソース n の現在の値
1.3.6.1.2.1.39.1.8.1.6.db	Integer	rdbmsSrvLimitedResourceFailures	0 <sup>1</sup>

OID	型	名前	戻り値
1.3.6.1.2.1.39.1.8.1.7.db	String	rdbmsSrvLimitedResourceDescription	リソース $n$ の名前

<sup>1</sup>この OID は SQL Anywhere SNMP Extension Agent ではサポートされていません。

## 1.8.3 パフォーマンス

データベースおよびデータベースサーバのパフォーマンスを改善するために、たくさんのツールを使用することができます。

このセクションの内容:

### [SQL Anywhere コックピット \[1349 ページ\]](#)

コックピットはデータベースサーバの可用性、容量、およびパフォーマンスについて最新のビューを提供するデータベースモニタツールです。ブラウザでコックピットにアクセスします。コックピットでは、SAP DB Control Center の情報も提供されます。

### [パフォーマンス向上のヒント \[1366 ページ\]](#)

ソフトウェアのパフォーマンスを向上させるヒントをいくつかご紹介します。

### [パフォーマンス改善のためのインデックスの最適化 \[1407 ページ\]](#)

インデックス分析ツールであるインデックスコンサルタントを実行して、データベースに最適なインデックスセットについて推奨を取得します。

### 1.8.3.1 SQL Anywhere コックピット

コックピットはデータベースサーバの可用性、容量、およびパフォーマンスについて最新のビューを提供するデータベースモニタツールです。ブラウザでコックピットにアクセスします。コックピットでは、SAP DB Control Center の情報も提供されます。

コックピットはシステムに何らかの問題がある可能性を示している条件を検知すると、警告を発行します。高い CPU 使用率、多数のデータベースサーバ接続、長時間の接続や操作のブロックといった条件については、あらかじめ定義された警告があります。コックピットから、警告のスレッシュホルド値や電子メール通知設定を制御します。警告を生成する条件が true であるとき、警告は存在します。条件が存在しなくなると、警告は消えます。

コックピットを使用して、データベースサーバに接続しているユーザを表示し、データベースサーバとデータベースプロパティの両方を表示します。バックアップの開始や接続の切断といった簡単な管理タスクを実行することができます。

コックピットはモニタリングしている同一のデータベースサーバ上で実行します。バックグラウンドで連続的に実行するよう設計されているため、必要な時に接続することができます。しかしながら、問題を調査するために簡単に開始および停止することもできます。

データベースサーバの起動時にコックピットを起動して、必要なときにいつでもデータベースサーバの現在の状態を表示できるようにします。

データベースサーバ上で実行中のデータベースのユーザのクレデンシャルを使用して、コックピットに接続します。このユーザには COCKPIT\_ROLE ユーザ定義ロールが必要です。新しく作成されたデータベースでは、データベース管理者には管理者権限のみ付与されます。そのため、コックピットへの接続が必要なすべてのユーザに、このユーザ定義ロールを付与する必要

があります。コックピットを使用してタスクを実行するには、関連付けられたロールまたは権限が必要です。たとえば、データベースをバックアップするには、そのデータベースに対する BACKUP DATABASE システム権限が必要です。

コックピット でサポートされているブラウザに関する情報は、[プラットフォーム別 SQL Anywhere コンポーネント](#) を参照してください。

コックピットの起動時に、設定およびセッション間のアラート履歴を保持するかどうかを決定する必要があります。

コックピットアプリケーションでは、コックピットデータベースと呼ばれる、組み込みデータベースが使用されます。データベースには、アラートスレッシュホールドやアラート履歴などの設定が保存されます。コックピットデータベースは、ファイル名やパスを指定する恒久的なデータベースか、またはテンポラリデータベースにすることができます。一時的なコックピットデータベースは、コックピットの停止時に削除されます。コックピットデータベースが恒久的なデータベースである場合、コックピットで行った変更はすべて自動的にファイルに保存されます。

コックピット起動時に、恒久的データベースかテンポラリデータベースのどちらを使用するか選択します。テンポラリデータベースを選択する場合、[モニタリングページ](#)を開き、[警告の設定](#)をクリックし、次に[恒久的なコックピットデータベースに切り替え](#)をクリックすることで、コックピットで恒久的なデータベースを使用するように切り替えることができます。

CockpitDB データベースサーバプロパティでは、データベースサーバで現在使用されている一連のコックピットオプションとテンポラリパラメータを一覧表示します。Temp が "はい" に設定されている場合、コックピットデータベースはテンポラリデータベースです。

[ホームページ](#)を開き、[詳細の表示](#)をクリックして、プロパティテーブルのプロパティを検索することで、CockpitDB データベースのサーバプロパティを表示することができます。

このセクションの内容:

#### [コックピットの起動とコックピットへの接続 \(SQL Central\) \[1351 ページ\]](#)

実行中のデータベースサーバ上でコックピットを起動し、これを使用してデータベースサーバの現在の状態を表示します。

#### [コックピットの起動とコックピットへの接続 \(sa\\_server\\_option システムプロシージャ\) \[1353 ページ\]](#)

実行中のデータベースサーバ上でコックピットを起動し、これを使用してデータベースサーバの現在の状態を表示します。

#### [コックピットの停止および再開 \(sa\\_server\\_option system procedure\) \[1355 ページ\]](#)

データベースサーバ上で実行中のコックピットを停止および再開します。

#### [コックピットの停止と再開 \(SQL Central\) \[1356 ページ\]](#)

データベースサーバ上で実行中のコックピットを停止および再開します。

#### [SAP DB Control Center にコックピットを設定 \[1357 ページ\]](#)

SAP DB Control Center にコックピットを登録し、データベースサーバの正常性を他のエンタープライズシステムと併せて一元的にモニタリングします。

#### [コックピットによる警告電子メールの送信の有効化 \[1358 ページ\]](#)

コックピットを設定して、警告が発生した場合にユーザに電子メール通知を送信します。

#### [コックピットのセキュリティ \[1359 ページ\]](#)

コックピットを保護するために、たくさんのセキュリティ機能があります。

#### [コックピット チュートリアル \[1361 ページ\]](#)

コックピットを安全に起動し、データベースサーバの可用性、容量、パフォーマンスをモニタリングする方法を学習します。

## 関連情報

[-cdb データベースサーバオプション \[389 ページ\]](#)

[COCKPIT\\_ROLE ユーザ定義ロール \[1471 ページ\]](#)

### 1.8.3.1.1 コックピットの起動とコックピットへの接続 (SQL Central)

実行中のデータベースサーバ上でコックピットを起動し、これを使用してデータベースサーバの現在の状態を表示します。

#### 前提条件

コックピットを起動、停止するには、sa\_server\_option システムプロシージャの EXECUTE システム権限と、SERVER OPERATOR システム権限が必要です。

コックピットはモニタリングしているデータベースサーバ上で実行します。データベースサーバ上で実行中のデータベースのクレデンシャル (データベースユーザ ID およびパスワード) を使用して、コックピットに接続します。

データベースサーバでは次のことが必要です。

- 共有メモリ接続の受け入れ。たとえば、データベースサーバを起動するコマンドに `-ec TLS` データベースサーバオプションが含まれる場合、`-es` データベースサーバオプションを含みます。
- バージョン 16 以降のデータベースが少なくとも 1 つは実行中であること。
- HTTPS 接続リスナがあり、コックピットと Web ブラウザ間の通信が暗号化されていること。保護されていない接続リスナを使用すると、第三者にコックピットと Web ブラウザ間の通信を閲覧され、コックピットへの接続に使用するデータベースユーザ名やパスワードといった機密情報が漏えいしてしまう可能性があります。HTTPS 接続リスナには、作成した証明書を使用して署名するか、または証明機関によって署名された証明書が必要です。一般的に、証明機関は証明書への署名を有料で行っています。ルート証明書を作成した後、それを使用して署名付き証明書を作成することで、証明書作成ユーティリティ (createcert) を使用して署名付き証明書を作成することができます。ブラウザの信頼できるルート証明書の一覧を更新し、作成したルート証明書ファイルを追加します。証明機関から署名付き証明書を購入した場合は、このステップは必要ありません。一般的な証明機関からの信頼できる証明書の一覧は、ブラウザによって更新されるためです。データベースの起動時に、`-xs` データベースサーバオプションを指定し、HTTPS を HTTPS 接続リスナの起動プロトコルとして指定して、HTTPS 接続リスナを起動します。次に例を示します。

```
dbsrv17 -xs "HTTPS (PORT=443; IDENTITY=c:¥my-signed-certificate-identity-  
file.id; IDENTITY_PASSWORD=my-signed-certificate-password) " "C:¥Users¥Public  
¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

デフォルトでは、HTTPS 接続リスナはデータベースサーバで実行中のすべてのデータベースと共有されます。HTTPS 接続リスナをコックピットだけに制限するには、DBN プロトコルオプションを使用し、データベース名として `SQLACockpit` を指定します。次に例を示します。

```
dbsrv17 -cdb "DBF=C:¥temp¥cockpitsettings.db" -xs  
"HTTPS (PORT=443; DBN=SQLACockpit; IDENTITY=c:¥my-signed-certificate-identity-  
file.id; IDENTITY_PASSWORD=my-signed-certificate-password) " "C:¥Users¥Public  
¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

## 警告

サンプル ID ファイルは、テスト作業と開発作業にのみ使用します。運用システムでは使用しないでください。代わりに、アプリケーションを配備する前に独自の証明書で置換してください。

データベースサーバで実行中のデータベースでは次のことが必要です。

- 標準的な (データベースユーザ ID およびパスワードによる) ログインの受け入れ。データベースが Windows 統合ログイン、Kerberos、または PAM UA などの認証メカニズムをサポートしている場合、標準データベースログインもサポートする必要があります。  
COCKPIT\_ROLE ユーザ定義ロールが定義されていること。バージョン 16 のデータベースには、デフォルトでは COCKPIT\_ROLE が存在しないため、作成する必要があります。

ユーザクレデンシャルには次のことが必要です。

- COCKPIT\_ROLE ユーザ定義ロールの行使権限が付与されていること。デフォルトでは、初期ユーザを含むいずれのユーザにも COCKPIT\_ROLE の行使権限は付与されていないので、付与する必要があります。
- コックピットの全機能にアクセスするため、次のシステム権限があること。
  - ACCESS DISK INFORMATION システム権限
  - BACKUP DATABASE システム権限
  - DROP CONNECTION システム権限
  - MONITOR システム権限
  - SERVER OPERATOR システム権限

## コンテキスト

sa\_server\_option システムプロシージャを使用して、実行中のデータベースサーバでコックピットを起動することもできます。

## 手順

- SQL Central でデータベースに接続します。
- フォルダウィンドウ枠でデータベースサーバを右クリックして、[SQL Anywhere コックピットを開く](#)をクリックします。
- ウィザードの指示に従います。

コックピットが起動し、ブラウザが開いてログインページが表示されます。

コックピットに割り当てられる URL は、次のクエリを実行して取得することもできます。`SELECT PROPERTY ( 'CockpitURL' );`

- データベースのユーザ ID、パスワード、およびデータベース名を使用して接続します。

## 次のステップ

バックグラウンドで連続的に実行するため、必要な時に接続することができます。

## 関連情報

[コックピットのセキュリティ \[1359 ページ\]](#)

[COCKPIT\\_ROLE ユーザ定義ロール \[1471 ページ\]](#)

### 1.8.3.1.2 コックピットの起動とコックピットへの接続 (sa\_server\_option システムプロシージャ)

実行中のデータベースサーバ上でコックピットを起動し、これを使用してデータベースサーバの現在の状態を表示します。

#### 前提条件

コックピットを起動、停止するには、sa\_server\_option システムプロシージャの EXECUTE システム権限と、SERVER OPERATOR システム権限が必要です。

コックピットはモニタリングしているデータベースサーバ上で実行します。データベースサーバ上で実行中のデータベースのクレデンシャル (データベースユーザ ID およびパスワード) を使用して、コックピットに接続します。

データベースサーバでは次のことが必要です。

- 共有メモリ接続の受け入れ。たとえば、データベースサーバを起動するコマンドに -ec TLS データベースサーバオプションが含まれる場合、-es データベースサーバオプションを含みます。
- バージョン 16 以降のデータベースが少なくとも 1 つは実行中であること。
- HTTPS 接続リスナがあり、コックピットと Web ブラウザ間の通信が暗号化されていること。保護されていない接続リスナを使用すると、第三者にコックピットと Web ブラウザ間の通信を閲覧され、コックピットへの接続に使用するデータベースユーザ名やパスワードといった機密情報が漏えいしてしまう可能性があります。

HTTPS 接続リスナには、作成した証明書を使用して署名するか、または証明機関によって署名された証明書が必要です。一般的に、証明機関は証明書への署名を有料で行っています。

ルート証明書を作成した後、それを使用して署名付き証明書を作成することで、証明書作成ユーティリティ (createcert) を使用して署名付き証明書を作成することができます。ブラウザの信頼できるルート証明書の一覧を更新し、作成したルート証明書ファイルを追加します。証明機関から署名付き証明書を購入した場合は、このステップは必要ありません。一般的な証明機関からの信頼できる証明書の一覧は、ブラウザによって更新されるためです。

データベースの起動時に、-xs データベースサーバオプションを指定し、HTTPS を HTTPS 接続リスナの起動プロトコルとして指定して、HTTPS 接続リスナを起動します。次に例を示します。

```
dbsrv17 -xs "HTTPS (PORT=443; IDENTITY=c:¥my-signed-certificate-identity-  
file.id; IDENTITY_PASSWORD=my-signed-certificate-password) " "C:¥Users¥Public  
¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

デフォルトでは、HTTPS 接続リスナはデータベースサーバで実行中のすべてのデータベースと共有されます。HTTPS 接続リスナをコックピットだけに制限するには、DBN プロトコルオプションを使用し、データベース名として *SQLACockpit* を指定します。次に例を示します。

```
dbsrv17 -cdb "DBF=C:¥temp¥cockpitsettings.db" -xs  
"HTTPS (PORT=443; DBN=SQLACockpit; IDENTITY=c:¥my-signed-certificate-identity-
```

```
file.id;IDENTITY_PASSWORD=my-signed-certificate-password) " "C:¥Users¥Public
¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

### 警告

サンプル ID ファイルは、テスト作業と開発作業にのみ使用します。運用システムでは使用しないでください。代わりに、アプリケーションを配備する前に独自の証明書で置換してください。

データベースサーバで実行中のデータベースでは次のことが必要です。

- 標準的な (データベースユーザ ID およびパスワードによる) ログインの受け入れ。データベースが Windows 統合ログイン、Kerberos、または PAM UA などの認証メカニズムをサポートしている場合、標準データベースログインもサポートする必要があります。  
COCKPIT\_ROLE ユーザ定義ロールが定義されていること。バージョン 16 のデータベースには、デフォルトでは COCKPIT\_ROLE が存在しないため、作成する必要があります。

ユーザクレデンシャルには次のことが必要です。

- COCKPIT\_ROLE ユーザ定義ロールの行使権限が付与されていること。デフォルトでは、初期ユーザを含むいずれのユーザにも COCKPIT\_ROLE の行使権限は付与されていないので、付与する必要があります。
- コックピットの全機能にアクセスするため、次のシステム権限があること。
  - ACCESS DISK INFORMATION システム権限
  - BACKUP DATABASE システム権限
  - DROP CONNECTION システム権限
  - MONITOR システム権限
  - SERVER OPERATOR システム権限

## 手順

1. データベースに接続したあと、次の sa\_server\_option システムプロシージャを実行してコックピットを起動します。

```
CALL sa_server_option('CockpitDB' 'START=ON;DBF=c:¥temp¥cockpitsettings.db');
```

2. 次のクエリを実行して URL を取得し、コックピットにアクセスします。

```
SELECT PROPERTY ( 'CockpitURL' );
```

3. ブラウザを開いてコックピットの URL を参照します。
4. データベースユーザ ID、パスワード、およびデータベース名を使用してコックピットに接続します。

## 次のステップ

バックグラウンドで連続的に実行するため、必要な時に接続することができます。

## 関連情報

[COCKPIT\\_ROLE ユーザ定義ロール \[1471 ページ\]](#)

[-cdb データベースサーバオプション \[389 ページ\]](#)

[ネットワークプロトコルオプション \[177 ページ\]](#)

[データベースサーバプロパティの一覧 \[859 ページ\]](#)

[コックピットのセキュリティ \[1359 ページ\]](#)

### 1.8.3.1.3 コックピットの停止および再開 (sa\_server\_option system procedure)

データベースサーバ上で実行中のコックピットを停止および再開します。

#### 前提条件

データベースサーバでは、少なくとも1つのバージョン 16 以降のデータベースが実行中である必要があります。このデータベースで、sa\_server\_option システムプロシージャの EXECUTE システム権限と、SERVER OPERATOR システム権限のあるユーザである必要があります。

#### 手順

1. コックピットが実行しているのと同じデータベースサーバ上のデータベースに接続します。
2. 次の sa\_server\_option システムプロシージャを実行してコックピットを停止します。

```
CALL sa_server_option('CockpitDB' 'START=OFF');
```

3. コックピットが停止中に、実行する必要があるタスクを完了させます。
4. 次の sa\_server\_option システムプロシージャを実行してコックピットを再開します。

```
CALL sa_server_option('CockpitDB' 'START=ON');
```

## 1.8.3.1.4 コックピットの停止と再開 (SQL Central)

データベースサーバ上で実行中のコックピットを停止および再開します。

### 前提条件

データベースサーバでは、バージョン 16 以降のデータベースが 1 つ以上実行中である必要があります。このデータベースで、sa\_server\_option システムプロシージャの EXECUTE システム権限と、SERVER OPERATOR システム権限のあるユーザである必要があります。

### コンテキスト

sa\_server\_option システムプロシージャを使用して、コックピットを停止および再開することができます。

### 手順

1. SQL Central では、コックピットが稼働しているのと同じデータベースサーバ上のデータベースに接続します。
2. 左ウィンドウ枠でデータベースサーバを右クリックして、**プロパティ**をクリックします。
3. **プロパティ**ウィンドウの**オプション**をクリックします。
4. **コックピットをすぐに停止**をクリックします。
5. コックピットが停止中に、実行する必要があるタスクを完了させます。
6. 停止を行ったのと同じデータベースサーバ**プロパティ**ウィンドウで、**コックピットをすぐに起動**をクリックして、コックピットを再開します。

### 関連情報

[-cdb データベースサーバオプション \[389 ページ\]](#)

## 1.8.3.1.5 SAP DB Control Center にコックピットを設定

SAP DB Control Center にコックピットを登録し、データベースサーバの正常性を他のエンタープライズシステムと併せて一元的にモニタリングします。

### 前提条件

コックピットは1つ以上の別のデータベースが実行中のデータベース上で実行している必要があります。データベースのユーザであり、かつ COCKPIT\_ROLE ユーザ定義ロールを持つことが必要です。

SAP DB Control Center が実行中である必要があります。コックピットシステムに SAP DB Control Center を追加できる SAP DB Control Center ユーザである必要があります。

### コンテキスト

SAP DB Control Center はコックピットに接続して、データベースサーバの正常性に関する情報の概要を検索し表示します。コックピットに接続するには、SAP DB Control Center でユーザ名とパスワードを指定する必要があります。コックピットにこれらのクレデンシャルが作成され保存されます。作成したユーザは、SAP SB Control Center からコックピットに接続して概要情報の検索のみすることができます。

### 手順

1. データベースサーバでコックピットを起動し、コックピットのみが使用する HTTPS 接続リスナを作成します。データベースを同じデータベースサーバで起動します。

次のコマンドで、サンプルデータベースとコックピットを起動します。DBN パラメータを -xs データベースサーバオプションと使用して、コックピットのみが使用する HTTPS 接続リスナを作成します。

```
dbsrv17 -n my_cockpitserver -gk all -xs  
"HTTPS (PORT=443;DBN=SQLACockpit;IDENTITY=C:¥Users¥Public¥Documents¥SQL Anywhere  
17¥Samples¥Certificates¥rsaserver.id;IDENTITY_PASSWORD=test)" -cdb "DBF=C:¥temp  
¥cockpitsettings.db" "C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

2. コックピットに接続します。
  - a. ブラウザを開いてコックピットの URL を参照します。

コックピットに割り当てられる URL は、次のクエリを実行して取得することができます。SELECT PROPERTY ( 'CockpitURL' );.
  - b. データベースユーザ ID、パスワード、およびデータベース名を使用してコックピットに接続します。
3. SAP DB Control Center のクレデンシャルを作成して、コックピットがモニタリングしているデータベースサーバの正常性について、情報の概要を検索および表示するために使用します。
  - a. ホームタブで、データベースサーバの詳細の表示をクリックし、SAP DB Control Center の設定をクリックします。

- b. [SAP DB Control Center のクレデンシャルを設定](#)をクリックして、コックピットに接続するための SAP DB Control Center のユーザ名とパスワードを作成します。
4. SAP DB Control Center にログインします。
5. SAP DB Control Center のマニュアルに従って、コックピットを登録します。SAP DB Control Center ユーザ用に作成したユーザ名とパスワードを使用して、コックピットを登録します。

## 関連情報

[ユーザおよびデータベースのセキュリティ \[1453 ページ\]](#)

[COCKPIT\\_ROLE ユーザ定義ロール \[1471 ページ\]](#)

[コックピットの起動とコックピットへの接続 \(SQL Central\) \[1351 ページ\]](#)

[-cdb データベースサーバオプション \[389 ページ\]](#)

### 1.8.3.1.6 コックピットによる警告電子メールの送信の有効化

コックピットを設定して、警告が発生した場合にユーザに電子メール通知を送信します。

## 前提条件

データベースサーバでデータベースが稼働している必要があり、COCKPIT\_ROLE ユーザ定義ロールと次のシステム権限を持つユーザである必要があります。

- ACCESS DISK INFORMATION システム権限
- BACKUP DATABASE システム権限
- DROP CONNECTION システム権限
- MONITOR システム権限
- SERVER OPERATOR システム権限

自分の電子メールサーバ設定を知っておく必要があります。

## 手順

1. **モニタリング**タブで、**警告の設定** > **電子メール通知** をクリックし、**警告の通知を電子メールで送信**を選択します。
2. 次の電子メールプロトコルのうち 1 つを選択し、必要な設定を指定します。

オプション	アクション
SMTP サーバ	電子メールサーバ、ポート、送信者名、送信者のアドレスを指定します。
MAPI サーバ	ユーザ ID とパスワードを指定します。

3. **テスト電子メールを送信**をクリックして、電子メール設定を確認します。
4. **アドレスタブ**を開き、**追加**をクリックして、電子メール通知を受信するための電子メールアドレスを指定します。
5. **操作タブ**を開きます。
  - a. 電子メール通知の送信規則を設定します。
  - b. **コックピットにこの外部 URL または IP アドレスを使用**フィールドで、コックピットが動作しているコンピュータの外部名または IP アドレスを指定します。この外部名または IP アドレスは、通知電子メールで提供されるリンクに使用されます。
6. **OK** をクリックします。

## 結果

特定の基準を満たすか、または特定の基準を超える警告が発生した場合、特定のアドレスに対して通知電子メールが送信されます。

### 1.8.3.1.7 コックピットのセキュリティ

コックピットを保護するために、たくさんのセキュリティ機能があります。

データベースサーバ上で実行中のデータベースのユーザのクレデンシャルを使用して、コックピットに接続します。最低でもこのデータベースはバージョン 16 以降であり、COCKPIT\_ROLE ユーザ定義ロールが定義されている必要があります。クレデンシャルには、COCKPIT\_ROLE ユーザ定義ロールの行使権限も必要です。

デフォルトでは、COCKPIT\_ROLE の行使権限は初期ユーザを含めどのユーザにも付与されません。そのため、ユーザに行使権限を付与する必要があります。バージョン 16 のデータベースで、COCKPIT\_ROLE を作成して、ユーザに付与します。

コックピットは従来の定義者セキュリティモデルを使用するデータベースをサポートしません。

#### ユーザのコックピットへの接続を防止

データベースユーザがコックピットへ接続するのを防ぐには、COCKPIT\_ROLE ユーザ定義ロールの行使権限をユーザ付与しないでください。すべてのデータベースユーザがコックピットに接続できないようにするには、COCKPIT\_ROLE ユーザ定義ロールを削除してください。

## コックピットでのユーザのアクションを制限

データベース上で認証済みの任意のタスクを実行するには、タスクを実行するのに必要なロールと権限を使用して、そのデータベースのユーザとして接続します。提供されたクレデンシャルによる通常のデータベース接続で、SQL ステートメントを使用して実行したり表示したりすることができなかったコックピットを使用しても、コックピットに接続しているユーザはなにも実行したり表示したりすることはできません。

新規に作成されたデータベースでは、COCKPIT\_ROLE ユーザ定義ロールにはコックピットで任意のタスクを実行するのに必要なシステム権限が格納されています。この権限リストを見直して必要に応じて権限を取り消し、セキュリティモデルを維持してください。たとえば、John と Jane が両者とも MySample データベースのユーザであり、両者ともコックピットに接続する必要があります。データベースのセキュリティモデルによると、Jane のみがデータベースのバックアップを許可されています。この場合、BACKUP DATABASE システム権限を COCKPIT\_ROLE から削除してから、COCKPIT\_ROLE ロールを John と Jane 付与します。その後、Jane に BACKUP DATABASE システム権限を付与します。

## ユーザによる コックピット の起動を防止

データベースサーバの起動時に、データベースを起動できる任意のユーザがコックピットを起動することができます。

-sf データベースサーバオプションを使用して、ユーザが動作中のデータベースサーバのコックピットを起動できないようにします。

## コックピットとブラウザ間の通信の暗号化

HTTPS 接続リスナを起動し、コックピットと Web ブラウザ間の通信を暗号化します。HTTPS 接続リスナを起動しない場合、コックピットは保護されていない HTTP 接続リスナを起動します。保護されていない HTTP 接続リスナを使用すると、第三者にコックピットと Web ブラウザ間の通信を閲覧され、コックピットへの接続に使用するデータベースユーザ名やパスワードといった機密情報が漏えいしてしまう可能性があります。-xs データベースサーバオプションを指定して、HTTPS 接続リスナを作成します。

### 例

次のコマンドで、サンプルデータベースとコックピットを起動します。DBN パラメータを使用して、コックピットのみが使用する HTTPS 接続リスナを作成します。

```
dbsrv17 -cdb "DBF=C:¥temp¥cockpitsettings.db" -xs
"HTTPS(PORT=443;DBN=SQLACockpit;IDENTITY=c:¥my-signed-certificate-identity-
file.id;IDENTITY_PASSWORD=my-signed-certificate-password)" "C:¥Users¥Public
¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

## 関連情報

[COCKPIT\\_ROLE ユーザ定義ロール \[1471 ページ\]](#)

[ユーザおよびデータベースのセキュリティ \[1453 ページ\]](#)

[-xs データベースサーバオプション \[507 ページ\]](#)

## 1.8.3.1.8 コックピット チュートリアル

コックピットを安全に起動し、データベースサーバの可用性、容量、パフォーマンスをモニタリングする方法を学習します。

### 前提条件

このチュートリアルでは、サンプルデータベースを使用します。サンプルデータベース (*demo.db*) に COCKPIT\_ROLE ユーザ定義ロールがあり、このユーザ定義ロールの行使権限だけでなく次のシステム権限の行使権限もユーザに付与されていることを確認します。

- MONITOR システム権限
- DROP CONNECTION システム権限
- BACKUP DATABASE システム権限
- SERVER OPERATOR システム権限

sa\_server\_option システムプロシージャの EXECUTE システム権限と、このチュートリアルのほかのステップを完了させるための SERVER OPERATOR システム権限が必要です。

### コンテキスト

このチュートリアルでは、コックピットと同じデータベースサーバで稼働しているサンプルデータベースのユーザのクレデンシャルを指定して、コックピットに接続します。

サンプルデータベースでは、データベース管理者ユーザには、COCKPIT\_ROLE ユーザ定義ロールに付与された管理者権限と行使権限の両方があります。しかし、新しく作成されたデータベースでは、データベース管理者には管理者権限のみ付与されます。そのため、コックピットへの接続が必要なすべてのユーザに、COCKPIT\_ROLE ユーザ定義ロールを付与する必要があります。

このセクションの内容:

[レッスン 1: HTTPS 接続リスナを使用した、コックピットの安全な起動 \[1362 ページ\]](#)

署名付き証明書のある HTTPS 接続リスナを使用して、コックピットとブラウザ間の通信を保護します。

[レッスン 2: コックピット \(SQL Central\) での警告の表示 \[1364 ページ\]](#)

SQL Anywhere コックピット ユーザインタフェースを使って、警告のテストをしてみます。

## 1.8.3.1.8.1 レッスン 1: HTTPS 接続リスナを使用した、コックピットの安全な起動

署名付き証明書のある HTTPS 接続リスナを使用して、コックピットとブラウザ間の通信を保護します。

### 前提条件

このチュートリアル冒頭に記載されているロールと権限を持っている必要があります。

コックピットを稼働させるコンピュータのホスト名 (コンピュータ名) を知っている必要があります。

### コンテキスト

HTTPS 接続リスナには、認証機関 (CA) から購入した署名付き証明書、または証明書作成ユーティリティ (createcert) を使用して作成した署名付き証明書のいずれかが必要です。

このチュートリアルでは、証明書作成ユーティリティ (createcert) を使用して署名付き証明書を作成する方法、およびコックピットの証明書の使用方法を説明します。署名付き証明書を作成した場合、ブラウザの信頼できる証明書の一覧を更新して、証明書の署名に使用されるルート証明書ファイルを追加する必要があります。証明機関から署名付き証明書を購入した場合は、このステップは必要ありません。一般的な署名機関からの信頼できる証明書の一覧は、ブラウザによって更新されるためです。

### 手順

1. 署名付き証明書を作成します。
  - a. 証明書作成ユーティリティ (createcert) を実行して、署名権限のあるルート証明書を作成します。

createcert を実行し、次のオプションを指定します。

```
Enter RSA key length (512-16384): 2048
Generating key pair...
Country Code: ca
State/Province: on
Locality: waterloo
Organization: mycompany
Organizational Unit: eng
Common Name:myrootcommonname
Enter file path of signer's certificate:
Certificate will be a self-signed root
Serial number [generate GUID]:
Generated serial number: a5cf457769724934aed4d584d0d153a9
Certificate valid for how many years (1-100): 1
Certificate Authority (Y/N) [N]: y
1. Digital Signature
2. Nonrepudiation
3. Key Encipherment
4. Data Encipherment
```

```

5. Key Agreement
6. Certificate Signing
7. CRL Signing
8. Encipher Only
9. Decipher Only
Key Usage [6,7]:
Enter file path to save certificate: myroot.crt
Enter file path to save private key: myroot.key
Enter password to protect private key: myrootpassword
Enter file path to save identity: myroot.id

```

- b. ルート証明書 myroot.crt を使用して、データベースサーバで HTTPS 接続リスナの保護に使用される署名付き証明書を作成します。

createcert を実行し、次のオプションを指定します。

```

Enter RSA key length (512-16384): 2048
Generating key pair...
Country Code: ca
State/Province: on
Locality: waterloo
Organization: mycompany
Organizational Unit: eng
Common Name: Specify the host name of the Cockpit
Enter file path of signer's certificate: myroot.crt
Enter file path of signer's private key: myroot.key
Enter password for signer's private key: myrootpassword
Serial number [generate GUID]:
Generated serial number: 78ea0f2e763247d49f3b5bc05a0d3dde
Certificate valid for how many years (1-100): 1
Certificate Authority (Y/N) [N]: n
1. Digital Signature
2. Nonrepudiation
3. Key Encipherment
4. Data Encipherment
5. Key Agreement
6. Certificate Signing
7. CRL Signing
8. Encipher Only
9. Decipher Only
Key Usage [1,3,4,5]:
Enter file path to save certificate: myservercert.crt
Enter file path to save private key: myservercert.key
Enter password to protect private key: myservercertpassword
Enter file path to save identity: myservercert.id

```

**共通名** パラメータの値は、コックピットを稼働させるホスト名 (コンピュータ名) と一致している必要があります。

2. ブラウザの信頼されたルート証明機関の一覧に、ルート証明書 myroot.crt を追加します。方法については、ブラウザのマニュアルを参照してください。コックピットのアクセスに使用するブラウザやコンピュータごとにこのステップを繰り返します。
3. 同じデータベースサーバ上にあるコックピット、サンプルデータベース、および demo.db を起動します。

```

dbsrv17 -cdb "DBF=C:¥temp¥cockpitsettings.db" -xs
"HTTPS (PORT=443; IDENTITY=myservercert.id; IDENTITY_PASSWORD=myservercertpassword) "
"C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Samples¥demo.db"

```

-cdb データベースサーバオプションでコックピットを起動し、HTTPS プロトコルを指定した -xs データベースサーバで HTTPS 接続リスナを起動します。

デフォルトでは、HTTPS 接続リスナはデータベースサーバで実行中のすべてのデータベースと共有されます。HTTPS 接続リスナをコックピットのみ制限するには、DBN プロトコルオプションを使用し、データベース名として *SQLACockpit* を指定します。

コックピットが起動します。

#### 4. コックピットを開きます。

- a. SQL Central で、サンプルデータベースに接続します。
- b. フォルダウィンドウ枠でデータベースサーバを右クリックして、*SQL Anywhere コックピットの開始* をクリックします。

ブラウザにコックピット接続ページが表示されます。

データベースサーバメッセージウィンドウに、コックピットにアクセスするための URL が表示されます。この URL は、次の CockpitURL データベースサーバプロパティのクエリによって取得することもできます。`SELECT PROPERTY ( 'CockpitURL' )`

#### 5. サンプルデータベースのユーザ ID およびパスワードを使用して、コックピットに接続します。

## 次のステップ

次のレッスンに進みます。

## 関連情報

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

[ネットワークプロトコルオプション \[177 ページ\]](#)

## 1.8.3.1.8.2 レッスン 2: コックピット (SQL Central) での警告の表示

SQL Anywhere コックピット ユーザインターフェースを使って、警告のテストをしてみます。

## 前提条件

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に記載されているロールと権限を持っている必要があります。

## 手順

1. データベースサーバプロパティを検索します。
  - a. コックピットで、[ホームページ](#)に移動します。
  - b. データベースサーバの[詳細の表示](#)をクリックして、データベースサーバの起動に使用されるコマンドラインを見つけます。
  - c. [プロパティページ](#)で、右上のフィルタボックスに `CommandLine` を入力します。

CommandLine プロパティとその値が一覧に表示されます。

- d. ウィンドウを閉じます。
2. 警告スレッシュホールドを設定します。
    - a. [モニタリングページ](#)を開きます。
    - b. [警告の設定](#)をクリックします。
    - c. X またはそれ以上のサーバ接続が存在する場合に低アラートを発行警告のスレッシュホールドを 1 に調整します。
    - d. [OK](#) をクリックします。

この警告の条件が true のため、警告がトリガされます。警告が[モニタリングページ](#)のテーブルに表示されます。警告が表示されるまでに最大で 20 秒かかります。

3. 警告の名前をクリックして詳細を表示し、[警告ウィンドウ](#)を閉じます。
4. コックピットを停止します。
  - a. SQL Central でデータベースサーバを右クリックし、[プロパティ](#)をクリックします。
  - b. [オプション](#)をクリックします。
  - c. [コックピット](#) をすぐに停止をクリックします。

## 結果

これでコックピットの使用に関するチュートリアルは終了です。

## 関連情報

[COCKPIT\\_ROLE ユーザ定義ロール \[1471 ページ\]](#)

[COCKPIT\\_ROLE ユーザ定義ロール \[1471 ページ\]](#)

## 1.8.3.2 パフォーマンス向上のヒント

ソフトウェアのパフォーマンスを向上させるヒントをいくつかご紹介します。

### サーバ関連のパフォーマンスに関するヒント

- ヒント: パフォーマンス向上のためのキャッシュの使用
- ヒント: 同時実行性の問題点の確認
- ヒント: オプティマイザの目標の選択
- ヒント: カラム統計の更新
- ヒント: インデックスの有効な使用
- ヒント: インデックスのパフォーマンスの向上
- ヒント: 混合負荷または OLAP 負荷の最適化
- ヒント: クエリ結果の効果的なソートの使用
- ヒント: 正しいカーソルタイプの指定
- ヒント: 明示的な選択性推定の提供を控える
- ヒント: マテリアライズドビューを使ったクエリのパフォーマンス改善
- ヒント: テーブル検証時の WITH EXPRESS CHECK オプションの使用
- ヒント: クエリ処理におけるワークテーブルの使用 (All-rows 最適化目標の使用)

### アプリケーション関連のパフォーマンスに関するヒント

- ヒント: 効率的な SQL クエリの構築
- ヒント: クエリのパフォーマンスのモニタ
- ヒント: コストの高いユーザ定義関数の削減
- ヒント: オートコミットモードをオフにする
- ヒント: インメモリモードの使用

### データベース関連のパフォーマンスに関するヒント

- ヒント: 常にトランザクションログを使用する
- ヒント: コミットの遅延の使用
- ヒント: 小さいテーブルに関する統計収集
- ヒント: 断片化の削減
- ヒント: テーブル構造の正規化
- ヒント: カスケードされた参照アクションを最小限に抑える
- ヒント: 制約の宣言

- ヒント: 異なるファイルの異なるデバイスへの配置
- ヒント: データベースの再構築
- ヒント: キーを使ったクエリのパフォーマンス改善
- ヒント: プライマリキー幅の縮小
- ヒント: テーブル幅の縮小
- ヒント: テーブル内のカラムの順序の確認
- ヒント: コストの高いトリガを置き換える
- ヒント: 適切なページサイズの使用
- ヒント: オートインクリメントを使用したプライマリキーの作成
- ヒント: 適切なデータ型の使用
- ヒント: バルクオペレーション方法の使用
- ヒント: リソースガバナを使用する

## 通信関連のパフォーマンスに関するヒント

- ヒント: クライアントとサーバとの間の要求数の削減
- ヒント: 圧縮の慎重な使用
- ヒント: パケットサイズを変更してパフォーマンスを向上させる

このセクションの内容:

### ヒント: パフォーマンス向上のためのキャッシュの使用 [1370 ページ]

キャッシュとは、メモリの領域で、データベースサーバがデータベースページを格納して繰り返し高速にアクセスするために使用します。

### ヒント: 同時実行性の問題点の確認 [1377 ページ]

ロックによって、他のトランザクションがデータベースに格納されている情報に同時にアクセスすることを防止し、情報の信頼性を維持します。

### ヒント: 速度が遅い文の原因の特定 [1377 ページ]

ステートメントパフォーマンスサマリ機能では、速度の遅い文のパフォーマンスについてトラブルシューティングを行うときに、文の実行回数を返します。SQL Anywhere **プロファイラ**でこの機能を使用して、ステートメントパフォーマンスサマリの結果を表示します。

### ヒント: オプティマイザの目標の選択 [1378 ページ]

クエリ処理の最適化の目的を、最初のローを迅速に返すことに設定するか、または完全な結果セットを返すコストを最小限に抑えることに設定するかを選択することができます。

### ヒント: カラム統計の更新 [1378 ページ]

オプティマイザのパフォーマンスを継続的に向上させるために、データベースサーバは SELECT 文、INSERT 文、UPDATE 文、DELETE 文の処理中に、自動的にカラム統計を更新します。

### ヒント: インデックスの有効な使用 [1380 ページ]

データベースサーバはインデックスを使用して探索条件を評価することができます。

### ヒント: インデックスのパフォーマンスの向上 [1381 ページ]

複合インデックスを再編成したり、ページサイズを増やしたりすることによって、インデックスのパフォーマンスを向上させることができます。

#### ヒント: 混合負荷または OLAP 負荷の最適化 [1381 ページ]

optimization\_workload オプションを使用して、最適化設定を指定します。

#### ヒント: クエリ結果の効果的なソートの使用 [1381 ページ]

クエリ結果の効果的なソートを使用して、データの不要なソート回数を減らします。

#### ヒント: 正しいカーソルタイプの指定 [1381 ページ]

正しいカーソルタイプを指定して、パフォーマンスを改善します。

#### ヒント: 明示的な選択性推定の提供を控える [1381 ページ]

ユーザ定義の推定はパフォーマンスを改善することがありますが、継続的に使用される文にユーザ定義の推定を明示的に指定しないでください。

#### ヒント: マテリアライズドビューを使ったクエリのパフォーマンス改善 [1382 ページ]

負荷の高い集約操作やジョイン操作などを含むクエリのように、頻繁に実行されるコストの高いクエリでは、マテリアライズドビューを使用することを検討してください。

#### ヒント: テーブル検証時の WITH EXPRESS CHECK オプションの使用 [1385 ページ]

VALIDATE TABLE 文で WITH EXPRESS CHECK オプションを指定するか、検証ユーティリティ (dbvalid) で -fx オプションを使用すると、テーブルの検証が大幅に速くなります。

#### ヒント: クエリ処理におけるワークテーブルの使用 (All-rows 最適化目標の使用) [1386 ページ]

ワークテーブルは、クエリの実行中に作成される一時的な結果セットを実体化したものです。

#### ヒント: 効率的な SQL クエリの構築 [1387 ページ]

クエリの処理パフォーマンスを向上させるために、より効率的なクエリを構築することを検討します。

#### ヒント: クエリのパフォーマンスのモニタ [1392 ページ]

クエリのパフォーマンスをテストするために複数のツールが用意されています。

#### ヒント: コストの高いユーザ定義関数の削減 [1393 ページ]

クエリ中で何度も実行されるコストの高いユーザ定義関数を削減すると、パフォーマンスが改善されます。

#### ヒント: オートコミットモードをオフにする [1393 ページ]

オートコミットモードで実行する代わりに、SQL 文をグループ分けして各グループが 1 つの論理タスクを実行するようにします。

#### ヒント: インメモリモードの使用 [1393 ページ]

最新のチェックポイント後にコミットされたトランザクションが失われても構わないアプリケーションには、インメモリモードを使用すると便利な場合があります。

#### ヒント: ラグタイムミラーオプション設定の確認 (ミラーリングされているシステムのみ) [1393 ページ]

ラグタイムミラーオプションは、ミラーリングされているシステムのパフォーマンスに影響を与えます。

#### ヒント: 常にトランザクションログを使用する [1394 ページ]

トランザクションログを使用すると、データを保護できます。また、データベースサーバのパフォーマンスを向上できます。

#### ヒント: コミットの遅延の使用 [1394 ページ]

遅延コミットによって、データベースの COMMIT 文に対する応答頻度を決定します。

#### ヒント: 小さいテーブルに関する統計収集 [1394 ページ]

デフォルトでは、データベースサーバは 5 つ以上のローがあるテーブルに関する統計を作成します。5 つ未満のローを持つテーブルの統計を作成するには、CREATE STATISTICS 文を使用します。

#### ヒント: 断片化の削減 [1395 ページ]

データベースを変更する際に断片化が発生しますが、ファイル、テーブル、インデックスが必要以上に断片化されていると、パフォーマンスが低下することがあります。

#### ヒント: テーブル構造の正規化 [1400 ページ]

データベーステーブルに同じ情報の複数のコピーが含まれる場合があります (たとえば、複数のテーブルで繰り返されているカラム)、テーブルの正規化が必要な場合があります。

#### ヒント: カスケードされた参照アクションを最小限に抑える [1400 ページ]

カスケードされた参照アクションは、コストが高くなります。これは、トランザクションごとに複数のテーブルが更新されるためです。このことがパフォーマンスに影響する可能性があります。

#### ヒント: 制約の宣言 [1400 ページ]

プライマリキーと外部キーの関係およびその制約を宣言することで、パフォーマンスを改善することができます。

#### ヒント: 異なるファイルの異なるデバイスへの配置 [1400 ページ]

異なるデータベースファイルを異なる物理デバイスまたはドライブに入れると、データベースのパフォーマンスが向上する場合があります。

#### ヒント: データベースの再構築 [1401 ページ]

データベースを再構築すると、パフォーマンスが向上する場合があります。

#### ヒント: キーを使ったクエリのパフォーマンス改善 [1401 ページ]

プライマリキーと外部キーを使用して、データベースのパフォーマンスを向上させることができます。

#### ヒント: プライマリキー幅の縮小 [1402 ページ]

プライマリキーに含まれるカラム数を減らすと、パフォーマンスが改善されます。

#### ヒント: テーブル幅の縮小 [1402 ページ]

ワイドテーブルが存在し、パフォーマンスが低下している場合は、テーブルをさらに正規化してカラム数を減らすことを検討してください。

#### ヒント: テーブル内のカラムの順序の確認 [1402 ページ]

テーブル内のカラムの順序はパフォーマンスに影響します。

#### ヒント: コストの高いトリガを置き換える [1403 ページ]

使用しているトリガを評価して、データベースサーバで利用できる機能で置き換えられるトリガはないかを確認します。

#### ヒント: 適切なページサイズの使用 [1403 ページ]

データベースページサイズは、データベースのパフォーマンスに影響を与える場合があります。

#### ヒント: オートインクリメントを使用したプライマリキーの作成 [1404 ページ]

オートインクリメント機能を使用したプライマリキーの作成は、値がデータベースサーバによって生成されるため、他のどの方法よりも高速です。

#### ヒント: 適切なデータ型の使用 [1404 ページ]

データに適切なデータ型を使用することで、パフォーマンスを改善できます。

#### ヒント: バルクオペレーション方法の使用 [1405 ページ]

非常に大量の情報をデータベースにロードする場合、各タスクに対する専用ツールを使用すると便利です。

#### ヒント: リソースガバナを使用する [1405 ページ]

リソースガバナを使用して、パフォーマンスを改善することができます。

#### ヒント: クライアントとサーバとの間の要求数の削減 [1405 ページ]

LazyClose と PrefetchOnOpen ネットワーク接続パラメータを使用して、クライアントとサーバ間の要求数を削減します。

#### ヒント: 圧縮の慎重な使用 [1406 ページ]

1 つまたはすべての接続について圧縮機能を有効にして、パケット圧縮時の最小サイズ制限を調整すると、パフォーマンスを大幅に向上できる場合があります。

ヒント: パケットサイズを変更してパフォーマンスを向上させる [1406 ページ]

場合によっては、パケットサイズを大きくすることによってパフォーマンスが改善することがあります。

### 1.8.3.2.1 ヒント: パフォーマンス向上のためのキャッシュの使用

キャッシュとは、メモリの領域で、データベースサーバがデータベースページを格納して繰り返し高速にアクセスするために使用します。

キャッシュでアクセスできるページが増えると、データベースサーバによるディスクからのデータの読み込み (遅い操作) の回数が減ります。そのため、キャッシュサイズがパフォーマンスにおける重要な要因になることがよくあります。

システムを全体としてモニタリングすることにより、キャッシュサイズが適切かつ自動的に調整されるよう、動的キャッシュサイズ決定がサポートされています。データベースの起動時にデータベースサーバのコマンドラインで `-c` オプションを使用して、データベースキャッシュのサイズを制御することもできます。

データベースサーバメッセージウィンドウには起動時のキャッシュサイズが表示されますが、次の文を使用して現在のキャッシュサイズを取得することもできます。

```
SELECT PROPERTY( 'CurrentCacheSize' );
```

暗号化されたデータベースには、I/O 操作を最小限に抑えるのに十分なキャッシュが必要です。暗号化や複合化は操作ごとに実行する必要があるため、暗号化されたデータベース上では、暗号化されていないデータベース上よりも I/O 操作の負荷が高くなるためです。

このセクションの内容:

#### キャッシュとメモリガバナ [1371 ページ]

データベースサーバは、キャッシュ (バッファプール) を使用し、データベースページのイメージをメモリに一時的に格納します。

#### キャッシュと最適化 [1372 ページ]

データベースサーバは、追加バッファキャッシュオーバーヘッドを必要とする実行プランに高いコストがかかるようにしています。

#### キャッシュメモリ使用の制限 [1372 ページ]

キャッシュの初期サイズ、最小サイズ、最大サイズは、データベースサーバのコマンドラインから制御できます。

#### 動的キャッシュサイズ決定 [1373 ページ]

動的キャッシュサイズ決定では、データベースサーバで使用可能なメモリがある限りキャッシュサイズは大きくなり、他のアプリケーションがキャッシュメモリを必要とするときはキャッシュサイズは小さくなります。

#### キャッシュサイズをモニタリングする統計 [1375 ページ]

Windows パフォーマンスモニタとデータベースのプロパティ関数には、複数の統計があります。

#### キャッシュウォーミング [1375 ページ]

キャッシュウォーミングによって、データベースに対して実行されるクエリの実行時間を低減します。

### 1.8.3.2.1.1 キャッシュとメモリガバナ

データベースサーバは、キャッシュ (バッファプール) を使用し、データベースページのイメージをメモリに一時的に格納します。

これらのページは通常、テーブルページとインデックスページです。ただし、データベースには他の種類の物理ページも格納されます。これらのページに加えて、データベースサーバは他に 2 つのメモリプールのキャッシュを使用します。これらのプールの 1 つは仮想メモリです。接続、文、カーソルを表す構造など、データベースサーバのデータ構造に使用されます。2 番目のプールは、クエリメモリの仮想ストレージとして使用されるキャッシュページで構成されます。

クエリの実行が効率的に動作するには、メモリが必要です。データベースサーバはメモリガバナを使用して、各文がクエリの実行に使用できるクエリメモリの量を決定します。メモリガバナはクエリメモリプールを文に割り付け、負荷を効率的に実行できるようにします。

メモリガバナは選択されたページ数を各文に付与し、文はメモリを大量に消費するクエリ処理にそのページを使用できます。クエリプロセッサがページを使用するまでは、引き続きクエリメモリプール内のメモリを他の目的 (テーブルのバッファリングやインデックスページなど) にも使用できます。クエリメモリを使用し、メモリを大量に消費するクエリ処理には、ハッシュ DISTINCT、ハッシュ GROUP BY、ハッシュジョインなどのあらゆるハッシュベースの演算子や、ソート演算子、ウィンドウ演算子があります。

次の設定、演算子、統計を使用して、メモリガバナによるキャッシュの使用方法を理解 (および制御) します。

#### QueryMemMaxUseful グラフィカルプラン演算子

文が実行を開始すると、メモリガバナはオプティマイザの推定を使用して、文が使用できるメモリ容量を決定します。この推定は、グラフィカルなプランに QueryMemMaxUseful として表示されます。

#### QueryMemActiveMax サーバプロパティ

メモリガバナは、メモリを大量に消費する要求の同時実行数を制限します。この最大値は、データベースサーバを実行するコンピュータのパフォーマンス特性に応じて選択されます。また、この制限は、サーバプロパティ QueryMemActiveMax に表示されます。

#### QueryMemActiveEst パフォーマンスモニタの統計

メモリガバナは、メモリを大量に消費する要求の同時実行数の推定を継続します。この推定はデータベースサーバプロパティおよびパフォーマンスモニタの統計 QueryMemActiveEst に使用できます。

#### query\_mem\_timeout データベースオプション

メモリを大量に消費する文の実行を開始したとき、メモリを大量に消費する要求の最大同時実行数にすでに達している場合は、いずれかの実行中の要求が割り付けられたメモリを解放するまで入力された文は待機します。

query\_mem\_timeout データベースオプションは、入力された要求にメモリが付与されるまで待つ時間を制御します。デフォルト設定 -1 を使用すると、要求はデータベースサーバで定義された期間待機します。待機後、使用できるメモリが付与されない場合は文のアクセスプランは少ないメモリ容量で実行されます。これによって実行が遅くなる場合がありますが、プラン内のメモリを大量に消費する物理演算子に対して実行時のメモリ量が少なくなる方式を使用できる場合は、その方式が使用されます。

#### QueryMemGrantWaiting サーバプロパティおよびパフォーマンスモニタの統計

データベースサーバプロパティおよびパフォーマンスモニタの統計 QueryMemGrantWaiting には、メモリ要求が付与されるのを待機している現在の要求数が表示されます。

#### QueryMemGrantWaitед サーバプロパティおよびパフォーマンスモニタの統計

データベースサーバプロパティおよびパフォーマンスモニタの統計 QueryMemGrantWaitед には、メモリ要求が付与されるまでに要求が待機する必要があった合計回数が表示されます。

#### QueryMemNeedsGrant グラフィカルプラン演算子

グラフィカルなプランでは、QueryMemNeedsGrant には、要求が単純な要求 (メモリの付与が不要) かメモリを大量に消費する (メモリの付与が必要) かをメモリガバナが検討した結果が表示されます。メモリガバナによってメモリの付与が不要と分類された要求は、ただちに実行されます。それ以外の場合、要求はクエリメモリプールの一部の使用を求めます。

#### QueryMemLikelyGrant グラフィカルプラン演算子

グラフィカルなプランでは、QueryMemLikelyGrant には、要求を実行するために付与されることが予想される推定ページ数が表示されます。

### 1.8.3.2.1.2 キャッシュと最適化

データベースサーバは、追加バッファキャッシュオーバーヘッドを必要とする実行プランに高いコストがかかるようにしています。

たとえば、カーソルの中身を保持するために追加メモリを予約すると、コストがかかります。キャッシュが一杯であれば、1 ページまたは複数のページをディスクに書き込み、新しいページの領域を確保しなければなりません。後続の操作を完了するために、数ページの再読み込みが必要になることもあります。このコストにより、最適化はワークテーブルを使用するプランを選択しないようになっています。ただし、パフォーマンスが向上する場合には、最適化は注意しながらメモリを使用します。たとえば、サブクエリの結果がクエリ処理中に繰り返し必要になる場合、これらの結果をキャッシュします。

### 1.8.3.2.1.3 キャッシュメモリ使用の制限

キャッシュの初期サイズ、最小サイズ、最大サイズは、データベースサーバのコマンドラインから制御できます。

#### 初期キャッシュサイズ

データベースサーバの初期キャッシュサイズを指定するには、データベースサーバの `-c` オプションを使用します。`-c` オプションを指定しない場合、データベースサーバによって初期キャッシュメモリの割り付けサイズが計算されます。

#### 最大キャッシュサイズ

最大キャッシュサイズを制御するには、データベースサーバの `-ch` オプションを指定します。デフォルトは、使用しているコンピュータの物理メモリによって異なるヒュースティックに基づいています。UNIX 以外のコンピュータでは、デフォルトの最大キャッシュサイズは、最大キャッシュサイズとコンピュータの物理メモリの 90% のうち、いずれか低い方になります。UNIX では、デフォルトの最大キャッシュサイズは次のように計算されます。

- 32 ビットの UNIX プラットフォームでは、物理メモリ量の合計の 90% または 1,834,880 KB のいずれか小さい方です。
- 64 ビットの UNIX プラットフォームでは、物理メモリ量の合計の 90% または 8,589,672,320 KB のいずれか小さい方です。

#### 最小キャッシュサイズ

最小キャッシュサイズを制御するには、データベースサーバの `-cl` サーバオプションを指定します。デフォルトでは、最小キャッシュサイズは初期キャッシュサイズと同じです。

`-cl` を指定しないで `-c` サーバオプションを指定した場合、最小キャッシュサイズは `-c` サーバオプションで指定された初期キャッシュサイズに設定されます。

`-c` または `-cl` サーバオプションを設定しない場合、必要に応じてキャッシュを縮小できるように、最小キャッシュサイズは非常に低いハードコード定数値に設定されます。

初期キャッシュサイズまたは最小キャッシュサイズを最大キャッシュサイズの 8 分の 1 より小さい値に設定すると、最大キャッシュサイズに応じて初期および最小のキャッシュサイズが自動的に増大します。

また、動的キャッシュサイズ決定を無効にするには、`-ca 0` サーバオプションを使用します。

次のデータベースサーバプロパティは、データベースサーバのキャッシュに関する情報を返します。

#### **CurrentCacheSize**

現在のキャッシュサイズ (キロバイト単位) を返す。

#### **MinCacheSize**

許容最小キャッシュサイズ (キロバイト単位) を返す。

#### **MaxCacheSize**

許容最大キャッシュサイズ (キロバイト単位) を返す。

#### **PeakCacheSize**

現在のセッションでキャッシュが到達した最大値 (キロバイト単位) を返します。

### 1.8.3.2.1.4 動的キャッシュサイズ決定

動的キャッシュサイズ決定では、データベースサーバで使用可能なメモリがある限りキャッシュサイズは大きくなり、他のアプリケーションがキャッシュメモリを必要とするときはキャッシュサイズは小さくなります。

動的キャッシュサイズ決定の効果は、オペレーティングシステムや、利用可能な物理メモリの量によって異なります。

キャッシュの必要量は、動的キャッシュサイズ決定によって通常 1 分ごとに評価されます。ただし、新しいデータベースが起動されたときやファイル量が大幅に増加した場合、30 秒間は 5 秒ごとに評価されるよう実行間隔が狭まる場合があります。最初の 30 秒間が経過すると、評価の実行頻度は 1 分間隔に戻ります。ファイル増加量が前回に評価の実行間隔を狭めたときの増加量よりも 8 分の 1 多い場合、またはファイル量がデータベース起動時より 8 分の 1 増加している場合に、これは大幅な増加であると判断されます。

動的キャッシュサイズ決定を使用する場合は、明示的にデータベースキャッシュを設定する必要はありません。

初期キャッシュサイズまたは最小キャッシュサイズを最大キャッシュサイズの 8 分の 1 より小さい値に設定しようとすると、最大キャッシュサイズに応じて初期および最小のキャッシュサイズが自動的に増大します。

このセクションの内容:

#### [動的キャッシュサイズ決定 \(Windows\) \[1374 ページ\]](#)

Windows では、データベースサーバがキャッシュとオペレーションの統計を 1 分ごとに評価し、最適なキャッシュサイズを計算します。

#### [動的キャッシュサイズ決定 \(UNIX\) \[1374 ページ\]](#)

UNIX では、データベースサーバはスワップ領域とメモリを使用してキャッシュサイズを管理します。

### 1.8.3.2.1.4.1 動的キャッシュサイズ決定 (Windows)

Windows では、データベースサーバがキャッシュとオペレーションの統計を 1 分ごとに評価し、最適なキャッシュサイズを計算します。

データベースサーバは、ターゲットのキャッシュサイズを計算します。このキャッシュは、現在使用されていない物理メモリの約 5 MB をシステムが使用できるように残し、それ以外をすべて使用します。ターゲットキャッシュサイズが、明示的または暗黙的に指定された最小キャッシュサイズより小さくなることはありません。ターゲットのキャッシュサイズは、明示的または暗黙的に指定された最大キャッシュサイズ、またはすべてのオープンデータベースとテンポラリファイル合計サイズにメインヒープのサイズを加えたものを超えることはありません。

キャッシュサイズの変動を防ぐために、データベースサーバはキャッシュサイズを段階的に増やします。すぐにターゲット値に調整するのではなく、調整するたびに現在のサイズとターゲットサイズの差の 75% ずつキャッシュサイズを修正します。

### 1.8.3.2.1.4.2 動的キャッシュサイズ決定 (UNIX)

UNIX では、データベースサーバはスワップ領域とメモリを使用してキャッシュサイズを管理します。

スワップ領域はシステムワイドなリソースで、ほとんどの UNIX オペレーティングシステムにあります。メモリとスワップ領域の合計をシステムリソースと呼びます。詳細については、使用しているオペレーティングシステムのマニュアルを参照してください。

起動時に、データベースは指定した最大キャッシュサイズをシステムリソースから割り付けます。この一部をメモリ (初期キャッシュサイズ) にロードし、残りをスワップ領域として残します。

データベースサーバが使用するシステムリソースの総量は、データベースサーバが停止するまで一定です。ただし、メモリにロードされる比率は変わります。データベースサーバは、1 分ごとにキャッシュとオペレーションの統計を評価します。データベースサーバがビジーでメモリが必要になると、キャッシュページをスワップ領域からメモリに移します。システム内の他の処理がメモリを必要とした場合、データベースサーバがキャッシュページをメモリからスワップ領域に移すことがあります。

#### 初期キャッシュサイズ

デフォルトでは、初期キャッシュサイズは使用可能なシステムリソースに基づいたヒューリスティックを使用して割り当てられます。初期キャッシュサイズは、データベースサイズの総量の 1.1 倍より常に小さくなります。

初期キャッシュサイズが使用可能なシステムリソースの 4 分の 3 より大きい場合、データベースサーバはメモリの不足を通知するエラーで終了します。

初期キャッシュサイズは、`-c` オプションを使用して変更できます。

#### 最大キャッシュサイズ

最大キャッシュサイズは、コンピュータの使用可能なシステムリソースより小さくしてください。デフォルトでは、最大キャッシュサイズはコンピュータの使用可能なシステムリソースと物理メモリ量の合計に基づいたヒューリスティックを使用して割り当てら

れます。キャッシュサイズは、明示的または暗黙的に指定された最大キャッシュサイズ、またはすべてのオープンデータベースとテンポラリファイル合計サイズにメインヒープのサイズを加えたものを超えることはありません。

使用可能なシステムリソースより大きい最大キャッシュサイズを指定すると、データベースサーバはメモリの不足を通知するエラーで終了します。使用可能なメモリより大きい最大キャッシュサイズを指定すると、データベースサーバはパフォーマンス低下の警告を出しますが、終了はしません。

データベースサーバはすべての最大キャッシュサイズをシステムリソースから割り付け、終了まで解放しません。他のアプリケーション用に領域を残しながら、パフォーマンスも低下させないように最大キャッシュサイズを選択してください。デフォルトの最大キャッシュサイズを求める式は、このバランスを考慮に入れたヒューリスティックを使用しています。デフォルト値がシステムに適さない場合にのみ、値を調整してください。

-ch サーバオプションを使用して最大キャッシュサイズを設定し、自動キャッシュ増加機能を制限できます。

## 最小キャッシュサイズ

-c オプションを指定した場合、最小キャッシュサイズは初期キャッシュサイズと同じです。-c オプションを指定しない場合は、UNIX 上での最小キャッシュサイズは 8 MB です。

-cl サーバオプションを使用して、最小キャッシュサイズを調整できます。

初期キャッシュサイズまたは最小キャッシュサイズを最大キャッシュサイズの 8 分の 1 より小さい値に設定しようとする、最大キャッシュサイズに応じて初期および最小のキャッシュサイズが自動的に増大します。

### 1.8.3.2.15 キャッシュサイズをモニタリングする統計

Windows パフォーマンスモニタとデータベースのプロパティ関数には、複数の統計があります。

#### **CurrentCacheSize**

キロバイト単位の現在のキャッシュサイズ

#### **MinCacheSize**

キロバイト単位の最小許容キャッシュサイズ

#### **MaxCacheSize**

キロバイト単位の最大許容キャッシュサイズ

#### **PeakCacheSize**

キロバイト単位のピークキャッシュサイズ

### 1.8.3.2.16 キャッシュウォーミング

キャッシュウォーミングによって、データベースに対して実行されるクエリの実行時間を低減します。

キャッシュを準備するために、データベースサーバはデータベースにすでに記録された収集ページがあるかどうかをチェックします。ある場合、データベースサーバが対応するページをキャッシュにロードします。データベースはキャッシュがページをロ

ードしている間、引き続き要求を処理できますが、大量の I/O アクティビティがデータベースで検出された場合は準備が停止することがあります。この場合、キャッシュに再ロードされるページセットに含まれないページにアクセスするクエリのパフォーマンスの低下を防ぐために、キャッシュウォーミングは停止します。

## 再起動後のキャッシュウォーミング

これは、データベースが最後に起動したときに参照していたデータベースサーバのキャッシュとデータベースページに対するキャッシュを事前にロードすることで実行されます。キャッシュを準備することにより、データベースが起動するたびに同一または類似のクエリがデータベースに対して実行される場合にも、パフォーマンスを向上させることができます。ただし、データベースの起動時に実行された文が、前回キャッシュの収集をオンにしてデータベースを起動したときのものとは異なる場合は、キャッシュウォーミングによってパフォーマンスが向上することはありません。

キャッシュウォーミング設定をデータベースサーバのコマンドラインで制御できます。データベースが起動してキャッシュウォーミングがオンになると、データベースページの収集とキャッシュの再ロード（準備）という 2 種類のアクティビティが実行されます。

参照されたデータベースページの収集は、`-cc` データベースサーバオプションで制御され、デフォルトでオンになっています。データベースページの収集がオンになると、データベースサーバは、収集ページ数が最大数に到達するまで（値はキャッシュサイズとデータベースサイズに基づく）、収集速度が最小しきい値を下回るまで、またはデータベースが停止するまで、データベース起動時に要求されたすべてのデータベースページを追跡続けます。データベースサーバは、収集最大ページ数と収集しきい値を制御します。いったん収集が完了すると、参照されたページはデータベースに記録されるので、次のデータベース起動時にキャッシュの準備に使用できます。

キャッシュウォーミング（再ロード）はデフォルトでオンになっていて、`-cr` データベースサーバオプションで制御されます。キャッシュウォーミングに関する情報をデータベースサーバメッセージウィンドウに表示する場合は、`-cv` オプションを指定できます。

## 安定した状態へのキャッシュウォーミング

データベースが安定した状態で実行されているときにキャッシュのコンテンツを保存し、必要に応じてキャッシュをその状態にリストアすることにより、安定した状態にキャッシュを準備します。安定した状態にキャッシュウォーミングすることにより、次のような状況でパフォーマンスを改善することができます。つまり、安定した状態ではパフォーマンスが良好だったのに、キャッシュコンテンツが安定するまでの間、特定のクエリやオペレーションが遅くなるといった場合です。各クエリが後のクエリを考慮して最適化されていないために、後のクエリに必要なページを使用するという、よりコストの高いプランを選択していることが、この間にパフォーマンスが遅くなる原因の一つである場合があります。たとえば、データベース管理者は、データベースの再起動前にキャッシュのコンテンツを保存したり、およびデータベースの再起動後にキャッシュされたコンテンツをリストアしたりして、通常の再起動で発生する中断を最小化することができます。

安定した状態へのキャッシュウォーミングは、デフォルトではオフになっているため、`ALTER DATABASE` キャッシュウォーミング構文、または `sp_db_cache_contents` システムプロシージャおよび `sp_read_db_pages` システムプロシージャによって制御します。データベース管理者は、キャッシュの状態を明示的に記録し、格納されている状態で存在していたページのリストアが必要です。

### 1.8.3.2.2 ヒント: 同時実行性の問題点の確認

ロックによって、他のトランザクションがデータベースに格納されている情報に同時にアクセスすることを防止し、情報の信頼性を維持します。

データベースサーバがトランザクションを処理するとき、テーブルローを1つまたは複数ロックできます。また、更新中の情報を識別して、結果クエリの精度を高めます。

データベースサーバがトランザクションを処理するとき、テーブルローを1つまたは複数ロックできます。ロックは他のトランザクションが同時にアクセスすることを防止し、データベースに格納されている情報の信頼性を維持します。また、更新中の情報を識別して、結果クエリの精度を高めます。

データベースサーバは自動的にこれらのロックを設定するので、明示的な指示は必要ありません。データベースサーバは、トランザクションが獲得したすべてのロックを、そのトランザクションが完了するまで保持します。ローにアクセスしているトランザクションは、ロックを保持しているといえます。ロックの種類により、他のトランザクションのそのローへのアクセスは限定されるか、まったくできなくなります。

頻繁に1つまたは複数のローに複数のユーザが同時にアクセスすると、パフォーマンスは低下することがあります。ロックが問題になっていると考えられる場合は、sa\_locks プロシージャを使用して、データベースのロックに関する情報を入手できます。

ロックの問題が識別された場合は、関連する接続プロセスに関する情報を AppInfo 接続プロパティを使って表示できます。

### 1.8.3.2.3 ヒント: 速度が遅い文の原因の特定

ステートメントパフォーマンスサマリ機能では、速度の遅い文のパフォーマンスについてトラブルシューティングを行うときに、文の実行回数を返します。SQL Anywhere プロファイラでこの機能を使用して、ステートメントパフォーマンスサマリの結果を表示します。

ステートメントパフォーマンスサマリ機能では、sp\_top\_k\_statements および sp\_find\_top\_statements システムプロシージャを使用して、実行に最も時間がかかっている文とプランの組合せをレポートします。

ステートメントパフォーマンスサマリ機能を使用して、次のような質問に対して回答します。

- 今日、この文は以前より実行に時間がかかっているか。
- この文によって返される、または変更されるデータ量は、機能と同量か。
- 文の実行プランは変更されたか。
- ある実行プランは、今日は昨日より使用されたか。
- この文の最大ランタイムは平均ランタイムより長いか。
- あるプランの文の最大/平均ランタイムは、ほかの文の最大/平均ランタイムと大幅に異なるか。その場合、あるプランの呼び出しにより、ほかのプランの呼び出しより多くのローを処理または返しているか。

最大ランタイムが 0.005 秒以上の文の SQL は、GTSYSPERFCACHESTMT システムビューで確認できます。最大ランタイム 0.05 秒の文またはプランのグラフィカルプランは、すべて GTSYSPERFCACHEPLAN システムビューで確認できます。

単一のデータベースでステートメントパフォーマンスサマリ収集を無効にするには、sa\_db\_option システムプロシージャの CollectStmtPerfStats オプションを設定します。このオプションを設定するには、MONITOR 権限が必要です。

ステートメントパフォーマンスサマリ機能では、最も長い最大ランタイムを持つ最大 10,000 個の文に関する情報のみ保持します。

### 1.8.3.2.4 ヒント: オプティマイザの目標の選択

クエリ処理の最適化の目的を、最初のローを迅速に返すことに設定するか、または完全な結果セットを返すコストを最小限に抑えることに設定するかを選択することができます。

optimization\_goal オプションを使用すると、データベースサーバにおいて SQL 文を応答時間 (First-row) に対して最適化するか、リソースの総消費量 (All-rows) に対して最適化するかを制御できます。

このオプションを First-row に設定すると、データベースサーバは、クエリの結果の最初のローをフェッチするまでの時間を短縮するアクセスプランを選択します。この場合、検索にかかる合計時間は長くなることがあります。また、通常オプティマイザでは、可能であれば結果の実体化を必要とするアクセスプランは使用しないで、最初のローを返すまでの時間を短縮します。この設定では、たとえばオプティマイザは、明示的なソートの操作を必要とするアクセスプランではなく、クエリの ORDER BY 句を満たすインデックスを使用するアクセスプランを採用します。

オプティマイザが特定の文に使用する最適化目標は、次の規則に従って決まります。

- メインクエリブロックの FROM 句で、テーブルヒントが FASTFIRSTROW に設定されたテーブルが存在する場合、文は First-row 最適化目標を使用して最適化されます。
- 文の OPTION 句で optimization\_goal オプションが設定されている場合、文はその設定を使用して最適化されます。
- それ以外の場合、オプティマイザは現在の optimization\_goal オプションの設定を使用します。

最適化目標が First-row であっても、最初のローをすぐに返すことができるプランをオプティマイザが見つけれない可能性があります。たとえば、DISTINCT、GROUP BY、または ORDER BY 句が存在するために実体化が必要な文で、必要な順序を実現するためのインデックスが存在しない場合は、All-rows 目標で最適化されます。

このオプションを All-rows (デフォルト) に設定すると、クエリは最適化され、予測される合計検索時間が最短になるアクセスプランを選択します。PowerBuilder DataWindow アプリケーションなど、結果セット全体の処理が必要になるアプリケーションでは、optimization\_goal を All-rows に設定するのが適切です。

### 1.8.3.2.5 ヒント: カラム統計の更新

オプティマイザのパフォーマンスを継続的に向上させるために、データベースサーバは SELECT 文、INSERT 文、UPDATE 文、DELETE 文の処理中に、自動的にカラム統計を更新します。

カラムの統計情報は、データベースのシステムテーブル SYSCOLSTAT に永久的に格納されます。これは、テーブルやカラムを参照する述部に一致するローの数をモニタリングし、その数を推定されたローの数と比較し、必要に応じて既存の統計を更新することで行います。

利用可能なカラム統計の精度が高くなると、それに従ってオプティマイザがより適切に推定を計算できるため、以降のクエリのパフォーマンスが向上します。

データベースオプションを使用して、カラム統計を更新するかどうかを設定できます。update\_statistics データベースオプションは、クエリの実行中にカラム統計を更新するかどうかを指定します。collect\_statistics\_on\_dml\_updates データベースオプションは、LOAD、INSERT、DELETE、UPDATE などのデータを変更する DML 文の実行中に統計を更新するかどうかを指定します。

統計が現在のカラムの値を正確に反映していないためにパフォーマンスが悪いと考えられる場合は、CREATE STATISTICS 文や DROP STATISTICS 文を実行します。CREATE STATISTICS 文は古い統計を削除して新しい統計を作成し、DROP STATISTICS 文は古い統計を削除するだけです。

CREATE INDEX 文を実行すると、インデックスの統計が自動的に作成されます。

LOAD TABLE 文を実行すると、テーブルの統計が自動的に作成されます。

このセクションの内容:

[統計ガバナが統計を保守する方法 \[1379 ページ\]](#)

統計ガバナでは、データベース内の各統計の正常性と有用性を自動的に評価し、統計を自己モニタリングして自己回復できるように、必要な保守を実行します。

### 1.8.3.2.5.1 統計ガバナが統計を保守する方法

統計ガバナでは、データベース内の各統計の正常性と有用性を自動的に評価し、統計を自己モニタリングして自己回復できるように、必要な保守を実行します。

統計の保守はバックグラウンドで実行され、データベースサーバのパフォーマンスに対する大幅な負荷は発生しません。

統計ガバナによって、以下のタスクが実行されます。

- クエリのフィードバックから取得した推定エラーと統計の使用状況の記録
- 精度の低い統計の修正または再作成
- 効率的に保守を行えない統計の自動保守の停止
- 有用な可能性のある統計の作成
- 使用されていない統計の削除

update\_statistics オプションは、指定した接続が統計ガバナにクエリフィードバックを送信できるかどうかを制御します。このオプションが Off に設定されている場合、統計ガバナは指定した接続からクエリフィードバックを受け取りません。ただし、統計ガバナはその他の接続からクエリフィードバックを受け取り、統計の保守操作を実行できます。

統計ガバナは、統計の正常性と使用状況に基づいて、いつ統計を修正または作成するかを決定します。統計を修正または作成するには、クエリの実行中に統計情報を収集するか、または統計クリーナと呼ばれる独立したプロセスを実行します。統計ガバナを無効にすることなく、統計クリーナを無効にするには、sa\_server\_option システムプロシージャの StatisticsCleaner オプションを使用します。ただし、統計クリーナをオフにすると、クエリの実行時にのみ統計が作成または修正されます。

サーバの負荷を減らすために、統計ガバナは修正が困難な統計や使用されない統計の保守を停止します。短期間に何回も修正され、精度の低い推定を返す統計は削除され、30 日間保守されません。削除された統計は 30 日後に再作成され、定期的な保守が再開されます。sa\_server\_option システムプロシージャの DropBadStatistics オプションを使用すると、この機能を無効にできます。最近 90 日間に使用されなかった統計も削除されます。この機能を無効にするには、sa\_server\_option システムプロシージャの DropUnusedStatistics オプションを使用します。CREATE STATISTICS、DROP STATISTICS、または ALTER STATISTICS 文を使用すると、統計の保守をいつでも再開できます。

統計はメモリにロードされているテーブルに対してのみモニタされます。これらの統計は 30 分ごとにフラッシュされます。フラッシュする間に、統計の正常性と使用状況が検査され、統計ガバナが統計の保守を実行します。統計のステータス情報 (正常性、使用状況、統計の更新または削除に関する情報など) はセッション間で保持されません。データベースサーバが停止されると、ステータス情報はリセットされます。

## 1.8.3.2.6 ヒント: インデックスの有効な使用

データベースサーバはインデックスを使用して探索条件を評価することができます。

インデックスを使用することで、オプティマイザはデータへのアクセスをスピードアップし、ベーステーブルから読み込んで処理する情報量を減らします。たとえば、クエリに `WHERE column-name=value` という探索条件があり、カラムにインデックスが存在する場合、インデックススキャンを使って、探索条件を満たすテーブルのローだけを読み込むことができます。テーブルをジョインする場合にインデックスがあれば、パフォーマンスを大幅に向上させることもできます。

クエリを実行するとき、データベースサーバは各テーブルへのアクセス方法を選択します。データベースサーバが適切なインデックスを見つけられないときは、テーブルを順にスキャンしなければならず、時間を要します。

たとえば、大規模なデータベースから複数の従業員を検索する必要があるのに、姓か名前のどちらかしかわからないとします。インデックスがない場合、データベースサーバはテーブル全体をスキャンします。しかし、姓が先に入力されているインデックスと名が先に入力されているインデックスの 2 つのインデックスが作成されていれば、データベースサーバはこの 2 つのインデックスを先にスキャンするので、より速く情報を返すことができます。

### インデックスを適切に選択するとパフォーマンスに大きな違いが生じる

データベースサーバでは、インデックスによって非常に効率よく情報を検索できますが、インデックスを追加するときは注意してください。各インデックスは、ローの挿入、削除、更新時に余分な作業を生みます。データベースサーバは影響を受けるインデックスもすべて更新するからです。

このため、データベースサーバがデータにより効率的にアクセスできるときに限りインデックスを追加するようにしてください。特に、大きなテーブルに連続して不必要なアクセスを行うような処理をなくすときにインデックスを追加します。ただし、テーブルにローを追加するときのパフォーマンスを向上させる必要があり、情報を迅速に検索する必要がない場合は、できるだけ少ないインデックスを使用します。

インデックスコンサルタントを使用すると、データベースに有効なインデックスを選択する手助けとなります。

### クエリ最適化

オプティマイザは、可能であれば常に、インデックス専用取得でクエリを処理しようとします。インデックス専用取得では、データベースサーバでインデックスのデータだけを使用してクエリが処理されるため、テーブル内のローにアクセスする必要がありません。オプティマイザは、最高のパフォーマンスが得られるインデックスを判断し、それを使用するように自動的に選択します。ただし、クエリでインデックスヒントを使用して、オプティマイザが使用するインデックスを指定することもできます。指定したインデックスが 1 つでも使用できなかった場合はエラーが返されます。インデックスヒントによってパフォーマンスが低下する場合がありますため、経験のあるユーザだけが実行するようにしてください。インデックスコンサルタントを使用して、データベースにインデックスを追加する必要があるかどうかを判別できます。

### クラスタドインデックス

クラスタドインデックスを使用すると、テーブル内のローをインデックス内の順序とほぼ同じ順序で格納できます。

### 1.8.3.2.7 ヒント: インデックスのパフォーマンスの向上

複合インデックスを再編成したり、ページサイズを増やしたりすることによって、インデックスのパフォーマンスを向上させることができます。

これらの措置によって、インデックスの選択性とインデックスファンアウトが向上します。

### 1.8.3.2.8 ヒント: 混合負荷または OLAP 負荷の最適化

optimization\_workload オプションを使用して、最適化設定を指定します。

通常はクエリと同時に更新、削除、または挿入が実行される (混合負荷) データベースに対してクエリ処理を最適化するかどうか、またはデータベース内の更新アクティビティの主な形式が、クエリ実行と同時に実施されないバッチ形式で更新するかどうかを制御することができます。

### 1.8.3.2.9 ヒント: クエリ結果の効果的なソートの使用

クエリ結果の効果的なソートを使用して、データの不要なソート回数を減らします。

予測可能な順序で返されるデータが必要でなければ、SELECT 文で ORDER BY 句を指定しないようにします。ソートを行うには、クエリを処理するために余計な時間とリソースが必要です。

### 1.8.3.2.10 ヒント: 正しいカーソルタイプの指定

正しいカーソルタイプを指定して、パフォーマンスを改善します。

たとえば、カーソルが読み込み専用の場合、読み込み専用と宣言することで、最適化と実行が迅速になります。これは、検査制約など、構築する実体が少ないためです。カーソルが更新可能な場合は、クエリ書き換えの一部を省略できます。また、クエリが更新可能な場合、オプティマイザが選択した実行プランによっては、データベースサーバはキーセット駆動型アプローチを使用する必要があります。キーセットカーソルは、コストが高いことに留意してください。

### 1.8.3.2.11 ヒント: 明示的な選択性推定の提供を控える

ユーザ定義の推定はパフォーマンスを改善することがありますが、継続的に使用される文にユーザ定義の推定を明示的に指定しないでください。

データが変更されると、明示的な推定が不正確になり、オプティマイザが誤って不適切なプランを選択することがあります。

ソフトウェアによって選択されたアクセスプランが不適切であり、パフォーマンス問題を回避するために選択性推定を使用したものの、それが不正確であった場合は、user\_estimates を Off に設定して、値を無視できます。

場合によっては、統計が不正確になることがあります。このような状況が最も発生しやすいのは、大量のデータが追加、更新、または削除されてから実行されたクエリが少ない場合です。統計が不正確な場合、または利用できない場合、パフォーマンス

ンスに悪影響を与えます。データベースサーバが統計の更新に長時間要している場合、CREATE STATISTICS または DROP STATISTICS を実行して、統計を再表示してください。

データベースサーバでは、LOAD TABLE 文の実行時、クエリの実行中、また更新 DML 文の実行時にも一部の統計が更新されます。

ただし、特異な状況では、この方法では効果的でないことがあります。条件の成功する確率がオプティマイザの推定とは異なることがあらかじめ判明している場合、ユーザ推定を明示的に探索条件として指定できます。

## 1.8.3.2.12 ヒント: マテリアライズドビューを使ったクエリのパフォーマンス改善

負荷の高い集約操作やジョイン操作などを含むクエリのように、頻繁に実行されるコストの高いクエリでは、マテリアライズドビューを使用することを検討してください。

マテリアライズドビューには、集約されジョインされたデータを格納するクエリ可能な構造体があります。マテリアライズドビューは、データベースのサイズが大きく、頻繁にクエリが行われるために大量のデータで繰り返し集約やジョイン操作が発生するような環境において、パフォーマンスが向上するように設計されています。たとえば、マテリアライズドビューは、データウェアハウスアプリケーションでの使用に適しています。

最適化時に送信されるクエリを部分的または全体的に満たす候補として見なされるマテリアライズドビューについて、オプティマイザはそのリストを管理します。オプティマイザがクエリの全体または部分を満たす候補となるマテリアライズドビューを検出すると、最適化の列挙フェーズ (コストを基に最適なプランを判断する) で行われる推奨の対象にそのビューを含めます。マテリアライズドビューをクエリに一致させるためにオプティマイザが使用する処理をビューマッチングと呼びます。オプティマイザがマテリアライズドビューを検討するには、そのビューが一定の条件を満たす必要があります。マテリアライズドビューの使用が許可されていることをオプティマイザが判断すると、候補となるマテリアライズドビューのそれぞれが検査されます。マテリアライズドビューがクエリによって明示的に参照されないかぎり、オプティマイザによって使用される保証はありません。ただし、検討されるビューが条件を満たすようにすることはできます。

このセクションの内容:

### [マテリアライズドビューとビューマッチング \[1382 ページ\]](#)

オプティマイザは、ビューマッチングアルゴリズムを使用して、クエリを満たすためにマテリアライズドビューを使用できるかどうかを判断します。

### [マテリアライズドビューの候補リストの取得 \(SQL の場合\) \[1384 ページ\]](#)

オプティマイザによって検討されるマテリアライズドビューの候補リストを取得します。

### [オプティマイザによって検討されたマテリアライズドビューの判断 \[1385 ページ\]](#)

オプティマイザがクエリを満たすために使用するマテリアライズドビューを判断します。

## 1.8.3.2.12.1 マテリアライズドビューとビューマッチング

オプティマイザは、ビューマッチングアルゴリズムを使用して、クエリを満たすためにマテリアライズドビューを使用できるかどうかを判断します。

この判断には、クエリ検査ステップとマテリアライズドビュー検査ステップが含まれます。

## クエリ検査ステップ

クエリ検査では、ビューマッチングアルゴリズムによってクエリが検査されます。次のいずれかの条件に該当する場合、クエリの処理にマテリアライズドビューは使用されません。

- クエリで参照されるすべてのテーブルが更新可能である。  
派生の関係で更新可能である SELECT 文や、更新可能なカーソルで明示的に宣言された SELECT 文については、オプティマイザはマテリアライズドビューを検討しません。この状況は、Interactive SQL を使用していると発生することがあります。デフォルトで、Interactive SQL は SELECT 文で更新可能なカーソルを使用します。
- 文は、オプティマイザバイパスを使用する単純な DML 文であり、ヒューリスティックに最適化される。ただし、OPTION 句で FORCE OPTIMIZATION オプションを使用することにより、任意の SELECT 文でコストベースの最適化を使用できます。
- ストアドプロシージャやユーザ定義関数内に含まれるクエリの場合、クエリの実行プランはキャッシュされます。データベースサーバは、再利用できるようにこれらのクエリの実行プランをキャッシュする場合があります。このクラスのクエリの場合、クエリ実行プランは実行後にキャッシュされます。次回クエリが実行されると、プランが取得され、実行フェーズまでのすべてのフェーズがスキップされます。

## マテリアライズドビュー検査ステップ

オプティマイザは、ビュー定義が次の状態である場合に、ビューマッチングアルゴリズムで検査されるマテリアライズドビューのセットにそのマテリアライズドビューを含めます。

- 含まれるクエリブロックが 1 つだけです。
- 含まれる FROM 句が 1 つだけです。
- 次の構成体や仕様のいずれも含んでいません。
  - GROUPING SETS
  - CUBE
  - ROLLUP
  - サブクエリ
  - 派生テーブル
  - UNION
  - EXCEPT
  - INTERSECT
  - マテリアライズドビュー
  - DISTINCT
  - TOP
  - FIRST
  - セルフジョイン
  - 再帰ジョイン
  - FULL OUTER JOIN
- (オプション) GROUP BY 句や HAVING 句 (HAVING 句に subselect やサブクエリが含まれない場合) が含まれる。

ビュー定義の基準を満たす以外に、次の条件を満たす必要があります。

- データベースサーバでマテリアライズドビューを使用できます

- 最適化でマテリアライズドビューを使用できます
- マテリアライズドビューは初期化されています (データが入力されています)
- マテリアライズドビューを作成するための重要なオプションの値が、クエリを実行する接続のオプションと一致しています
- 最後に再表示したマテリアライズドビューで、materialized\_view\_optimization データベースオプションに設定された古さのしきい値を超えていません

マテリアライズドビューが上記の基準を満たし、クエリの全体または一部を満たすことがわかった場合、コストベースの最適プランが検出されると、ビューマッチングアルゴリズムは最適化の列挙フェーズ用の推奨にマテリアライズドビューを含めます。ただし、それにより最終的にそのマテリアライズドビューが最終実行プランで使用されるというわけではありません。たとえば、マテリアライズドビューを使用しない別のアクセスプランの方がコストが低いと推定されると、クエリの結果を計算するのに適切であると考えられるマテリアライズドビューであっても使用されない可能性があります。

## 1.8.3.2.12.2 マテリアライズドビューの候補リストの取得 (SQL の場合)

オプティマイザによって検討されるマテリアライズドビューの候補リストを取得します。

### 手順

1. 次の文を実行します。

```
SELECT * FROM sa_materialized_view_info( ) WHERE AvailForOptimization='Y';
```

オプティマイザはリストの生成時にオプション設定を考慮するため、返されるリストは要求している接続に固有です。接続に指定したオプションとマテリアライズドビューの作成時のオプションとが一致しない場合、そのマテリアライズドビューは候補と見なされません。

2. オプション設定の不一致が原因で候補と見なされないすべてのマテリアライズドビューのリストを取得するには、クエリを実行する接続から次のクエリを実行します。

```
SELECT * FROM sa_materialized_view_info( ) WHERE AvailForOptimization='O';
```

### 結果

マテリアライズドビューの候補リストが、表示されます。

### 1.8.3.2.12.3 オプティマイザによって検討されたマテリアライズドビューの判断

オプティマイザがクエリを満たすために使用するマテリアライズドビューを判断します。

#### コンテキスト

##### i 注記

スナップショットアイソレーションが使用されている場合、オプティマイザは現在のトランザクションのスナップショットの開始後に再表示されたマテリアライズドビューを検討しません。

#### 手順

1. Interactive SQL で、データベースに接続します。
2. **▶ ツール ▶ プランビューアを開く ▶**をクリックするか、Shift + F5 キーを押します。
3. SQL 文ウィンドウ枠にクエリを入力します。
4. 統計レベル、カーソルタイプ、更新のステータスを選択します。
5. プランの取得をクリックします。
6. 詳細と高度な詳細ウィンドウ枠を見て、クエリを満たすために使用したマテリアライズドビューを判断します。

#### 結果

クエリを満たすために使用したマテリアライズドビューが、表示されます。

### 1.8.3.2.13 ヒント: テーブル検証時の WITH EXPRESS CHECK オプションの使用

VALIDATE TABLE 文で WITH EXPRESS CHECK オプションを指定するか、検証ユーティリティ (dbvalid) で -fx オプションを使用すると、テーブルの検証が大幅に速くなります。

少ないキャッシュ容量で大きいデータベースを検証するのに時間がかかる場合は、このヒントを検討してください。

## 1.8.3.2.14 ヒント: クエリ処理におけるワークテーブルの使用 (All-rows 最適化目標の使用)

ワークテーブルは、クエリの実行中に作成される一時的な結果セットを実体化したものです。

ワークテーブルは、代替方式よりもコストが低い場合に使用します。通常、ワークテーブルを使用すると、最初のいくつかのローをフェッチする所要時間は長くなります。ただし、ワークテーブルを使用できれば、すべてのローを検索するコストは大幅に低下する場合があります。このような違いがあるため、データベースサーバは optimization\_goal の設定に基づいてさまざまな方式を選択します。デフォルトは All-rows です。All-rows に設定されている場合、クエリの合計実行コストが減少するならば、ワークテーブルが使用されます。optimization\_goal が First-row に設定されている場合、データベースサーバはワークテーブルを使用しないようにします。

ワークテーブルが使用されるのは、次の場合です。

- クエリに ORDER BY 句、GROUP BY 句、または DISTINCT 句があり、データベースサーバがローのソートにインデックスを使用しないとき。適切なインデックスが存在し、optimization\_goal が First-row に設定されている場合、データベースサーバはワークテーブルを使用しません。ただし、optimization\_goal が All-rows に設定されている場合は、ワークテーブルを作成してローをソートするよりも、インデックスを使用してクエリのローをすべてフェッチする方が高コストになることがあります。最適化目標が All-rows に設定されている場合、データベースサーバは負荷がより低い方式を選択します。GROUP BY と DISTINCT の場合、ハッシュベースのアルゴリズムではワークテーブルが使用されますが、通常はクエリからローをすべてフェッチする方が効率的です。
- ハッシュジョインアルゴリズムが選択されたとき。この場合、ワークテーブルが中間結果の格納 (入力がメモリに収まらない場合) とジョイン結果の格納に使用されます。
- sensitive 値を使用してカーソルが開かれたとき。この場合、ベーステーブルのロー識別子とプライマリキーを入れるワークテーブルが作成されます。このワークテーブルは、クエリから前方にローがフェッチされるたびに埋められます。ただし、カーソルから最後のローをフェッチすると、テーブル全体が埋められます。
- insensitive セマンティックを使用してカーソルが開かれたとき。この場合、クエリが開かれるときに、ワークテーブルにクエリの結果が設定されます。
- 複数のローに UPDATE を実行し、UPDATE の WHERE 句または更新に使用するインデックスに、更新されるカラムが表示されるとき。
- 複数のローに対する UPDATE または DELETE が、修正されるテーブルを参照する WHERE 句にサブクエリを持つとき。
- SELECT 文から INSERT を実行し、その SELECT 文が挿入テーブルを参照するとき。
- 複数のローに INSERT、UPDATE、または DELETE を実行し、その操作中に起動するように、対応するトリガがテーブルに定義されているとき。

これらの場合は、操作の影響を受けるレコードがワークテーブルに入れられます。キーセット駆動型カーソルなど、場合によっては、ワークテーブルにテンポラリインデックスが作成されます。要求されたレコードをワークテーブルに抽出する操作は、クエリの結果が出るまでかなりの時間がかかります。上記の最初の例でソートを実行するのに使用できるインデックスを作成すると、最初のいくつかのローを検索する時間が短縮されます。ただし、ワークテーブルを使用すると、すべてのローをフェッチするための合計時間を短縮できる場合があります。これは、ハッシュとマージソートに基づくクエリアルゴリズムが許可されるためです。これらのアルゴリズムでは、シーケンシャル I/O が使用されず、これは、インデックススキャンと併用されるランダム I/O より高速です。

オプティマイザは、各クエリを分析し、ワークテーブルを使用した場合に最適なパフォーマンスが得られるかどうかを判断します。こうした最適化を利用するのに、ユーザ側の操作は必要ありません。

## 注記

上述の INSERT、UPDATE、DELETE は通常は 1 回限りの操作のため、通常、パフォーマンスは問題にはなりません。ただし、問題が発生した場合は、文を書き直して、競合とワークテーブルが作成されるのを防止できる場合があります。これは、常に可能とはかぎりません。

### 1.8.3.2.15 ヒント: 効率的な SQL クエリの構築

クエリの処理パフォーマンスを向上させるために、より効率的なクエリを構築することを検討します。

以下のヒントは、クエリをより効率的に書き換えるために、クエリ処理中にオプティマイザによって選択される最適化を反映しています。これらの効率化をクエリに組み込むことによって、オプティマイザの作業量が減少する可能性があります。

ヒント	改善の前後	説明
不要な DISTINCT 条件の排除	<p>改善前:</p> <pre>SELECT DISTINCT p.ID, p.Quantity FROM Products p;</pre> <p>改善後:</p> <pre>SELECT p.ID, p.Quantity FROM Products p;</pre>	Products テーブルにプライマリキー p.ID が あり、結果セットの一部になっているので、最 初の文の DISTINCT キーワードは不要で す。
不要な DISTINCT 条件の排除	<p>改善前:</p> <pre>SELECT DISTINCT * FROM SalesOrders o JOIN Customers c ON o.CustomerID = c.ID WHERE c.State = 'NY';</pre> <p>改善後:</p> <pre>SELECT * FROM SalesOrders o JOIN Customers c ON o.CustomerID = c.ID WHERE c.State = 'NY';</pre>	最初のクエリには両方のテーブルのプライマ リキーが含まれているため、結果内の各ロー は異なります。

ヒント	改善の前後	説明
サブクエリのネスト解除	<p>改善前:</p> <pre>SELECT s.* FROM SalesOrderItems s WHERE EXISTS ( SELECT * FROM Products p WHERE s.ProductID = p.ID AND p.ID = 300 AND p.Quantity &gt; 20);</pre> <p>改善後:</p> <pre>SELECT s.* FROM Products p JOIN SalesOrderItems s ON p.ID = s.ProductID WHERE p.ID = 300 AND p.Quantity &gt; 20;</pre>	<p>ネストされたクエリをジョインとして書き換えると、より効率的な実行とより効果的な最適化につながる場合があります。一般的に、サブクエリのネスト解除は、FROM 句にテーブルを多くても1つだけ含んだ関連サブクエリに対して常に実行されます。これらの関連サブクエリは、ANY、ALL、EXISTS の各述部で使用されます。クエリのセマンティクス上、サブクエリが返すローは多くても1つだけであると判断できる場合は、非関連サブクエリ、または FROM 句に複数のテーブルを含んだサブクエリはフラットにされます。</p> <p>この例では、サブクエリは外部ブロック内の各ローに対して、多くても1つのローしか一致させられません。多くても1つのローしか一致させられないため、内部ジョインに変換できます。</p>
サブクエリのネスト解除	<p>改善前:</p> <pre>SELECT p.* FROM Products p WHERE EXISTS ( SELECT * FROM SalesOrderItems s WHERE s.ProductID = p.ID AND s.ID = 2001);</pre> <p>改善後:</p> <pre>SELECT DISTINCT p.* FROM Products p JOIN SalesOrderItems s ON p.ID = s.ProductID WHERE s.ID = 2001;</pre>	<p>改善前のクエリは、サブクエリ内に結合 EXISTS 述部があり、複数のローと一致する可能性があります。SELECT リストに DISTINCT を使って、内部ジョインに変換できます。</p>

ヒント	改善の前後	説明
サブクエリのネスト解除	<p>改善前:</p> <pre>SELECT * FROM Products p WHERE p.ID = ( SELECT s.ProductID FROM SalesOrderItems s WHERE s.ID = 2001 AND s.LineID = 1 );</pre> <p>改善後:</p> <pre>SELECT p.* FROM Products p, SalesOrderItems s WHERE p.ID = s.ProductID AND s.ID = 2001 AND s.LineID = 1;</pre>	サブクエリで、外部ブロックの各ローに対して多くても1つのローしか一致しない場合、比較しているサブクエリを削除します。
インデックスカラムを問い合わせる場合の IN 述部の使用の検討	<p>改善前:</p> <pre>SELECT * FROM SalesOrders WHERE SalesRepresentative = 902 OR SalesRepresentative = 195;</pre> <p>改善後:</p> <pre>SELECT * FROM SalesOrders WHERE SalesRepresentative IN ( 195, 902 );</pre>	書き換えられた形式では、IN リスト述部を検索インデックス可能として扱い、インデックス検索に利用できます。また、オプティマイザが IN リストをインデックスのソート順に合わせてソートし、検索効率を向上させることができます。定数だけ、またはクエリブロック1回の実行中に定数である値 (外部参照など) だけを IN リストに指定してください。

ヒント	改善の前後	説明
不要なジョインの排除	<p>改善前:</p> <pre data-bbox="544 405 940 562">SELECT s.ID, s.LineID, p.ID FROM SalesOrderItems s KEY JOIN Products p FOR READ ONLY;</pre> <p>改善後:</p> <pre data-bbox="544 629 940 808">SELECT s.ID, s.LineID, s.ProductID FROM SalesOrderItems s WHERE s.ProductID IS NOT NULL FOR READ ONLY;</pre>	<p>次の場合にジョインの排除を検討してください。</p> <ul data-bbox="963 439 1358 1099" style="list-style-type: none"> <li>• ジョインがプライマリーキーから外部キーへのジョインで、プライマリテーブルからのプライマリーキーのカラムだけがクエリで参照される場合。この場合は、プライマリキーテーブルが更新可能でないと、削除されます。</li> <li>• ジョインが同じテーブルの2つのインスタンス間におけるプライマリーキーからプライマリーキーへのジョインである場合。この場合は、一方のテーブルが更新可能でないと、そのテーブルは削除されません。</li> <li>• ジョインが外部ジョインであり、NULL 入力テーブル式は、外部ジョインの保護された側の各ローについて、多くても1つのローしか返さず、NULL 入力テーブル式で生成される式は、外部ジョイン以降のクエリの残りの部分で必要ない場合。</li> </ul> <p>この場合、ジョインはプライマリーキーから外部キーへのジョインであり、プライマリキーテーブル Products は削除できます。つまり、2番目のクエリは、SalesOrderItems テーブルの中で、Products への NULL 外部キーを持つローが結果に現れないため、セマンティック上は1番目のクエリと同じです。</p>
不要なジョインの排除	<p>改善前:</p> <pre data-bbox="544 1458 940 1637">SELECT s.ID, s.LineID FROM SalesOrderItems s LEFT OUTER JOIN Products p ON p.ID = s.ProductID WHERE s.Quantity &gt; 5 FOR READ ONLY;</pre> <p>改善後:</p> <pre data-bbox="544 1727 940 1839">SELECT s.ID, s.LineID FROM SalesOrderItems s WHERE s.Quantity &gt; 5 FOR READ ONLY;</pre>	<p>最初のクエリでは、NULL 入力テーブル式が保護された側のローに対して複数のローを生成できず、LEFT OUTER JOIN を超えて Products のカラムが使用されないため、OUTER JOIN を削除できます。</p>

ヒント	改善の前後	説明
大文字と小文字の不要な変換の排除	<p>改善前:</p> <pre>SELECT * FROM Customers WHERE UPPER(Surname) = 'SMITH';</pre> <p>改善後:</p> <pre>SELECT * FROM Customers WHERE Surname = 'SMITH';</pre>	<p>大文字と小文字を区別しないデータベースでは、最初のクエリは書き換えることができます。そのため、オプティマイザでは Customers.Surname のインデックスを使用することを検討できます。</p> <p>デフォルトでは、明示的なテキスト変換指示 (UPPER、UCASE、LOWER、LCASE の使用) が指定されないかぎり、データベースサーバで大文字と小文字を区別しない文字列比較が実行されます。大文字と小文字の不要な変換の排除により、述部を検索指数可能な述部にすることができます。検索指数可能な述部は、対応するテーブルのインデックス検索を使用できます。</p>
関数のインライン化の検討	<p>改善前:</p> <pre>CREATE FUNCTION F1( arg1 INT, arg2 INT ) RETURNS INT BEGIN RETURN arg1 * arg2 END; SELECT F1( e.EmployeeID, 2.5 ) FROM Employees e;</pre> <p>改善後:</p> <pre>SELECT CAST( e.EmployeeID AS INT ) * CAST( 2.5 AS INT ) FROM Employees e;</pre>	<p>次のいずれかの形式の場合、ユーザ定義関数をインライン化できます。</p> <ul style="list-style-type: none"> <li>RETURN 文を1つ含んでいます</li> <li>1つの変数を宣言し、その変数に代入して1つの値を返します</li> <li>1つの変数を宣言し、その変数に SELECT INTO して1つの値を返します</li> </ul> <p>このヒントは、テンポラリ関数、再帰関数、または NOT DETERMINISTIC 句のある関数には適用されません。</p> <p>また、このヒントは関数がサブクエリを使用して引数として呼び出された場合、または関数がテンポラリプロシージャの内部から呼び出された場合にも適用されません。</p>

ヒント	改善の前後	説明
単純なストアードプロシージャのインライン化の検討	<p>改善前:</p> <pre>CREATE PROCEDURE Test1( arg1 INT ) BEGIN SELECT * FROM Employees WHERE EmployeeID=arg1 END; SELECT * FROM Test1( 200 );</pre> <p>改善後:</p> <pre>SELECT * FROM ( SELECT * FROM Employees WHERE EmployeeID=CAST( 200 AS INT ) ) AS Test1;</pre>	<p>単一の SELECT 文としてのみ定義されたストアードプロシージャは、クエリの FROM 句で呼び出すときにインライン化できます。プロシージャがインライン化される場合は、派生テーブルとして書き換えられます。このヒントは、デフォルトの引数を使用し、単一の SELECT 文以外が本文に存在するプロシージャには適用されません。</p>

### 1.8.3.2.16 ヒント: クエリのパフォーマンスのモニタ

クエリのパフォーマンスをテストするために複数のツールが用意されています。

プロファイラを使用して、ストアードプロシージャ、ファンクション、トリガ、およびイベントのクエリをモニタリングします。

sa\_get\_request\_profile システムプロシージャおよび sa\_get\_request\_times システムプロシージャでも、クエリの実行時間を計測します。

次の表は、[%SQLANYSAMPI7%](#)¥SQLAnywhere のサブディレクトリに格納されているその他のツールを表しています。各ツールの完全なマニュアルについては、ツールと同じフォルダにある readme.txt ファイルを参照してください。

ツール	機能	ロケーション
fetchtst	結果セットの取り出しに必要な時間を決定します。	<a href="#">%SQLANYSAMPI7%</a> ¥SQLAnywhere ¥PerformanceFetch
odbcfet	結果セットの取り出しに必要な時間を決定します。このツールは fetchtst に似ていますがより限定的です。	<a href="#">%SQLANYSAMPI7%</a> ¥SQLAnywhere ¥PerformanceFetch
instest	ローをテーブルに挿入するための所要時間を決定します。	<a href="#">%SQLANYSAMPI7%</a> ¥SQLAnywhere ¥PerformanceInsert
trantest	データベース設計とトランザクションセットが与えられている特定のデータベースサーバ設定によって処理できる負荷を測定します。	<a href="#">%SQLANYSAMPI7%</a> ¥SQLAnywhere ¥PerformanceTransaction

### 1.8.3.2.17 ヒント: コストの高いユーザ定義関数の削減

クエリ中で何度も実行されるコストの高いユーザ定義関数を削減すると、パフォーマンスが改善されます。

### 1.8.3.2.18 ヒント: オートコミットモードをオフにする

オートコミットモードで実行する代わりに、SQL 文をグループ分けして各グループが 1 つの論理タスクを実行するようにします。

アプリケーションがオートコミットモードで実行されている場合、データベースサーバは各文を別々のトランザクションとして扱います。これは、それぞれの文の最後に COMMIT 文を付加して実行するのと同じ効果があります。

オートコミットを無効にする場合は、SQL 文の各論理グループの後に明示的にコミットを実行してください。また、論理トランザクションが大きい場合は、ブロッキングとデッドロックが発生する可能性があることに注意してください。

トランザクションログファイルを使用していない場合、オートコミットモードを使用するコストは高くなります。各文の終わりで、チェックポイントが強制的に実行されます。チェックポイントとは、多数の情報ページをディスクに書き込む操作です。

各アプリケーションインタフェースには、オートコミット動作を設定する独自の方法があります。Open Client、ODBC、JDBC インタフェースでは、オートコミットはデフォルトの動作です。

### 1.8.3.2.19 ヒント: インメモリモードの使用

最新のチェックポイント後にコミットされたトランザクションが失われても構わないアプリケーションには、インメモリモードを使用すると便利な場合があります。

大量のメモリを使用できる (通常、すべてのデータベースファイルをキャッシュ内に保持できる) システムを実行している場合、パフォーマンスの向上が求められるアプリケーションでこのモードを使用すると便利です。

インメモリモードは 2 種類から選択できます。非書き込みモードでは、コミットされたトランザクションはディスク上のデータベースファイルに書き込まれません。非書き込みモードを指定すると、複数の LOAD TABLE 文を同じまたは異なるテーブルで同時にアクティブにできます。データベースが停止した場合、または接続が切断された場合は、すべての変更が失われます。チェックポイント専用モードでは、データベースサーバはトランザクションログを使用しないため、最新のコミットされたトランザクションにリカバリすることはできません。ただし、チェックポイントログは有効になっているため、データベースを最新のチェックポイントにリカバリすることができます。

インメモリモードには別途ライセンスが必要です。

### 1.8.3.2.20 ヒント: ラグタイムミラーオプション設定の確認 (ミラーリングされているシステムのみ)

ラグタイムミラーオプションは、ミラーリングされているシステムのパフォーマンスに影響を与えます。

ミラーリングシステムのパフォーマンスが低い場合、ラグタイムミラーオプションに関連する可能性があります。ミラーサーバの処理があまりにも遅い場合、この設定を調整弁として、プライマリサーバ上のトランザクションのレートを遅くすることができます。

す。ラグタイムミラーオプションがパフォーマンスに影響を与えている可能性がある場合は、オプションを最大値に設定することを検討してください。

### 1.8.3.2.21 ヒント: 常にトランザクションログを使用する

トランザクションログを使用すると、データを保護できます。また、データベースサーバのパフォーマンスを向上できます。

トランザクションログなしで操作すると、データベースサーバは各トランザクションの終わりにチェックポイントを実行します。これにより、大量のリソースが消費されます。

トランザクションログを使用して操作すると、データベースサーバは変更が発生したときにその詳細を示すメモを書き込むだけです。新しいデータベースページすべてを、最も効率のいい時に一度に書き込むように選択できます。チェックポイントは、情報がデータベースファイルに入力され、矛盾せず、最新のものであるということを確実にします。

#### 警告

データベースファイルとトランザクションログファイルは、データベースサーバと同じ物理コンピュータに保存してください。または SAN や iSCSI 設定でアクセスできるようにしてください。リモートネットワークディレクトリにデータベースファイルやトランザクションログファイルを配置すると、パフォーマンスが低下したり、データが破壊されたり、サーバが不安定になったりする可能性があります。

### 1.8.3.2.22 ヒント: コミットの遅延の使用

遅延コミットによって、データベースの COMMIT 文に対する応答頻度を決定します。

データベースに対してコミットされる変更の頻度が高い場合、トランザクションログの書き込みの頻度が、データベース全体のパフォーマンスを左右する最大の要因になります。トランザクションログのパフォーマンスを改善する場合は、`delayed_commits` オプションを On に設定します。On に設定すると、データベースサーバは COMMIT のトランザクションログエントリがディスクに書き込まれるのを待たずに、直ちに COMMIT 文に応答します。Off に設定すると、アプリケーションは COMMIT がディスクに書き込まれるまで待たなければなりません。`delayed_commits` オプションをオンにすると、一部が埋まっているログページの複数回の書き換えが回避され、トランザクションログの書き込みが減少します。このオプションは、接続ごとまたは接続全体に対して設定できます。`delayed_commits` オプションをオンにすると、トランザクションログページがディスクにフラッシュされる前にサーバがダウンした場合、コミットされた操作が失われるリスクがあります。

### 1.8.3.2.23 ヒント: 小さいテーブルに関する統計収集

デフォルトでは、データベースサーバは 5 つ以上のローがあるテーブルに関する統計を作成します。5 つ未満のローを持つテーブルの統計を作成するには、`CREATE STATISTICS` 文を使用します。

ある文の処理中に収集された統計は、以降の文の効率的な実行方法を見いだすときに使用できます。

## 1.8.3.2.24 ヒント: 断片化の削減

データベースを変更する際に断片化が発生しますが、ファイル、テーブル、インデックスが必要以上に断片化されていると、パフォーマンスが低下することがあります。

断片化の削減は、データベースのサイズに比例して重要になります。

パフォーマンスが大幅に低下した場合は、次の方法で対処できます。

- 複数のテーブルで多数の削除/更新/挿入アクティビティを実行した場合は、テーブルやインデックスの断片化を削減するためにデータベースを再構築します。
- データベースをディスクパーティションに単独で入れ、ファイルの断片化を低減します。
- 利用できる Windows ユーティリティの 1 つを定期的に行って、ファイルの断片化を低減します。
- データベースを再編成して、データベースの断片化を低減します。
- REORGANIZE TABLE 文を使用して、テーブル内のローをデフラグしたり、DELETE によって散在したインデックスを圧縮します。テーブルを再編成すると、テーブルとインデックスの格納に使用される合計ページ数を減らし、インデックスツリーのレベル数も減らすことができます。

このセクションの内容:

### [ファイルの断片化の削減 \[1395 ページ\]](#)

ディスク断片化を減らすことは、データベースのサイズが大きくなるに従って重要になります。

### [テーブルの断片化削減 \[1396 ページ\]](#)

テーブルの断片化は、ローが連続して格納されていない場合、またはローが複数のページに分割されている場合に発生します。

### [テーブルの断片化を削減する方法 \[1396 ページ\]](#)

テーブルの断片化は、複数の方法で制御されます。

### [オブジェクトの断片化の詳細の表示 \[1398 ページ\]](#)

SQL Central の断片化タブから、オブジェクトの断片化情報を表示します。

### [ベーステーブルとインデックスの再編成 \[1398 ページ\]](#)

SQL Central の断片化タブから、ベーステーブルとインデックスを再編成します。

### [インデックスの断片化とスキューの削減 \[1399 ページ\]](#)

インデックスは特定のカラムの検索が高速になるように設計されていますが、インデックス付きテーブルに対して多数の DELETE 操作が実行されると、インデックスが断片化 (密度が下がる) およびスキューする (偏る) ことがあります。

## 1.8.3.2.24.1 ファイルの断片化の削減

ディスク断片化を減らすことは、データベースのサイズが大きくなるに従って重要になります。

Windows でデータベースを起動すると、データベースサーバが各 DB 領域内でファイルの断片化数を判断します。断片化の数が 1 より大きい場合は、データベースサーバメッセージウィンドウに次のパフォーマンス警告が表示されます。Database file "mydatabase.db" consists of nnn disk fragments

また、DBFileFragments データベースプロパティを使用して、データベースファイルの断片化の数を取得することもできます。

ファイルの断片化の問題を解決するには、データベースをディスクパーティションに単独で入れ、使用可能な Windows ディスクデフラグユーティリティの 1 つを定期的に行います。

## 1.8.3.2.24.2 テーブルの断片化削減

テーブルの断片化は、ローが連続して格納されていない場合、またはローが複数のページに分割されている場合に発生します。

これらのローには追加のページアクセスが必要になるため、データベースサーバのパフォーマンスを低下させます。

断片化がパフォーマンスに与える影響は、さまざまなものがあります。テーブルが極端に断片化されていても、メモリ内に収まっていて、かつアクセス方法によってページのキャッシュが許可されている場合、影響は最小限に抑えられる可能性があります。ただし、断片化されたテーブルにより、大量の I/O 処理が発生することがあります。分割されたローが頻繁にアクセスされ、かつ余計な I/O のコストがキャッシュによって削減されない場合は、パフォーマンスが大幅に低下する可能性があります。

テーブルの再編成とデータベースの再構築を行うと断片化を削減できますが、行う回数が多すぎても少なすぎても、パフォーマンスに影響を与えることがあります。次のツールや方法を実際を使ってみて、テーブルの許容可能な断片化レベルを判断してください。

断片化を削減しても依然としてパフォーマンスがよくない場合は、統計が不正確であるなどの別の問題が考えられます。

### テーブルの断片化レベルの判断

パフォーマンスを改善するためにデフラグを実行するかどうかを判断するうえで、テーブルの断片化のチェックを 1 回実行するだけでは役立ちません。データベースを再構築してからタブの断片化をチェックし、ベースラインとなる結果を取得してください。次に、長期間にわたってテーブルの断片化を定期的にチェックし、断片化における変化とパフォーマンス測定時の変化との相関を調べます。この方法によって、テーブルの断片化がパフォーマンスに影響をきたすまでの状態になる速度を確認でき、それによりテーブルのデフラグを実行する最適な頻度を決定できます。

データベーステーブルの断片化の程度に関する情報を取得するには、次のいずれかの方法を使用します。

- sa\_table\_fragmentation システムプロシージャを呼び出します。次に例を示します。

```
CALL sa_table_fragmentation( );
```

- SQL Anywhere プラグインの断片化タブ: 断片化タブを使用すると、ベーステーブルで sa\_table\_fragmentation システムプロシージャを実行した結果がグラフィックで表示されます。

## 1.8.3.2.24.3 テーブルの断片化を削減する方法

テーブルの断片化は、複数の方法で制御されます。

### PCTFREE の使用

データベースサーバでは、ローが少し大きくなることを見込んで各ページに余分な領域を確保します。ローが更新されたために、最初に割り付けられていた領域より大きくなると、そのローは分割され、最初のローのロケーションにはロー全体

が格納されている別のページへのポインタが入れられます。たとえば、空のローを UPDATE 文で埋めたり、テーブルに新しいカラムを挿入したりすると、ローが分割される可能性があります。別個のページに格納されるローが増えるほど、追加ページへのアクセス所要時間が長くなります。

テーブルページに今後の更新のための予約領域の割合を指定しておく、と、テーブルの断片化量を減らすことができます。この PCTFREE 指定は、CREATE TABLE、ALTER TABLE、DECLARE LOCAL TEMPORARY TABLE、または LOAD TABLE で設定できます。

テーブルの再編成

REORGANIZE TABLE 文を使用するか、SQL Central の断片化タブで再編成をクリックすると、特定のテーブルの断片化を解除できます。

データベースの再構築

データベースを2段階で再構築すると、システムテーブルを含むすべてのテーブルの断片化が解除されます。2段階で行うには、データをディスクにアンロードして保存してから、再ロードします。この方法で再構築すると、テーブルのローが並べ替えられ、クラスタドインデックスとプライマリキーで指定された順序になります。-ar、-an、-ac などのオプションを使用して一度に再構築を行うと、テーブルの断片化は減りません。

このセクションの内容:

[断片化タブ \(SQL Anywhere プラグインの場合\) \[1397 ページ\]](#)

断片化タブを使用して、ベーステーブルの断片化やそれらのテーブルのインデックスを表示したり、テーブルとインデックスを再編成したりすることができます。

## 1.8.3.2.24.3.1 断片化タブ (SQL Anywhere プラグインの場合)

断片化タブを使用して、ベーステーブルの断片化やそれらのテーブルのインデックスを表示したり、テーブルとインデックスを再編成したりすることができます。

### DB 領域マップでのズーム

断片化タブの下部ウィンドウ枠で DB 領域マップを開くと、デフォルトで、ズームレベルはウィンドウに合わせるに設定されます。ズームインするには DB 領域マップをクリックし、ズームアウトするには [Shift] キーを押しながらクリックします。DB 領域マップ内でクリックするか、[Shift] キーを押しながらクリックすると、ズームレベルの変更後、クリックしたページが中央に表示されます。

ツールバーボタンを使用して、次のレベルにズームすることもできます。

ツールバーボタン	定義
1:1	1 ページ: 1 ピクセル
64KB: 1	164 KB ブロック: 1 ピクセル (1 データベース読み込み)
ウィンドウに合わせる	ウィンドウのすべての使用可能な領域を使用

## 1.8.3.2.24.4 オブジェクトの断片化の詳細の表示

SQL Central の断片化タブから、オブジェクトの断片化情報を表示します。

### 前提条件

CHECKPOINT システム権限が必要です。

### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠でデータベースを選択します。右ウィンドウ枠で、**[断片化]** タブをクリックします。
3. 上部ウィンドウ枠からオブジェクトを選択します。下部ウィンドウ枠の DB 領域マップに断片化情報が表示されます。
  - ベーステーブルを選択すると、下部ウィンドウ枠の DB 領域マップにテーブル、テーブルの拡張ページ、該当するインデックスページが表示されます。
  - インデックスを選択すると、下部ウィンドウの DB 領域マップにインデックスページが表示されます。
4. **チェックポイントと再表示** をクリックして、チェックポイントを実行し、最新の断片化情報を確認します。
5. ページインデックスを表示します。
  - 下部ウィンドウの DB 領域マップで、カーソルを色付きの縦線の上に置き、その位置の最初と最後のページを確認します。
  - DB 領域マップで、**[Ctrl]** キーを押したままマウスポインタを色付きの縦線の上に置き、その位置のすべてのページインデックスを確認します。

### 結果

指定されたオブジェクトの断片化の詳細が、下部のウィンドウ枠の dbspace マップに表示されます。

## 1.8.3.2.24.5 ベーステーブルとインデックスの再編成

SQL Central の断片化タブから、ベーステーブルとインデックスを再編成します。

### 前提条件

オブジェクトの所有者であるか、REORGANIZE ANY OBJECT システム権限を持っている必要があります。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠でデータベースを選択します。右ウィンドウ枠で、[\[断片化\]](#) タブをクリックします。
3. 上部ウィンドウ枠からオブジェクトを選択します。下部ウィンドウ枠の DB 領域マップに断片化情報が表示されます。
4. 次のいずれかの方法を選択して、オブジェクトを再編成します。
  - [再編成](#) をクリックして、選択したオブジェクトに対して REORGANIZE TABLE 文を実行します。
  - 上部ウィンドウ枠から Interactive SQL の [\[SQL 文\]](#) ウィンドウ枠にオブジェクトをコピーします。SQL 文ウィンドウ枠にオブジェクトの REORGANIZE TABLE 文が表示されます。文を実行します。  
この方法は、オブジェクトの再編成を後で行う場合、またはオブジェクトを再編集しながら SQL Central を引き続き使用する場合に有用です。

## 結果

指定されたテーブルまたはインデックスが再編成されます。

### 1.8.3.2.24.6 インデックスの断片化とスキューの削減

インデックスは特定のカラムの検索が高速になるように設計されていますが、インデックス付きテーブルに対して多数の DELETE 操作が実行されると、インデックスが断片化 (密度が下がる) およびスキューする (偏る) ことがあります。

インデックスの密度は、インデックスページの平均の満杯度を反映しています。インデックススキューは、平均密度からの一般的な偏差を反映しています。オプティマイザが選択性推定を行う場合、このスキュー量は重要です。

データベース内に許容できないレベルの断片化またはスキューを含むインデックスがあるかどうかを判断するには、プロファイラを使用します。

sa\_index\_fragmentation システムプロシージャを使用して、インデックスの断片化およびスキューのレベルを確認することもできます。たとえば、次の文は sa\_index\_density システムプロシージャを呼び出して、Customers テーブルのインデックスを調べます。

```
CALL sa_index_density( 'Customers' );
```

TableName	TableId	IndexName	IndexID	IndexType	LeafPages	Density	Skew
Customers	718	CustomersKey	0	PKEY	1	0.127686	1.000000
Customers	718	IX_customer_name	1	NUI	1	0.789795	1.000000

データベースサーバはプライマリキーのインデックスを自動的に作成します。これらのインデックスでは、sa\_index\_density システムプロシージャの結果に、IndexID が 0 のインデックスが存在します。

リーフページ数が少ない場合は、密度とスキューの値を考慮する必要はありません。密度とスキューの値は、リーフページ数が多いときだけ重要です。リーフページ数が多い場合、密度の値が低いと断片化を表し、不均衡値が高いとインデックスが偏

っていることを示します。どちらも、パフォーマンス低下の要因になります。REORGANIZE TABLE 文を実行すると、この両方の問題に対処できます。

SQL Anywhere プラグインで **断片化** タブを使用して、ベーステーブルに関連付けられたインデックスのインデックス断片化のレベルを確認することもできます。

### 1.8.3.2.25 ヒント: テーブル構造の正規化

データベーステーブルに同じ情報の複数のコピーが含まれる場合があります (たとえば、複数のテーブルで繰り返されているカラム)、テーブルの正規化が必要な場合があります。

正規化によって、リレーショナルデータベース内の重複データは減少します。たとえば、従業員が複数の異なるオフィスに勤務しているとします。データベースを正規化するには、住所や代表電話番号などオフィスに関する情報を、各従業員用にコピーせず、個別のテーブルに入れておくことを考えます。

重複する情報量が少ない場合は、情報をコピーし、トリガなどの制約を使用して整合性を維持する方が良い場合があります。

### 1.8.3.2.26 ヒント: カスケードされた参照アクションを最小限に抑える

カスケードされた参照アクションは、コストが高くなります。これは、トランザクションごとに複数のテーブルが更新されるためです。このことがパフォーマンスに影響する可能性があります。

たとえば、Employees テーブルから Departments テーブルへの外部キーが ON UPDATE CASCADE を使用して定義されている場合、department ID を更新すると Employees テーブルが自動的に更新されます。カスケードされた参照アクションは便利ですが、アプリケーションの論理で実装する方がより効率的な場合があります。

### 1.8.3.2.27 ヒント: 制約の宣言

プライマリーキーと外部キーの関係およびその制約を宣言することで、パフォーマンスを改善することができます。

これは、コストモデルの推定精度が上がるため、ジョインが発生する場合に特に有効です。

関係を宣言しない場合は、インデックスの管理に必要な時間を節約することができます。異なるテーブルのカラム値間に暗示的な関係がある場合、テーブル間には、宣言されていないプライマリーキー - 外部キー関係が存在します。

### 1.8.3.2.28 ヒント: 異なるファイルの異なるデバイスへの配置

異なるデータベースファイルを異なる物理デバイスまたはドライブに入れると、データベースのパフォーマンスが向上する場合があります。

たとえば、1 つのディスクドライブがキャッシュとデータベースページとの相互スワップを実行中の場合も、別のドライブはトランザクションログファイルへの書き込みができます。このようなパフォーマンスの向上を得るためには、各ドライブを独立させます。小さい論理ドライブに分割した単一のディスクでは、利点はありませぬ。

ディスクドライブは、最新のプロセッサや RAM よりオペレーション速度が非常に遅くなります。しばしば、ディスクによるページの読み込みまたは書き込みをただ待っていることが、データベースサーバの処理を低速にする原因になっています。

データベースサーバでは、データベースファイル、トランザクションログファイル、トランザクションログミラー、およびテンポラリファイルの4種類のファイルが使用されます。これらのファイルは別々のドライブに置かれている必要があります。

メディア障害から保護するために、データベースファイルとトランザクションログファイルは別の物理ドライブに格納することをお奨めします。

トランザクションログミラーファイルとテンポラリファイルを別の物理ドライブに格納すると、データベースサーバの高速化に役立つ場合があります。トランザクションログファイルとトランザクションログミラーファイルが別のドライブに格納されていると、データベースサーバの書き込みが効率化されます。データベースサーバでテンポラリファイルが使用される必要があるとき、データベース全体のパフォーマンスはテンポラリファイルが格納されるドライブの速度に大きく依存します。テンポラリファイルを使用する操作の多くでは、データベースからも情報を取得する必要があるため、テンポラリファイルを別のドライブに格納することによって、それらの操作を同時に実行できるようになります。

1つのデータベースは13個までのファイルに保管されます(メインファイル1つと12のDB領域)。これらは複数の別のドライブに配置できます。テーブルを個別のDB領域に配置して、一般的なジョイン操作によって異なるDB領域から情報を呼び込めるようにします。

すべてのテーブルやインデックスをシステムDB領域以外のロケーションに作成した場合、システムDB領域は、チェックポイントログとシステムテーブルの保存にのみ使用されます。この設定は、パフォーマンス上の理由からチェックポイントログを他のデータベースオブジェクトとは別のドライブに保存する場合に便利です。別のDB領域にベーステーブルを作成するには、IN DBSPACE 句を使用するようにすべてのCREATE TABLE 文を変更して、代わりにDB領域を指定するか、テーブルを作成する前にdefault\_dbpace オプションの設定を変更します。テンポラリテーブルは、TEMPORARY DB領域だけに作成できます。

同様の方式では、テンポラリファイルとデータベースファイルのRAIDデバイスまたはストライプセットへの配置があります。これらのデバイスは論理ドライブとして動作しますが、ファイルを多くの物理デバイスに分散し、複数のヘッドを使用して情報にアクセスすることによってパフォーマンスを飛躍的に向上させます。

データベースサーバの起動時に-fc オプションを指定して、データベースサーバでファイルシステムがいっぱいになった場合のコールバック関数を実装できます。

### 1.8.3.2.29 ヒント: データベースの再構築

データベースを再構築すると、パフォーマンスが向上する場合があります。

再構築することで、データの編成を改善します。

### 1.8.3.2.30 ヒント: キーを使ったクエリのパフォーマンス改善

プライマリキーと外部キーを使用して、データベースのパフォーマンスを向上させることができます。

#### 例

次の例は、クエリを高速で実行するためのプライマリキーの使用法を示します。

```
SELECT *
```

```
FROM Employees
WHERE EmployeeID = 390;
```

データベースサーバでこのクエリを実行する最も簡単な方法は、Employees テーブルの 75 個のローすべてを調べて、各ローの EmployeeID 番号が 390 であるかどうかを確認することです。従業員が 75 人であれば、この方法でもそれほど時間がかかりませんが、何千ものエントリのあるテーブルでは長い時間がかかってしまいます。

各プライマリキーまたは外部キーによって埋め込まれる参照整合性制約は、各キーの宣言で暗黙的に作成されるインデックスの助けを借りて、データベースサーバによって確保されます。EmployeeID カラムは Employees テーブルのプライマリキーです。対応するプライマリキーインデックスによって、従業員番号 390 をすばやく検索できます。この検索は、Employees テーブルのローの数が 100 の場合も、1000000 の場合もほとんど同じ時間で行うことができます。

プライマリキーと外部キーについて、インデックスが個別に自動的に作成されます。これによって、データベースサーバで多数のオペレーションを効率的に実行できます。

### 1.8.3.2.31 ヒント: プライマリキー幅の縮小

プライマリキーに含まれるカラム数を減らすと、パフォーマンスが改善されます。

幅の広いプライマリキーは、2 つ以上のカラムで構成されます。プライマリキーに含まれるカラム数が多いほど、データベースサーバに負荷がかかります。

### 1.8.3.2.32 ヒント: テーブル幅の縮小

ワイドテーブルが存在し、パフォーマンスが低下している場合は、テーブルをさらに正規化してカラム数を減らすことを検討してください。

テーブルを正規化できない場合は、データベースページサイズを拡大するとパフォーマンスが改善されることがあります。特にほとんどのテーブルの幅が広い場合は有効です。

結合されたカラム (または個々のローのサイズ) がデータベースページサイズを超えるため、複数のデータベースページに分割する必要があるテーブルは、ワイドテーブルと呼ばれます。ローが多数のページに分割されるほど、データベースサーバは各ローの読み込みに時間がかかります。

### 1.8.3.2.33 ヒント: テーブル内のカラムの順序の確認

テーブル内のカラムの順序はパフォーマンスに影響します。

ローのカラムは、作成された順に逐次方式でアクセスされます。たとえば、ローの末尾のカラムにアクセスするために、データベースサーバは、ロー内でそのカラムより前にあるカラムも処理しなければなりません。そのため、サイズの小さいカラムや頻繁にアクセスされるカラムが、アクセス頻度の低いカラムや幅の広いカラムより前に配置されるように、テーブルのカラムの順序を配列してください。

幅の広いカラムとは、15 バイトより大きなサイズのカラム、LONG 型データのカラム (LONG VARCHAR など)、または XML として定義されたカラムを指します。プライマリキーカラムは、常にローの先頭に格納されます。

### 1.8.3.2.34 ヒント: コストの高いトリガを置き換える

使用しているトリガを評価して、データベースサーバで利用できる機能で置き換えられるトリガはないかを確認します。

たとえば、最終更新時間とユーザ情報でカラムを更新するトリガは、データベースサーバ内の対応する特別値で置き換えられます。また、既存のトリガにデフォルト設定を使用することでも、パフォーマンスを改善できます。

### 1.8.3.2.35 ヒント: 適切なページサイズの使用

データベースページサイズは、データベースのパフォーマンスに影響を与えます。

ページサイズが大きい場合も小さい場合も、それぞれ利点と欠点があります。

データベースサーバは、ページをできるだけ埋めようとします。空き領域が累積するのは、新しいオブジェクトが大きすぎて既存のページの空き領域に収まらない場合だけです。したがって、ページサイズを調整しても、データベース全体のサイズに大きく影響することはありません。

ページサイズを選択するときは、パフォーマンスとその他の動作をテストすることを強くお奨めします。そして、満足できる結果を得られた最小のページサイズを選択します。同じサーバで複数のデータベースが起動される場合は、正しく適切なページサイズを選択することが重要です。

小さいページは保持できる情報量が少ないため、特に、ページの半分よりも少し大きいサイズのローを挿入する場合に領域の使用効率が低下します。ただし、ページサイズが小さいと、同じサイズのキャッシュにより多くのページを格納できるため、データベースサーバをより少ないリソースで実行できます。メモリが限られている小型のコンピュータでデータベースを実行する場合に便利です。また、不特定のロケーションから少量の情報を取得するためにデータベースを使う場合にも便利です。

ページサイズが大きいと、データベースサーバはデータベースをより効率的に読み込めます。データベースの規模が大きい場合や、逐次テーブルスキャンを実行するクエリの場合にパフォーマンスが高くなる傾向があります。通常は、ディスクの物理的な設計のために、大きなブロックを少数取り出す方が、小さなブロックを多数取り出す場合よりも効率的です。大きいページサイズには、他の利点もあります。インデックスのファンアウトを改善することによってインデックスレベル数を減らし、カラム数の多いテーブルが作成できます。大きなページサイズを選択すると、同じサイズのキャッシュに収まるページ数が少なくなるので、キャッシュサイズを大きくする必要がある場合があります。キャッシュに十分なページを格納できない場合、データベースサーバは頻繁に使用されるページをディスクにスワップし始めるため、パフォーマンスが低下します。

ページサイズが大きいと、メモリも余分に必要になります。また、大容量のデータベースサーバキャッシュを常に利用できないかぎり、大半のアプリケーションでは、極端に大きなページサイズ (16 KB または 32 KB) の使用はお奨めしません。

データベースサーバのメモリの使用状況は、ロードされたデータベース数とデータベースのページサイズに比例します。ページサイズを選択するときは、パフォーマンステスト (およびテスト全般) を実行することを強くお奨めします。その上で、満足できる結果を得られた最小のページサイズ (4 KB 以上) を選択します。同じサーバで多数のデータベースが起動される場合は、正しい (かつ適切な) ページサイズの選択が重要です。

既存のデータベースのページサイズは変更できません。その代わりに、新しいデータベースを作成し、dbinit の -p オプションを使用してページサイズを指定します。たとえば、次のコマンドは 4 KB のページサイズのデータベースを作成します。

```
dbinit -dba DBA,passwd -p 4096 new.db
```

新しいページサイズでデータベースを作成するために、PAGE SIZE 句を指定した CREATE DATABASE 文を使用することもできます。

データベースサーバでは、DB 領域ファイル全体での各テーブルページの位置を示すビットマップがテーブルごとに作成されます。データベースサーバでは、テーブルページが 1 ページずつ読み込まれるのではなく、このビットマップを使用して大きな

ブロック (64 KB) で読み込まれます。グループ読み込みと呼ばれるこの方法で、ディスクへの I/O 操作の合計数が減り、パフォーマンスが向上します。ユーザは、ビットマップの作成や使用に関するデータベースサーバの条件を制御できません。

## ページサイズとインデックス

ページサイズは、インデックスにも影響を与えます。各インデックスルックアップでは、インデックスのレベルごとに 1 ページを読み込み、さらにテーブルページについて 1 ページを読み込む必要があります。また、1 回のクエリには数千のインデックスルックアップが必要になることがあります。ページのサイズはファンアウトにかなりの影響を与え、またテーブルに必要なインデックスの深さに影響します。大きなファンアウトは、通常、必要なインデックスレベル数が少ないことを意味し、検索パフォーマンスを大幅に向上できます。テーブル内のローの数が多い大規模なデータベースでは、高パフォーマンスのために 1 ページを 8 KB にできます。

## 分散読み込み

Windows システムでは、最小のページサイズ 4 KB を使用すると、データベースサーバはディスク上で連続した広範囲のデータベースページをキャッシュ内の適切な場所に直接読み込んで、64 KB のパツファを完全に回避できます。この機能によって、パフォーマンスを大幅に向上できます。

### i 注記

リモートコンピュータ上のファイルや UNC 名 (たとえば `¥¥mycomputer¥myshare¥mydb.db`) で指定されたファイルに対して、分散読み込みは使用されません。

## 1.8.3.2.36 ヒント: オートインクリメントを使用したプライマリキーの作成

オートインクリメント機能を使用したプライマリキーの作成は、値がデータベースサーバによって生成されるため、他のどの方法よりも高速です。

オートインクリメントのデフォルト設定は、ユニークな値を管理するカラムのすべてに使用できます。

## 1.8.3.2.37 ヒント: 適切なデータ型の使用

データに適切なデータ型を使用することで、パフォーマンスを改善できます。

データ型は、値の範囲、値に対して可能な演算、メモリでの値の格納方法など、特定のデータセットに関する情報を格納します。たとえば、数値データのみを含む値にデータ型 CHAR を割り当てないでください。また、コストの高い数値型や文字列型でなく、効率的なデータ型をできるだけ選択してください。

### 1.8.3.2.38 ヒント: バルクオペレーション方法の使用

非常に大量の情報をデータベースにロードする場合、各タスクに対する専用ツールを使用すると便利です。

大きなファイルをロードする場合は、データのロード後にテーブルにインデックスを作成するとより効率的になります。

### 1.8.3.2.39 ヒント: リソースガバナを使用する

リソースガバナを使用して、パフォーマンスを改善することができます。

リソースガバナは、リソースの制御に使用できる一連のデータベースオプションです。これらのオプションは、SET OPTION 文を使用して設定できます。

ユーザとロールのセットを作成すると、データベースの権限を管理できます。またデータベースのセキュリティと管理によって、個々のユーザが使用できるリソースを制限することもできます。

たとえば、他のユーザのデータベースへの接続速度を落とさないように、1つの接続がメモリや CPU を大量に使用することを防止できます。

#### 管理できるリソース

次のオプションを使用して、リソースを管理できます。

##### **max\_cursor\_count**

接続用のカーソルの数を制限します。

##### **max\_statement\_count**

接続用の準備文の数を制限します。

##### **priority**

接続からの要求を実行する優先度レベルを設定します。

##### **max\_priority**

接続の最高優先度レベルを制御します。

### 1.8.3.2.40 ヒント: クライアントとサーバとの間の要求数の削減

LazyClose と PrefetchOnOpen ネットワーク接続パラメータを使用して、クライアントとサーバ間の要求数を削減します。

ネットワークの遅延時間が長い場合、またはアプリケーションでカーソルを開く要求と閉じる要求を多数送信する場合は、このオプションでパフォーマンスが向上します。

### 1.8.3.2.41 ヒント: 圧縮の慎重な使用

1つまたはすべての接続について圧縮機能を有効にして、パケット圧縮時の最小サイズ制限を調整すると、パフォーマンスを大幅に向上できる場合があります。

#### 接続用パケットの圧縮

圧縮を有効にすることに効果があるかどうかを判断するために、アプリケーションを使用してネットワークのパフォーマンス分析を行ってから、運用環境で通信の圧縮を使用します。

圧縮機能を有効にすると、データパケットに格納される情報量が増大し、特定のデータセットの送信に必要なパケット数が減少します。パケット数を減らすと、データを高速で送信できます。

圧縮しきい値を指定すると、圧縮対象になるデータパケットの最小サイズを選択できます。圧縮しきい値の最適値は、使用するネットワークのタイプや速度など、さまざまな要素の影響を受けることがあります。

#### データベースの圧縮

圧縮レイヤによって IO 操作のコストが大幅に増加することがあるため、データベースおよびログファイルのファイルレベルまたはディスクレベルの圧縮を使用することはお奨めしません。

### 1.8.3.2.42 ヒント: パケットサイズを変更してパフォーマンスを向上させる

場合によっては、パケットサイズを大きくすることによってパフォーマンスが改善することがあります。

また、パフォーマンスが低下することもあります。パケットサイズを大きくすると、クライアントとデータベースサーバが使用するメモリの量も増えます。運用環境では、調整を行う前に、アプリケーションを使用してネットワークのパフォーマンス分析を行ってください。

パケットサイズを大きくすると、データベース要求の応答時間を短縮できます。特に、クライアントとデータベースサーバの間で大量のデータを転送する要求では大変有効です。たとえば、大きな結果セット (64 KB 以上) を多数持つアプリケーション、または多数の大きな (64 KB 以上) BLOB を高速なローカルネットワーク上で転送するアプリケーションでは、パケットサイズをデフォルトよりも大きくすることで、パフォーマンスが大きく向上する場合があります。

パケットサイズを設定するには、-p dbeng17/dbsrv17 サーバオプションを使用するか、接続文字列に CommBufferSize (CBSIZE) 接続パラメータを設定します。

### 1.8.3.3 パフォーマンス改善のためのインデックスの最適化

インデックス分析ツールであるインデックスコンサルタントを実行して、データベースに最適なインデックスセットについて推奨を取得します。

適切なインデックスセットを選択すると、データベースのパフォーマンスを向上させることができます。

インデックスコンサルタントは、単一のクエリまたは一連のデータベース要求（負荷と呼ばれる）に対するインデックスの選択プロセスを支援します。インデックスコンサルタントで候補インデックスを生成し、パフォーマンスに対する影響を判断します。異なる候補インデックスの効果を調べるために、インデックスコンサルタントは、インデックスセットごとにクエリの最適化を繰り返します。実際にクエリは実行しません。

どのインデックスを使用すると実行プランが改善されるかを判断するために、インデックスコンサルタントでは、これらのインデックスを使用してクエリ実行コストを見積もります。また、複数カラムのインデックスおよび単一カラムのインデックスを評価し、クラスターインデックスまたは非クラスターインデックスの影響を調べます。

このセクションの内容:

#### [クエリに推奨されるインデックスの確認 \(プロファイラ の場合\) \[1407 ページ\]](#)

プロファイラを使用して、1つまたは複数のクエリ (SELECT UPDATE または DELETE 文) に推奨されるインデックスを確認します。

#### [クエリに推奨されるインデックスの確認 \(Interactive SQL の場合\) \[1409 ページ\]](#)

Interactive SQL のクエリに対するインデックスコンサルタントの推奨内容にアクセスします。

#### [CREATE INDEX 文の一覧または LOAD TABLE 文の一覧の同時実行によるパフォーマンスの向上 \[1410 ページ\]](#)

BEGIN PARALLEL WORK 文を使用して、CREATE INDEX 文および LOAD TABLE 文の一覧を同時に実行します。

### 1.8.3.3.1 クエリに推奨されるインデックスの確認 (プロファイラ の場合)

プロファイラを使用して、1つまたは複数のクエリ (SELECT UPDATE または DELETE 文) に推奨されるインデックスを確認します。

#### 前提条件

SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。

#### 手順

1. プロファイラを起動し、データベースに接続します。
  1. 次のコマンドを実行して、Interactive SQL を起動し、データベースに接続します。

```
dbisql -c "START=dbeng17 -x TCPIP;UID=DBA;PWD=sql;DBF=C:¥Users¥Public  
¥Documents¥SQL Anywhere 17¥Samples¥demo.db"
```

- In Interactive SQL で、**ツール** > **プロファイラ** をクリックします。
- プロンプトが表示されたら、デフォルトのプロファイリングオプションを受け入れて、**OK** をクリックします。

データベースからのプロファイリング情報が、プロファイラに表示されます。

- Interactive SQL で、SELECT 文、UPDATE 文、または DELETE 文を実行します。
- プロファイラ で、次のオプションのうちの 1 つを選択してください。

オプション	説明
プロファイル開始以降実行されているすべてのクエリに対して、インデックスの推奨を生成します。	1. <b>ツール</b> > <b>負荷に対するインデックスの提示</b> をクリックして、ウィザードの指示に従います。
特定のクエリに対するインデックスの推奨を生成します。	1. <b>命令文タブ</b> を開き、SELECT 文、UPDATE 文、または DELETE 文を選択します。 2. <b>ツール</b> > <b>文に対するインデックスの提示</b> をクリックして、ウィザードの指示に従います。

- プロンプトが表示されたら、希望するインデックス推奨のタイプを選択します。

#### クラスタドインデックスの推奨

このオプションを選択すると、インデックスコンサルタントは、クラスタドインデックスと非クラスタドインデックスの効果を分析します。

一部の負荷に対しては、クラスタドインデックスを適切に選択すると、非クラスタドインデックスよりも大幅にパフォーマンスが改善されます。ただし、このためには REORGANIZE TABLE 文を使用してテーブルを再編成する必要があります。さらに、クラスタドインデックスの効果を考慮すると分析に時間がかかります。

#### 既存のセカンダリインデックスの維持

インデックスコンサルタントは、データベース内の既存のセカンダリインデックスセットを維持または無視して分析を実行できます。セカンダリインデックスは、一意性制約、プライマリキー、または外部キー以外のインデックスです。参照整合性制約を確保するためのインデックスは、アクセスプランの選択時に常に考慮されます。

- ウィザードの指示に従います。

提案される変更が表示され、変更を実装するための SQL スクリプトが提示されます。

インデックスコンサルタントの実行中に発生したエラーの報告を分析するには、"DBO"."ix\_consultant\_log" テーブルの sql\_code カラムと log\_message カラムを参照します。

- データベースに関する知識に対して提案された変更を見直し、提案によってパフォーマンスが向上することを確認します。SQL スクリプトを編集して、各自のデータベースに適合するようにカスタマイズすることができます。

分析によって生成され、提案されたインデックス名を変更することができます。

- SQL スクリプトを実行し、変更を実装します。

## 1.8.3.3.2 クエリに推奨されるインデックスの確認 (Interactive SQL の場合)

Interactive SQL のクエリに対するインデックスコンサルタントの推奨内容にアクセスします。

### 前提条件

DIAGNOSTICS システムロール、または次に示すすべてのシステム権限が必要です。

- SELECT ANY TABLE
- INSERT ANY TABLE
- DELETE ANY TABLE
- UPDATE ANY TABLE

### 手順

1. Interactive SQL で、データベースに接続します。

```
dbisql -c "START=dbeng17 -x TCPIP;UID=DBA;PWD=sql;DBF=C:¥Users¥Public¥Documents  
¥SQL Anywhere 17¥Samples¥demo.db"
```

2. *SQL Statements* ウィンドウ枠で、SELECT 文、UPDATE 文、または DELETE 文を実行します。
3. **▶ ツール ▶ インデックスコンサルタント ▶**をクリックして、ウィザードの指示に従います。

### 結果

インデックスの推奨内容が、**インデックスコンサルタントウィザードの概要**ウィンドウ枠に表示されます。

### 次のステップ

スクリプトを実行して実装する前に、推奨内容を見直してデータベースの要件を満たすことを確認してください。

### 1.8.3.3 CREATE INDEX 文の一覧または LOAD TABLE 文の一覧の同時実行によるパフォーマンスの向上

BEGIN PARALLEL WORK 文を使用して、CREATE INDEX 文および LOAD TABLE 文の一覧を同時に実行します。

#### 前提条件

一覧内の文を実行するための権限が必要です。

#### コンテキスト

BEGIN PARALLEL WORK 文内に一覧表示されている文で、同時に実行します。同時に実行できる文の数は、データベースサーバが稼働しているコンピュータの論理プロセッサの数により制限されます。また、max\_parallel\_statements オプション、-gta および -gtc データベースサーバオプション、および sa\_server\_option システムプロシージャの ProcessorAffinity オプションによって、さらに制限されます。

アンロードユーティリティ (dbunload) を使用してデータベースをアップロードする際に、reload.sql ファイルにある BEGIN PARALLEL WORK 文内の CREATE INDEX 文と LOAD TABLE 文のグループに -bp オプションを指定します。

#### 手順

次のオプションのうちの 1 つを選択してください。

オプション	アクション
CREATE INDEX 文の一覧を同時に実行する。	CREATE INDEX 文の一覧を使用して BEGIN PARALLEL WORK 文を実行します。例: <pre>BEGIN PARALLEL WORK   CREATE INDEX L_SHIPDATE_IDX     ON LINEITEM(l_shipdate);   CREATE INDEX O_ORDERDATE_IDX     ON ORDERS(o_orderdate);   ... END PARALLEL WORK;</pre>
LOAD TABLE 文の一覧を同時に実行する。	LOAD TABLE 文の一覧を使用して BEGIN PARALLEL WORK 文を実行します。例: <pre>BEGIN PARALLEL WORK   LOAD TABLE dba.Part   FROM 'D:¥¥data¥¥part.tbl'   FORMAT 'ASCII'   QUOTES OFF ESCAPES ON STRIP OFF HEXADECIMAL OFF   DELIMITED BY ' '</pre>

オプション	アクション
	<pre> ORDER OFF; LOAD TABLE dba.Supplier FROM 'D:¥¥data¥¥supplier.tbl' FORMAT 'ASCII' QUOTES OFF ESCAPES ON STRIP OFF HEXADECIMAL OFF DELIMITED BY ' ' ORDER OFF; ... END PARALLEL WORK; </pre>

## 結果

BEGIN PARALLEL WORK 文内に一覧表示された文と同時に、BEGIN PARALLEL WORK 文が実行されます。

## 関連情報

[max\\_parallel\\_statements オプション \[728 ページ\]](#)

[アンロードユーティリティ \(dbunload\) \[1176 ページ\]](#)

## 1.8.4 診断

データベースおよびデータベースサーバの問題を診断するために、たくさんのツールを使用することができます。

このセクションの内容:

[SQL Anywhere プロファイラ \[1411 ページ\]](#)

プロファイラは、データベースで発生するアクティビティをリアルタイムに記録する開発及び診断ツールです。また、パフォーマンスの問題に関する情報を分析します。

[その他の診断ツールと方法 \[1428 ページ\]](#)

プロファイラの他にも、データベースの現在のパフォーマンスを分析し、モニタリングするのに役立つさまざまな診断ツールと方法が用意されています。

### 1.8.4.1 SQL Anywhere プロファイラ

プロファイラは、データベースで発生するアクティビティをリアルタイムに記録する開発及び診断ツールです。また、パフォーマンスの問題に関する情報を分析します。

プロファイラを使用すると、次のような問題を特定するのに役立ちます。

- デッドロックおよびブロックされた接続
- 実行に時間のかかるクエリ、コストのかかるクエリ、頻繁に実行される文
- トリガ、イベント、ネストされたストアドプロシージャコールなどのコストの高い隠れたプロシージャ
- プロシージャ本文内の、問題の可能性のある領域

プロファイラを使用して、すべてのデータベースアクティビティに関する情報を収集したり (包括的なプロファイリング)、実行中の SQL 文のサブセットを収集したり (ターゲットプロファイリング) します。どちらのオプションでも、データサーバの負荷に関する統計を収集し、負荷を分析してアプリケーションのパフォーマンスに影響している可能性のある問題を特定および除外します。パフォーマンス調査は常に、この負荷の分析により開始されます。

## 包括的プロファイリングとターゲットプロファイリング

包括的プロファイリングでは、負荷に関するほとんどの情報を収集します。接続、SQL 文、イベント、コンソールメッセージ、Web サーバメッセージ、データベースサーバの内部操作に関する情報など、データベースで発生したすべてのアクティビティが収集されます。包括的プロファイリングでは、データベースで発生するアクティビティについて最大限の可視性が提供されます。対照的に、ターゲットプロファイリングではデータベースサーバの負荷と指定した基準を満たす SQL 文に関する情報のみ収集されます。

包括的プロファイリングセッション中は大量の情報が収集されるため、モニタリング対象のデータベースのパフォーマンスに影響します。それに比べ、ターゲットプロファイリングでは特定の基準を満たす SQL 文に関する情報のみを収集します。ターゲットプロファイリングでは収集される情報が少ないため、パフォーマンスへの影響は最小限となります。運用環境で動作しているデータベースで、ターゲットプロファイリングセッションを延長して実行することができます。

包括的プロファイリングは、ストアド SQL オブジェクトのプロファイリング、およびブロックされたオブジェクトと接続の確認のための機能を提供するための専用オプションです。負荷の分析には検討する情報の種類がさらに多くなるため、より大量の情報量となります。しかし、ターゲットプロファイリングでは、収集する SQL 文の収集基準を作成することができます。

## どちらのプロファイリングオプションを選択するか

データベースのパフォーマンスに懸念がある多くの場合、ターゲットプロファイリングをおすすめします。運用環境で動作しているデータベースをプロファイリングする場合は、特におすすめです。しかし、ターゲットプロファイリングで収集される情報が十分ではない場合、包括的プロファイリングを使用して、プロファイリングセッションの時間を区切ります。包括的プロファイリングは、データベースで発生するすべてのアクティビティに関する情報を収集するため、より強力なオプションです。ストアド SQL 文のプロファイリング機能などの追加機能も提供されます。

主として、データベースサーバの負荷に関する情報や、プロファイラによるアプリケーションの分析に関心がある場合は、ターゲットプロファイリングを選択します。ターゲットプロファイリングは、パフォーマンスに影響する可能性の高い SQL 文の特徴を把握している場合に向いています。また、多くのデータベースアクティビティを生成している複雑なアプリケーションがある場合にもおすすめです。というのも、ターゲットプロファイリングでは、関心のある情報を識別するようフィルタリングが必要な情報の量が削減されるためです。

包括的プロファイリングは、多くの原因が考えられるパフォーマンスの問題を調査する場合に有益です。設計を検証するためにアプリケーションを開発する場合に、包括的プロファイリングを使用します。

このセクションの内容:

[包括的プロファイリングセッションの実行 \(プロファイラ\) \[1413 ページ\]](#)

データベースで発生するすべてのアクティビティに関する情報を収集し、データベース内のオブジェクトが相互にやり取りする仕組みを調査し、データベースのパフォーマンスに影響する問題についてのプロファイラの分析を確認します。

#### [ターゲットプロファイリングセッションの実行 \(プロファイラ\) \[1414 ページ\]](#)

指定した基準を満たすデータベースサーバのパフォーマンス統計や SQL 文に収集される情報を制限します。

#### [ストアドプロシージャ、関数、イベントおよびトリガのさまざまな実行結果の比較 \(プロファイラ\) \[1417 ページ\]](#)

プロシージャ、関数、イベント、トリガなどのストアド SQL オブジェクトをプロファイリングして、システムのパフォーマンスに影響する可能性のあるオブジェクトや文を特定します。可能性のあるソリューションが特定されたら、テストを行い、結果を比較します。

#### [データベースサーバの診断情報のスナップショットの SAP サポートへの送信 \(プロファイラ\) \[1418 ページ\]](#)

SAP サポートに指示された場合、データベースサーバの内部パフォーマンス統計に関する情報をキャプチャし、この情報をファイルに保存して SAP サポートに送信します。

#### [トラブルシューティング: 負荷のサマリおよびサーバ負荷グラフを使用した、ハードウェアリソースによるパフォーマンスへの影響タイミングの検知 \(プロファイラ\) \[1419 ページ\]](#)

プロファイリングセッションを実行し、負荷のサマリ結果を確認して、ハードウェアリソースがパフォーマンスの制限要因となっているかどうかを調べます。

#### [トラブルシューティング: アプリケーションの論理の問題 \(プロファイラ の場合\) \[1421 ページ\]](#)

アプリケーションのコードやプロシージャ、トリガ、ファンクション、またはイベントにエラーがある場合は、データベースサーバで実行された、不正なコードに関連する文をすべて確認することをおすすめします。

#### [パフォーマンスプロファイリングのチュートリアル \(プロファイラ の場合\) \[1422 ページ\]](#)

これらのチュートリアルを使用して、デッドロック、速度が遅い文、インデックスの断片化、低速のプロシージャなどの一般的なパフォーマンス問題を分析する方法を学習します。

## 関連情報

[診断トレーシング \(廃止予定\) \[1438 ページ\]](#)

[SQL Anywhere コックピット \[1349 ページ\]](#)

[プロファイラユーティリティ \(dbprof\) \[1144 ページ\]](#)

### 1.8.4.1.1 包括的プロファイリングセッションの実行 (プロファイラ)

データベースで発生するすべてのアクティビティに関する情報を収集し、データベース内のオブジェクトが相互にやり取りする仕組みを調査し、データベースのパフォーマンスに影響する問題についてのプロファイラの分析を確認します。

## 前提条件

プロファイラを使用するには、SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。

## コンテキスト

包括的プロファイリングセッションで、プロファイラは、接続、SQL 文、イベント、コンソールメッセージ、およびサーバの内部操作に関する情報を収集します。

このオプションによって、実行された各文のテキストプランを収集します。プロファイリングセッション中にはグラフィカルプランのみ表示することができます。テキストプランは、プロファイラへの接続を切断した後に利用することができます。また、セッションを .sqlap ファイルに格納すると、テキストプランも含まれます。

## 手順

1. ▶ **スタート** ▶ **プログラム** ▶ **SQL Anywhere17** ▶ **管理ツール** ▶ **Profiler** ▶ をクリックして、プロファイラを起動します。
2. データベースに接続します。

たとえば、**接続**ウィンドウで、**接続文字列**を使用して**接続**を選択し、次の接続パラメータを指定します。

```
UID=DBA;PWD=sql;"DSN=SQL Anywhere 17 Demo"
```

3. 入力を要求されたら、**包括**を選択します。
4. アプリケーションを実行します。
5. **分析**タブを開き、**負荷のサマリ**をクリックして、負荷分析を確認します。
6. ▶ **ファイル** ▶ **名前を付けて保存** ▶ をクリックしてプロファイリングデータを保存します。

## 次のステップ

▶ **ファイル** ▶ **切断** ▶ をクリックして、プロファイラへの接続を切断します。

### 1.8.4.1.2 ターゲットプロファイリングセッションの実行 (プロファイラ)

指定した基準を満たすデータベースサーバのパフォーマンス統計や SQL 文に収集される情報を制限します。

## 前提条件

プロファイラを使用するには、SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。

## コンテキスト

ターゲットプロファイリングセッション中に収集される情報は少量なので、プロファイラ内でフィルタする情報も少なく、プロファイリングされたデータベースのパフォーマンスに与える影響は最小限です。ターゲットプロファイリングセッションを延長して実行することができるため、運用環境で動作しているデータベースで実行できます。プロファイラでは、データベースサーバの負荷についても分析およびレポートされます。

結果を `.sqlap` に保存すると、ターゲットプロファイリングセッションの一部として収集されたグラフィカルなプランが含まれます。

## 手順

1. **▶ スタート ▶ プログラム ▶ SQL Anywhere17 ▶ 管理ツール ▶ SQL Anywhere プロファイラ** をクリックして、プロファイラの開始と接続を行います。

たとえば、**接続**ウィンドウで、**接続文字列**を使用して**接続**を選択し、次の接続パラメータを指定します。

```
UID=DBA;PWD=sql;"DSN=SQL Anywhere 17 Demo"
```

2. 要求に応じて**ターゲット**を選択し、必要な基準を調整することで、指定した基準を満たす文に関する情報のみプロファイラで収集されます。

デフォルトではグラフィカルなプランが収集されますが、テキストプランが収集されるよう選択することができます。

3. アプリケーションを実行します。
4. プロファイラの**文**タブを開き、基準を満たす文を表示します。
5. **文**タブの上部ウィンドウ枠で、調べようとしている文に関連付けられたローをクリックします。

選択した文のそれぞれの実行に関する詳細情報は、一番下のテーブルに表示されます。

6. 下部ウィンドウ枠でローをクリックし、特定の文の実行に関する詳細を表示します。

データベースサーバが不適切な実行プランを選択していたために、文の実行時間が長くなる場合もあります。文に対してグラフィカルなプランが作成された場合は、**プラン**タブに表示されます。

7. **▶ ファイル ▶ 名前を付けて保存** をクリックして結果を保存します。
8. **▶ ファイル ▶ 切断** をクリックして、プロファイリングを停止します。

### 例

プロファイリングオプションの選択を要求されたら、**ターゲット**を選択し、**文が有効である最小時間**フィールドの 10 秒に **2** と指定します。

Interactive SQL で、データベースに接続し、長時間実行される次の文を実行します。

```
SELECT COUNT(*) FROM customers t1, customers t2, customers t3, customers t4;
```

プロファイラで、**文**タブを開きます。実行された文が表に一覧表示されます。

このセクションの内容:

[ターゲットプロファイリングセッション中に収集する SQL 文を指定する基準 \(プロファイラ\) \[1416 ページ\]](#)

次の基準を使用して SQL 文の特徴を定義し、ターゲットプロファイリングセッション中にプロファイラが収集できるようにします。

### 1.8.4.1.2.1 ターゲットプロファイリングセッション中に収集する SQL 文を指定する基準 (プロファイラ)

次の基準を使用して SQL 文の特徴を定義し、ターゲットプロファイリングセッション中にプロファイラが収集できるようにします。

実行するターゲットプロファイリングセッションを選択すると、デフォルトでは 10 秒以上動作する SQL 文を収集します。以下の表に挙げられている基準を使用して、このしきい値を調整したり、使用するプロファイラの別の特徴を指定したりします。プロファイラは、指定した特徴を満たす SQL 文のみ収集します。

プロファイラでは、ターゲットプロファイリングセッション中に収集された SQL 文のプロパティには、以下の表に挙げられた基準と同じ名前や定義のプロパティが含まれます。

基準	説明
active_time	データベースサーバで文の実行にかかった秒単位の時間。
actual_rows	文により返されたローの数。
approx_cpu_time	文による CPU の使用概算時間 (秒単位)。
blocked_io_count	I/O 待機のため文がブロックした回数。
blocked_lock_count	データベースロックのため文がブロックした回数。
blocked_lock_time	データベースロックのため文がブロックした秒単位の時間。たとえば、10 または 10.55 など。
conn_id	文が実行された接続 ID。
dbn	データベースの名前。
elapsed_time	カーソルの初回のフェッチと最新のフェッチ間の実際の秒数。
estimated_io	文の実行に必要な I/O の推定回数。
estimated_rows	文により返されるローの推定数。負の数の場合は、返されるローの最大数をオプティマイザが正確に推定できなかったことを示しています。
estimated_time	文の実行に必要な推定秒数。
procedure_creator	プロシージャ、関数、トリガ、またはイベントを作成したユーザの名前。
procedure_name	CurrentProcedure 接続プロパティの値。
line_number	プロシージャ内の行番号 (1..n)
plan	イベントタイプに関連付けられたフォーマットによる文のプラン。

基準	説明
plan_signature	実行プランを特徴付ける数。この数はジョイン方式によって異なります。
sql_string	文に関連付けられた SQL 文字列。この文字列は構築により生成されたもので、元の文字列と完全には一致しません。
stmt_handle	ほかのトレースイベントとのリンクに使用するステートメントハンドル。
traceback	プロシージャ/ユーザ関数のスタックトレース。
user	UserID 接続プロパティの値。

### 1.8.4.1.3 ストアドプロシージャ、関数、イベントおよびトリガのさまざまな実行結果の比較 (プロファイラ)

プロシージャ、関数、イベント、トリガなどのストアド SQL オブジェクトをプロファイリングして、システムのパフォーマンスに影響する可能性のあるオブジェクトや文を特定します。可能性のあるソリューションが特定されたら、テストを行い、結果を比較します。

#### 前提条件

プロファイラを使用するには、SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。

#### コンテキスト

ストアド SQL オブジェクトを構成する実際の SQL 文を表示するには、[プロファイリングタブ](#)を開きます。このタブでは、実行する各ストアド SQL オブジェクトについて、コードの各行だけでなく、各行の実行時間に関する統計も表示します。

#### 手順

1. **▶ スタート ▶ プログラム ▶ SQL Anywhere17 ▶ 管理ツール ▶ SQL Anywhere プロファイラ ▶**をクリックして、プロファイラの開始と接続を行います。

たとえば、[接続ウィンドウ](#)で、アクション[接続文字列を使用して接続](#)を選択し、次の接続パラメータを指定します。

```
UID=DBA;PWD=sql;"DSN=SQL Anywhere 17 Demo"
```

2. 入力を要求されたら、[包括](#)を選択します。
3. アプリケーションで、プロファイリングを行うストアド SQL オブジェクトを実行します。
4. [プロファイリングタブ](#)を開きます。

プロファイリング開始以降に実行した SQL オブジェクトが、上部ウィンドウ枠のテーブルで一覧表示されます。

5. 上部ウィンドウ枠で、実行したストアド SQL オブジェクトに関連付けられているローをクリックします。

該当の SQL オブジェクト固有の統計が下部ウィンドウ枠に表示されます。統計情報には、ストアド SQL オブジェクトの各行の実行時間などが含まれます。

ストアド SQL オブジェクトが複数回実行された場合、テーブルには、SQL オブジェクトのベースラインの実行と任意の後続の実行との時間の差異が含まれます。デフォルトでは、別の実行を指定しない限り、ストアド SQL オブジェクトの最初の実行がベースラインとして使用されます。プロファイラに再接続すると、ベースライン値はクリアされます。

6. 下ウィンドウ枠の内容を確認し、**ベースライン**をクリックして新しいベースラインを作成します。コード内の他の行に比べて実行時間が長い行は、より効率のよい別の方法で同じ機能を実行できるかどうかを分析してください。
7. アプリケーションのコードに変更を加えて、アプリケーションのストアド SQL オブジェクトを再実行します。
8. プロファイラの**プロファイリング**タブでパフォーマンスの変更を確認します。
9. 変更の問題がなくなるまで、アプリケーションの変更と再実行を続けます。
10. **ファイル** > **名前を付けて保存** をクリックしてプロファイリング結果を保存します。

#### 1.8.4.1.4 データベースサーバの診断情報のスナップショットの SAP サポートへの送信 (プロファイラ)

SAP サポートに指示された場合、データベースサーバの内部パフォーマンス統計に関する情報をキャプチャし、この情報をファイルに保存して SAP サポートに送信します。

#### 前提条件

プロファイラを使用するには、SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。

#### コンテキスト

収集した情報には、データベースサーバの起動以降に実行された、最もコストの高い文の実際の SQL コードが含まれます。また、これらの文に関連付けられたすべてのグラフィカルプランも含まれます。

#### 警告

スナップショットの内容を確認してから、Sybase 製品の保守契約を結んでいるサポートセンタに送信します。実行された SQL 文によっては、データベーススキーマ、知的財産、および機密情報である可能性があるデータが見えてしまう場合があります。たとえば、コストの高い文の 1 つが VALUES 句を使用して機密情報を挿入する INSERT 文である場合、スナップショットに機密情報が表示され、.sqlap ファイルに保存されます。

## 手順

1. ▶ **スタート** ▶ **プログラム** ▶ **SQL Anywhere17** ▶ **管理ツール** ▶ **SQL Anywhere プロファイラ** ▶ をクリックして、プロファイラの開始と接続を行います。

たとえば、**接続**ウィンドウで、アクション**接続文字列**を使用して**接続**を選択し、次の接続パラメータを指定します。

```
UID=DBA;PWD=sql;"DSN=SQL Anywhere 17 Demo"
```

2. 要求に応じて、**サポートオプション**を選択し、**OK** をクリックします。  
このオプションでは、.sqlap ファイルにこのセッションを保存する際に含まれるグラフィカルプランを収集します。
3. 省略可能です。プロファイラで、**分析タブ**を開き、**ステートメントパフォーマンスサマリ**をクリックしてスナップショットを表示します。  
上部ウィンドウ枠に、データベースサーバを起動以降に実行された最もコストの高い文が、テーブルで一覧表示されます。ローをクリックして、使用されたグラフィカルプランを含む、文のパフォーマンスサマリ情報を表示します。
4. ▶ **ファイル** ▶ **名前を付けて保存** ▶ をクリックし、ファイルをコンピュータに保存します。
5. 指示されたとおりに .sqlap ファイルを SAP サポートに送信します。

### 1.8.4.1.5 トラブルシューティング: 負荷のサマリおよびサーバ負荷グラフを使用した、ハードウェアリソースによるパフォーマンスへの影響タイミングの検知 (プロファイラ)

プロファイリングセッションを実行し、負荷のサマリ結果を確認して、ハードウェアリソースがパフォーマンスの制限要因となっているかどうかを調べます。

データベースの負荷が増えると、CPU サイクル、ディスク I/O 帯域幅、およびネットワーク I/O 帯域幅によってもパフォーマンスが制限されます。非効率的なアプリケーションやデータベースサーバが、パフォーマンスの問題の原因となる場合もあります。効率の悪い箇所が見つからなかった場合は、ハードウェアリソースを追加する必要がある場合があります。

リソースを追加してもスケーラビリティの問題が解決されず、コンピュータのパフォーマンスが改善されない場合があります。たとえば、データベースサーバが割り当てられた CPU を完全に使用している場合は、CPU リソースの追加が必要であることを示している可能性があります。ただし、データベースサーバが使用できる CPU を倍増しても、実行できる処理量が 2 倍になるとは限りません。

発生する可能性のあるハードウェアの問題を調査する場合、プロファイリングセッションを実行してから、**分析タブ**を開き、**負荷のサマリ**をクリックして、負荷のサマリ結果を確認します。負荷のサマリでは、自動的に負荷をスキャンし、サーバが CPU、ディスク、ネットワーク、またはデータベースワーカーによる制限と考えられる時間を特定します。それぞれの間隔にドリルダウンし、サーバで処理されていた操作を表示することができます。

負荷のサマリでは、次の条件がパフォーマンスの問題となっている時間を特定することができます。

#### CPU 使用率が高い

長期間にわたり、CPU 使用率が 90% 以上で、ディスク I/O およびネットワーク I/O が低いことを示している場合、システムが CPU バウンドとなっている可能性があります。**プロファイリングタブ**と**分析タブ**で実行時間が長いクエリをチェックして、実行をより速くできる可能性があるものを確認します。

CPU 使用率とアクティブな要求数が高く、ディスク I/O とネットワーク I/O が通常の場合は、ハードウェアのアップグレード、または、負荷を分散するためのミラーリングやスケールアウトソリューションの設定を検討します。

ディスク I/O が高い

長期間にわたり、ディスク I/O が異常に高く、CPU 使用率とネットワーク I/O 使用率が低い場合、システムが I/O バウンドとなっている可能性があります。より高速なハードディスクなどのハードウェアをシステムに追加したり、ディスク I/O を高速化する RAID アレイを使用したりすると、パフォーマンスが改善されることがよくあります。あるいは、アプリケーションを調整することで問題が解決することもあります。たとえば、ディスク I/O パフォーマンスの問題は、キャッシュサイズが小さ過ぎたり、インデックスの設定が不適切であったりすることが原因である場合があります。▶ ツール ▶ 負荷に対するインデックスの提示 ▶ をクリックして、アプリケーションのスループットを向上させる可能性のある、一連の推奨インデックスを作成します。

ネットワーク使用率が高い

長期間にわたり、ネットワーク I/O が異常に高く、CPU 使用率とディスク I/O 使用率が低い場合、システムがネットワークバウンドとなっている可能性があります。ネットワーク設定を調べるか、アプリケーションの調整を検討します。たとえば、ネットワークカードの問題があるか、または、ネットワーク上で大量のデータを送信している可能性があります。

アクティブな要求の数が多

アクティブな要求の数が多という事は、データベース内で何かを行おうとしている接続数が大量にあるということです。

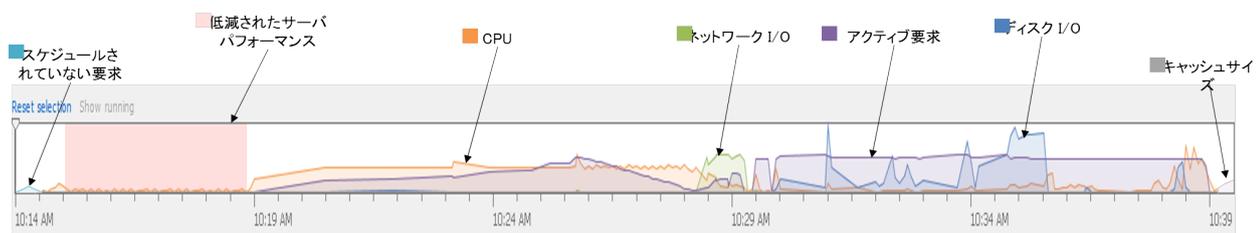
長期間にわたり、アクティブな要求の数が多く CPU 使用率が低い場合、調査対象は多数あります。例を以下に示します。

- ブロックおよびデッドロックが発生した接続。ブロックングタブで、ブロックまたはデッドロックが発生した接続に関連する問題を確認します。
- データベースのエラー。
- 特定のクエリ用に最適化できる、実行時間の長い文。

## サーバ負荷グラフ

サーバ負荷パネルで、CPU、ディスク、ネットワーク、およびサーバ要求のローデータを調査することもできます。サーバ負荷グラフでは、CPU、ディスク I/O、ネットワーク I/O の使用率のほか、アクティブな要求、スケジュールされていない要求、およびキャッシュサイズの数を追跡します。これらの統計値が高い場合でも、通常の状態を示すか問題のある状態を示すかは、システムによって異なります。

下記にサーバ負荷グラフの例を挙げます。



サーバ負荷グラフで診断する際は、指標を総合的に検討し、システム全体の状態を把握します。それぞれの指標を個別に検討しても、結論はほとんど得られません。サーバ負荷パネルのプリセットドロップダウンメニューを使用して、表示する値とグラフの設定方法を選択します。

これらの指標の折れ線グラフでは、縦の高さは使用量のみを表しています。プロファイラでは、CPU、ディスク、またはネットワークの処理能力を把握していません。これらの折れ線グラフを使用して使用傾向を識別し、グラフにカーソルを合わせて絶対値を確認します。

カタログをロックするようなバックアップなど、データベースサーバが内部プロセスを実行しているために、パフォーマンスの問題が発生する場合があります。サーバ負荷グラフのピンクの線やエリアは、内部プロセスによってデータベースサーバのパフォーマンスが低下した期間を示しています。内部プロセスには、チェックポイントの発行、バックアップの実行、クリーナの実行、キャッシュサイズの変更、テーブルの再編成、マルチプログラミングレベルの変更などがあります。パフォーマンスの問題の解消と同時にピンクの線が消えた場合は、内部プロセスが問題を引き起こしていたと考えられます。

パフォーマンスの問題の原因が内部プロセスであり、許容できない程度のパフォーマンスの低下が発生している場合、原因となるプロセスのオフピーク時への再スケジュールや、データベースミラーリング、スケールアウトソリューションなど、調査できることは数多くあります。

## 関連情報

[パフォーマンス向上のヒント \[1366 ページ\]](#)

[パフォーマンスプロファイリングのチュートリアル \(プロファイラの場合\) \[1422 ページ\]](#)

[ヒント: パフォーマンス向上のためのキャッシュの使用 \[1370 ページ\]](#)

[ヒント: インデックスの有効な使用 \[1380 ページ\]](#)

[ヒント: 効率的な SQL クエリの構築 \[1387 ページ\]](#)

[ヒント: 適切なページサイズの使用 \[1403 ページ\]](#)

[パフォーマンス改善のためのインデックスの最適化 \[1407 ページ\]](#)

### 1.8.4.1.6 トラブルシューティング: アプリケーションの論理の問題 (プロファイラの場合)

アプリケーションのコードやプロシージャ、トリガ、ファンクション、またはイベントにエラーがある場合は、データベースサーバで実行された、不正なコードに関連する文をすべて確認することをおすすめします。

SQL を動的に生成するアプリケーションでは、プロファイラを使用し、データベースサーバに渡される実際の SQL を確認することで、アプリケーションによる SQL の構築方法に関するエラーを検出するのに役立ちます。このようなエラーがあると、クエリの実行に失敗するか、予想とは異なる結果がクエリから返される可能性があります。たとえば、開発時にアプリケーションで SQL 構文エラーが発生したと報告されるが、失敗したクエリの SQL テキストを報告する機能がアプリケーションにないとなります。プロファイラを起動し、アプリケーションを実行すると、構文 (またはその他の) エラーを返した文を検索し、アプリケーションで生成された正確なテキストを確認できます。

プロシージャやトリガなどのデータベースオブジェクトを分析するには、SQL Central のデバッグを使用することもできます。ただし、プロファイラを使用して、指定のプロシージャで実行されたすべての文をトレースした方が効果的な場合もあります。たとえば、特定のストアドプロシージャが、呼び出し 1000 回のうち 1 回不正な結果を返すけれども、失敗する条件がわからないとします。この場合、デバッグでプロシージャのコードを 1000 回実行するのではなく、プロファイラを起動してアプリケーションを実行するという方法を取ることができます。次に、データベースサーバで実行された一連の文を確認し、プロシージャの不正な実行に対応する一連の文を見つけ、プロシージャが失敗した理由または予想外の動作をする条件を特定できます。プロシージャが予想外の動作をする条件がわかれば、プロシージャにブレークポイントを設定し、デバッグで詳細に調べることができます。

## 1.8.4.1.7 パフォーマンスプロファイリングのチュートリアル (プロファイラ の場合)

これらのチュートリアルを使用して、デッドロック、速度が遅い文、インデックスの断片化、低速のプロシージャなどの一般的なパフォーマンス問題を分析する方法を学習します。

このセクションの内容:

[チュートリアル: プロファイリング情報の収集 \(プロファイラ\) \[1422 ページ\]](#)

プロファイラを使用して、データベースに関するプロファイリング情報を収集し表示する方法を学習します。

[チュートリアル: 接続のブロックとデッドロックの診断 \(プロファイラ の場合\) \[1424 ページ\]](#)

プロファイラを使用してデータベースで発生している接続のブロックとデッドロックを特定する方法を学習します。

[チュートリアル: プロシージャのプロファイリング \(プロファイラ の場合\) \[1426 ページ\]](#)

あるストアドプロシージャについてさまざまな実装の実行時間を比較し、パフォーマンスが高いバージョンを特定します。

### 1.8.4.1.7.1 チュートリアル: プロファイリング情報の収集 (プロファイラ)

プロファイラを使用して、データベースに関するプロファイリング情報を収集し表示する方法を学習します。

#### 前提条件

プロファイラを使用するには、SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。

#### コンテキスト

プロファイラには現在実行している文に関する情報が表示されます。

#### 手順

- Interactive SQL を起動し、デフォルトの DBA ユーザとしてサンプルデータベースに接続します。
  - Interactive SQL で、**SQL > 接続** をクリックします。
  - アクションドロップダウンリストで、**接続文字列を使用して接続** を選択し、次の接続パラメータを指定します。

```
UID=DBA;PWD=sql;"DSN=SQL Anywhere 17 Demo"
```

- プロファイラを起動してデータベースに接続し、データベースのプロファイリングを開始します。

- a. **ツール** > **SQL Anywhere プロファイラ** をクリックします。
- b. 入力を要求されたら、**包括的**を選択します。

このオプションでは、データベースサーバに関する最も包括的な情報を収集します。また、これによりデッドロックとブロックされた接続を調査し、ストアド SQL をプロファイリングすることができます。

データベースから収集された診断情報が、プロファイラに表示されます。

3. プロファイラで、**分析**タブを開き**負荷のサマリ**をクリックして、重点的にパフォーマンスを向上させるエリアを強調表示した負荷分析を確認します。

調査は常に、負荷のサマリの確認により開始されます。負荷のサマリでは、次のような要素によりパフォーマンスが制限されているかどうかという質問に対処しています。

- ブロッキング
- データベースサーバのハードウェア
- サーバワーカー
- バックアップ、チェックポイント、または他の内部的なサーバアクティビティ

4. **操作**タブを開きます。

このタブでは、データベースでプロファイリングの開始以降実行されているすべての操作が一覧表示されます。

右ウィンドウ枠の棒グラフは、操作の処理にかかる時間を表しています。操作がブロックされた場合は、ブロックされていた時間を棒グラフ上のストライプの線で表します。文や操作についてさらに詳細を表示するには、該当の操作をダブルクリックしてプロパティウィンドウを開きます。

データベースサーバには、パフォーマンスに影響すると考えられる内部オペレーションがいくつかあります。たとえば、データベースのバックアップ、チェックポイントの発行、キャッシュサイズの変更、クリーナの実行、テーブルの再編成、マルチプログラミングレベルの変更などです。これらの内部プロセスが実行される際には、右ウィンドウ枠の背景が赤くなり、この間はサーバのパフォーマンスが低下する可能性があることを示します。

5. Interactive SQL で次の文を実行します。

```
SELECT COUNT( * ) FROM customers t1, customers t2;
```

6. プロファイラで、**操作**タブに表示される情報を、実行した文に関する操作のみに限定します。
  - a. **フィルタ**ボックスをクリックし、次にドロップダウンメニューから**フィルタ式**の**編集**をクリックします。
  - b. 実行に 5 秒より長くかかった文のみ表示するフィルタを作成します。

7. **ファイル** > **名前を付けて保存** をクリックし、プロファイリング結果をファイルに保存します。

8. プロファイリングセッションを停止します。

- a. **ファイル** > **切断** をクリックします。
- b. プロンプトが表示されたら、切断する前にプロファイリングの停止を選択してください。

プロファイラで、情報の収集が停止されます。グラフィカルなプランを除き、プロファイリングセッション中に収集した情報には引き続きアクセスできます。

## 結果

プロファイリング情報を収集しました。

## 次のステップ

次のチュートリアルに進みます。

### 1.8.4.1.7.2 チュートリアル: 接続のブロックとデッドロックの診断 (プロファイラ の場合)

プロファイラを使用してデータベースで発生している接続のブロックとデッドロックを特定する方法を学習します。

#### 前提条件

プロファイラを使用するには、SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。

#### コンテキスト

デッドロックは、複数のトランザクションが互いにブロックされている場合に発生します。たとえば、トランザクション A がテーブル B にアクセスする必要があるが、テーブル B はトランザクション B によってロックされ、トランザクション B はテーブル A にアクセスする必要があるが、テーブル A はトランザクション A によってロックされている場合などです。これは環状ブロッキングの競合が発生している状態です。

デッドロックが発生している状況は、SQLCODE -306 と -307 が返されることで分かります。デッドロックを解決するために、データベースサーバはデッドロックが作成された最後の文を自動的にロールバックします。文が絶え間なくロールバックされる場合は、パフォーマンス問題が発生しています。

#### 手順

1. Interactive SQL を起動し、デフォルトの DBA ユーザとしてサンプルデータベースに接続します。
  - a. Interactive SQL で、**SQL > 接続** をクリックします。
  - b. **アクションドロップダウンリスト**で、**接続文字列を使用して接続**を選択し、次のパラメータを指定します。

```
UID=DBA;PWD=sql;"DSN=SQL Anywhere 17 Demo"
```

2. デッドロックが発生しやすいようにデータベースを設定します。
  - a. 次の SQL 文を実行して、2 つのテーブルと 2 つのプロシージャを作成します。後で、これらのプロシージャを実行して、デッドロックを発生させます。

```
CREATE OR REPLACE TABLE "DBA"."deadlock1" (  
  "id" UNSIGNED BIGINT NOT NULL DEFAULT AUTOINCREMENT,  
  "val" CHAR(1) );  
CREATE OR REPLACE TABLE "DBA"."deadlock2" (  
  "id" UNSIGNED BIGINT NOT NULL DEFAULT AUTOINCREMENT,  
  "val" CHAR(1) );
```

```

    "id" UNSIGNED BIGINT NOT NULL DEFAULT AUTOINCREMENT,
    "val" CHAR(1) );
INSERT INTO "deadlock1"("val") VALUES('x');
INSERT INTO "deadlock2"("val") VALUES('x');
CREATE OR REPLACE PROCEDURE "DBA"."proc_deadlock1"( )
BEGIN
    LOCK TABLE "DBA"."deadlock1" IN EXCLUSIVE MODE;
    WAITFOR DELAY '00:00:20.000';
    UPDATE deadlock2 SET val='y';
END;
CREATE OR REPLACE PROCEDURE "DBA"."proc_deadlock2"( )
BEGIN
    LOCK TABLE "DBA"."deadlock2" IN EXCLUSIVE MODE;
    WAITFOR DELAY '00:00:20.000';
    UPDATE deadlock1 SET val='y';
END;
COMMIT;

```

3. プロファイラを起動してデータベースに接続し、データベースのプロファイリングを開始します。
  - a. **ツール** > **SQL Anywhere プロファイラ** をクリックします。
  - b. **OK** をクリックしてプロファイルオプションのデフォルト設定を受け入れ、データベースのプロファイリングを開始します。

データベースから収集された診断情報が、プロファイラの **操作** タブに表示されます。

4. Interactive SQL で、データベースへの別の接続を作成します。
  - a. **ウィンドウ** > **新しいタブ** をクリックします。
  - b. ユーザ名を **DBA**、パスワードを **sql** と指定し、サンプルデータベースに接続します。
5. 次の手順に従って、環状デッドロックを発生させます。
  - a. 最初の Interactive SQL タブで、次の SQL 文を実行します。

```
CALL "DBA"."proc_deadlock1" ();
```

- b. 最初の Interactive SQL タブで SQL 文を実行してから 20 秒以内に、2 番目の Interactive SQL タブで次の SQL 文を実行します。

```
CALL "DBA"."proc_deadlock2" ();
```

デッドロックが発生し、以下ようになります。

- 2 番目の Interactive SQL タブで、エラーメッセージが表示されます。
- 最初の Interactive SQL タブで、ステータスバーのロック区分に 2 つのロックが表示されます。ロック区分をクリックして詳細を表示します。

proc\_deadlock1 プロシージャは deadlock2 テーブルにアクセスする必要がありますが、このテーブルが proc\_deadlock2 プロシージャによってロックされていることによって、このデッドロックが発生しました。同様に、proc\_deadlock2 プロシージャは deadlock1 テーブルにアクセスする必要がありますが、このテーブルも proc\_deadlock1 プロシージャによってロックされています。

6. プロファイラでデッドロックを表示します。
  - a. **操作** タブで、**CALL "DBA"."proc\_deadlock1" ()** という実行文を探します。

右ウィンドウ枠の棒グラフは、文の処理にかかる時間を表しています。文がブロックされた場合は、棒グラフではブロックされていた時間をストライプの線で表します。文についてより詳しい情報を表示するには、文をダブルクリックします。
  - b. **ブロック** タブをクリックして、**デッドロック** をクリックします。

一番上のテーブルには、データベースのプロファイリング開始以降デッドロックに関連した文がグループ化されています。

選択した文のそれぞれの呼び出しに関する詳細情報は、一番下のテーブルに表示されます。

c. `CALL "DBA"."proc_deadlock1" ()` という実行文をクリックすると、下部ウィンドウ枠に詳細が表示されます。

7. **分析** タブの **概要** ウィンドウ枠をクリックして、ブロックによってパフォーマンスが制限されるかどうかを決定します。

**概要** ウィンドウ枠では、重点的にパフォーマンスを向上させるエリアを強調表示した負荷分析が提供されます。パフォーマンスがブロックによって制限される場合は、最も長時間ブロックしている文が表示されます。**設定** アイコンをクリックして、文がブロックされた場合にレポートが必要となる時間の長さを設定することができます。

8. **ファイル** > **切断** をクリックして、プロファイリングを停止します。

## 結果

デッドロックを作成し、その情報を収集しました。

## 次のステップ

省略可能です。このチュートリアルで作成したスタアドプロシージャを削除し、サンプルデータベースをもとの状態にリストアします。

### 1.8.4.1.7.3 チュートリアル: プロシージャのプロファイリング (プロファイラの場合)

あるスタアドプロシージャについてさまざまな実装の実行時間を比較し、パフォーマンスが高いバージョンを特定します。

## 前提条件

プロファイラを使用するには、SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。

## コンテキスト

プロシージャプロファイリングは、プロシージャ、ユーザ定義関数、イベント、システムトリガ、トリガの実行所要時間を示します。これらのオブジェクトについて、行ごとの実行時間を表示できます。データベース内のパフォーマンス向上のために調整するオブジェクトを特定したら、保存されている結果をベースラインとして設定します。コードに差分の変更を加え、変更するごとに再実行します。新しい結果をベースラインと比較して、変更が目的どおり機能しているかどうかを確認できます。

## 手順

1. Interactive SQL を起動し、デフォルトの DBA ユーザとしてサンプルデータベースに接続します。
  - a. Interactive SQL で、**SQL > 接続** をクリックします。
  - b. **アクション** ドロップダウンリストで、**接続文字列を使用して接続** を選択し、次のパラメータを指定します。

```
UID=DBA;PWD=sql;"DSN=SQL Anywhere 17 Demo"
```

2. プロファイラ を起動してデータベースに接続し、データベースのプロファイリングを開始します。
  - a. **ツール > SQL Anywhere プロファイラ** をクリックします。
  - b. **OK** をクリックしてプロファイルオプションのデフォルト設定を受け入れ、データベースのプロファイリングを開始します。

データベースから収集された診断情報が、プロファイラに表示されます。

3. Interactive SQL で、次の SQL 文を実行してスタアドプロシージャを作成し、テスト後に実行します。
  - a. 次の文を実行して、プロシージャ longproc を作成します。

```
CREATE OR REPLACE PROCEDURE LONGPROC ()
BEGIN
  DECLARE LOCAL TEMPORARY TABLE RET (pkey INT, fkey INT);
  FOR lp AS CURSOR CURSOR FOR SELECT EmployeeID AS emp_id FROM GROUPO.Employees
  DO
    INSERT INTO ret
      SELECT emp_id, so."ID"
      FROM GROUPO.SalesOrders so
      WHERE so.SalesRepresentative = emp_id;
  END FOR;
  SELECT *
  FROM ret;
END;
```

- b. 次の文を実行して、longproc プロシージャを呼び出します。

```
CALL DBA.longproc ();
```

4. プロファイラ で、**プロファイリングタブ** をクリックします。

**プロファイリングタブ**の一番上のテーブルにスタアド SQL オブジェクト (プロシージャ、ファンクション、トリガ、プロファイリングの開始以降実行されているイベントなど) がグループ化されています。

システムで実行に時間がかかっているスタアド SQL オブジェクトを見つけるには、**回数カラム**と**時間カラム**を調べます。これらのカラムによって、データベースサーバで処理されるオブジェクトの実行時間に関する情報が提供されます。実行のたびに時間がかかるスタアド SQL オブジェクト、または頻繁に実行されるオブジェクトに注目します。

5. 上部ウィンドウ枠で CALL longproc 文に対応するローをクリックします。

スタアドプロシージャの SQL コードが下部ウィンドウ枠に表示されます。表示された情報には、SQL コードの各行にかかった実行時間が含まれています。

6. **ベースライン**として**設定**をクリックし、結果を比較するためのベースラインを作成します。

**プロファイリングタブ**の内容がクリアされます。

7. longproc プロシージャをより効率的なものに編集した後、結果をテストします。

- a. Interactive SQL で次の文を実行します。

```
CREATE OR REPLACE PROCEDURE LONGPROC ()
BEGIN
```

```
SELECT e.EmployeeID, so."ID" FROM GROUPO.Employees e, GROUPO.SalesOrders so
WHERE so.SalesRepresentative = e.EmployeeID;
END;
```

- b. 次の文を実行して、新しい longproc プロシージャを実行します。

```
CALL DBA.LONGPROC ();
```

8. プロファイラの **プロファイリングタブ**の下部ウィンドウ枠で、結果を確認します。

回数および時間カラムに表示される差分のカラムは、ベースラインからの統計と、プロシージャの最新の実行による統計を比較した結果です。具体的には、プロシージャ内のコード行ごとに、それぞれの実行回数と実行時間を比較します。

このチュートリアルでは、更新されたプロシージャの SELECT 文の方が、ベースラインプロシージャの SELECT 文よりも実行時間が長いということです。

9. ▶ **ファイル** ▶ **切断** をクリックして、プロファイリングを停止します。

## 次のステップ

次のステップは任意です。このチュートリアルで追加したプロシージャを削除し、サンプルデータベース (demo.db) をもとの状態にリストアします。

## 1.8.4.2 その他の診断ツールと方法

プロファイラの他にも、データベースの現在のパフォーマンスを分析し、モニタリングするのに役立つさまざまな診断ツールと方法が用意されています。

パフォーマンスデータを分析するためのその他の方法には、次のものがあります。

ツール	詳細
診断トレーシング	<p><b>データベーストレーシング</b></p> <p>データベーストレーシングでは、収集するパフォーマンスデータのタイプをカスタマイズできます。このツールを使用して、特定のユーザやアクティビティのパフォーマンスをモニタリングできます。</p> <p><b>インデックスコンサルタント</b></p> <p>この機能では、データベース内のインデックスが分析され、改善のための推奨内容が表示されます。プロファイラまたは Interactive SQL からこのツールにアクセスすることができます。</p> <p>プロシージャプロファイリング SQL 文を使用してプロシージャプロファイリングを実行することもできます。</p>

ツール	詳細
その他のツール	<p><b>要求ロギング</b></p> <p>要求ロギングは、アプリケーションから受け取った要求と、アプリケーションに送られた応答のログを個別にテキストファイルに記録します。データベースサーバがアプリケーションに何を要求されているかを特定したい場合に最も役立ちます。特定のアプリケーションのパフォーマンスを分析するときに、データベースサーバとクライアントのどちらに原因があるか不明な場合は、要求ロギングから始めることもおすすめします。要求ロギングは、問題の原因となっている可能性の高い、データベースサーバに対する特定の要求を特定するためにも使用できます。要求ログは、診断トレーシングとイベントトレーシングによってもたらされる情報のサブセットを提供します。</p> <p><b>イベントのトレース</b></p> <p>イベントのトレースは運用環境で使用することをおすすめします。また、イベントのトレースによって、記録された情報を詳細に制御することができます。データベースサーバとアプリケーションの両方で、ユーザ定義のトレースイベントも、システム定義のトレースイベントも記録することができます。また、トレースイベントをカスタマイズして、パフォーマンスの問題を特定することもできます。</p> <p><b>実行プラン</b></p> <p>この機能では、実行プランを調べて、文に関連するデータベース内の情報にアクセスできます。Interactive SQL または SQL 関数を使用すると、実行プランを表示できます。実行プランは、さまざまなフォーマットで取り出すことができます。また、プランを保存することもできます。</p>

このセクションの内容:

[要求ロギング \[1430 ページ\]](#)

要求ロギングは、アプリケーションから受け取った要求と、アプリケーションに送られた応答のログを個別に記録します。

[タイミングユーティリティ \[1432 ページ\]](#)

Fetchst、Instest、Trantest など、いくつかのパフォーマンステストユーティリティが、`%SQLANYWHERE17%` `SQLAnywhere` にあります。

[データベースのパフォーマンスのモニタ \[1432 ページ\]](#)

データベースのパフォーマンスをリアルタイムにモニタリングします。

[プロシージャプロファイリング \(システムプロシージャの場合\) \[1434 ページ\]](#)

プロファイラは、ストアド SQL オブジェクトのプロファイリングに適していますが、システムプロシージャを使用して、ストアドプロシージャ、ファンクション、イベント、システムトリガ、およびトリガのプロシージャプロファイリング情報を表示することもできます。

[診断トレーシング \(廃止予定\) \[1438 ページ\]](#)

診断トレース機能は廃止予定です。SQL Anywhere プロファイラを使用して、データベースの問題を診断します。

### データベースサーバのモニタリング (Windows パフォーマンスモニタ) [1449 ページ]

データベース、サーバ、または接続に関連したカウンタを表示するには、Windows パフォーマンスモニタを使用します。

### インデックスの断片化の識別と修正 (SQL の場合) [1450 ページ]

sa\_index\_density ストアドプロシージャを使用してインデックスの断片化を識別し、次にインデックスを再構築して問題を解決します。

### テーブルの断片化の識別と修正 (SQL の場合) [1452 ページ]

データベースにテーブルの断片化が発生しているかどうかを確認し、必要に応じて、テーブルの断片化を修正する方法を学習します。

## 1.8.4.2.1 要求ロギング

要求ロギングは、アプリケーションから受け取った要求と、アプリケーションに送られた応答のログを個別に記録します。

データベースサーバがアプリケーションに何を要求されているかを特定したい場合に最も役立ちます。

特定のアプリケーションのパフォーマンスを分析するときに、データベースサーバとクライアントのどちらに原因があるか不明な場合は、要求ロギングから始めることもおすすめします。要求ロギングは、問題の原因となっている可能性の高い、データベースサーバに対する特定の要求を特定するためにも使用できます。

#### i 注記

要求ロギング機能で提供されるすべての機能とデータは、診断トレーシングを使用した場合も利用できます。診断トレーシングでは、それ以外の機能やデータも提供されます。

ログに取られる情報には、タイムスタンプ、接続 ID、要求タイプなどがあります。クエリについては、独立性レベル、フェッチされたローの数、カーソルタイプもあります。INSERT 文、UPDATE 文、DELETE 文については、対象のロー数と実行されたトリガ数もあります。

#### i 注記

要求ログには、難読化された機密情報を持つ文が含まれる場合があります。プレーンテキストでパスワードやキーをサーバに格納する場合、これらの情報は記録されます。機密情報が難読化されていない唯一のケースは、文に解析エラーがある場合です。

-zr サーバオプションを使用すると、データベースサーバの起動時に要求ロギングをオンにできます。-zo サーバオプションを使用すると、出力を要求ログファイルにリダイレクトして、さらに分析を進めることができます。-zn オプションと -zs オプションを使用すると、保存する要求ログファイルの数と要求ログファイルの最大サイズを指定できます。加えて、sa\_server\_option システムプロシージャを使用して、すでに稼働中のサーバで要求ロギングを設定することもできます。

#### i 注記

サーバオプションは、SQL Central の診断トレーシングに影響しません。ファイルベースの要求ロギングは、SQL Central の診断トレーシング機能とは完全に別個のものです。診断トレーシング機能では、データベース内の dbo 所有の診断テーブルを使用して要求ログ情報が格納されます。

sa\_get\_request\_times システムプロシージャは、要求ログを読み込み、ログの文とその実行時間をグローバルテンポラリテータブル (SATMP\_request\_time) に格納します。INSERT、UPDATE、DELETE の各文の場合は、記録される時間は、文が

実行された時間です。クエリの場合、記録される時間は、PREPARE から DROP (DESCRIBE/OPEN/FETCH/CLOSE) までの合計経過時間です。まだ開いているカーソルに関して、要求時間は提示されません。

改善候補の文については SATMP\_request\_time を分析します。低コストでも頻繁に実行される文が、パフォーマンスの問題を引き起こしている可能性があります。

sa\_get\_request\_profile を使用して sa\_get\_request\_times を呼び出し、SATMP\_request\_time を SATMP\_request\_profile という別のグローバルテンポラリテーブルにまとめることができます。また、このプロシージャは、文をグループ化し、呼び出しの回数、実行時間などの情報を提供します。

### 警告

tracetime.pl Perl スクリプトを使用してログが分析される場合は、要求ログが取得される間、max\_client\_statements\_cached オプションを 0 に設定して、クライアントでの文のキャッシュを無効にする必要があります。

### 例

要求ログへの出力は、sa\_server\_option システムプロシージャを使用してフィルタを適用し、特定の接続または特定のデータベースからの要求のみを含めるようにできます。これにより、アクティブな接続の多いデータベースサーバや複数のデータベースのあるデータベースサーバをモニタリングするときのログサイズを縮小できます。

接続に基づいてフィルタするには、次の手順に従います。

```
CALL sa_server_option( 'RequestFilterConn' , connection-id );
```

connection-id は CALL sa\_conn\_info( ) を実行して取得できます。

データベースに基づいてフィルタするには、次の手順に従います。

```
CALL sa_server_option( 'RequestFilterDB' , database-id );
```

database-id は、データベースに接続しているときに SELECT CONNECTION\_PROPERTY( 'DBNumber' ) を実行して取得できます。フィルタは、明示的にリセットされるまで、またはデータベースサーバが停止されるまで有効です。

フィルタをリセットするには、次の 2 つの文のいずれかを使用して、接続またはデータベースごとにフィルタをリセットします。

```
CALL sa_server_option( 'RequestFilterConn' , -1 );
```

```
CALL sa_server_option( 'RequestFilterDB' , -1 );
```

ホスト変数の値を要求ログに含めるには、次の操作を行います。

- -zr サーバオプションに値 hostvars を指定します。
- 次の文を実行します。

```
CALL sa_server_option( 'RequestLogging' , 'hostvars' );
```

要求ログ分析プロシージャ sa\_get\_request\_times は、ログ内のホスト変数を認識し、それらの変数をグローバルテンポラリテーブル SATMP\_request\_hostvar に追加します。

## 1.8.4.2.2 タイミングユーティリティ

Fetchst、Instest、Trantest など、いくつかのパフォーマンステストユーティリティが、`%SQLANYSAMPI7%` ¥SQLAnywhere にあります。

fetchst ユーティリティは、任意のクエリについてフェッチ速度を測定します。instest ユーティリティは、ローをテーブルに挿入するための所要時間を決定します。trantest ユーティリティは、特定のデータベース設計とトランザクションセットが与えられている特定のサーバ設定で処理できる負荷を測定します。

これらのツールを使用すると、統計情報付のグラフィカルなプランよりもさらに正確に測定でき、特定のサーバとデータベース構成について、達成可能な最善のパフォーマンス (たとえばスループット) の目安がわかります。

ツールの完全なマニュアルについては、ユーティリティと同じフォルダの `readme.txt` ファイルを参照してください。

## 1.8.4.2.3 データベースのパフォーマンスのモニタ

データベースのパフォーマンスをリアルタイムにモニタリングします。

データベースのパフォーマンスのモニタリングに使用する統計値のセットが用意されています。これらの統計値には数多くの方法でアクセスできます。

### SQL 関数

これらの関数を使用すると、アプリケーションからデータベースの統計値に直接アクセスできます。

### Windows パフォーマンスモニタ

Windows オペレーティングシステムに付属するモニタツールです。データベースサーバのオプションでモニタリング対象を調整することができます。

### UNIX 用パフォーマンス統計値ユーティリティ (dbstats)

データベースサーバ、データベース、UNIX 上で実行されているデータベースサーバの接続の統計値をモニタリングできます。

このセクションの内容:

#### [統計のモニタに使用される SQL 関数 \[1432 ページ\]](#)

接続ごと、データベースごと、またはサーバ全体のデータを提供するシステム関数のセットが用意されています。

#### [Windows パフォーマンスモニタ \[1434 ページ\]](#)

Windows パフォーマンスモニタを使用して、データベースサーバをモニタリングすることもできます。

### 1.8.4.2.3.1 統計のモニタに使用される SQL 関数

接続ごと、データベースごと、またはサーバ全体のデータを提供するシステム関数のセットが用意されています。

利用できる情報は、データベースサーバ名のような静的な情報からディスクとメモリの使用状況のようなパフォーマンス関連の詳細な統計にまで及びます。

## システム情報を取り出す関数

次の関数は、システム情報を取り出します。

### PROPERTY 関数

この関数は、サーバワイドに指定されたプロパティの値を返します。

### DB\_PROPERTY 関数と DB\_EXTENDED\_PROPERTY 関数

これらの関数は、指定されている場合にはそのデータベースの、デフォルトでは現在のデータベースの指定されたプロパティ値を返します。

### CONNECTION\_PROPERTY 関数と CONNECTION\_EXTENDED\_PROPERTY 関数

これらの関数は、指定されている場合はその接続の、デフォルトでは現在の接続の指定されたプロパティ値を返します。

取り出したいプロパティの名前のみ、引数として指定します。関数は、現在のサーバ、接続、またはデータベースの値を返します。

## クエリの効率を改善する

よりよいパフォーマンスを得るには、データベースのアクティビティをモニタするクライアントアプリケーションは PROPERTY\_NUMBER 関数を使用して、名前が付けられたプロパティを識別すべきです。その後そのプロパティ番号を使用して統計を繰り返し取り出すようにします。

こうして取得したプロパティ名は、さまざまなデータベースの統計値を得るのに使うことができます。トランザクションログページへの書き込み回数や、チェックポイント実行回数から、メモリキャッシュからインデックスリーフページを読み出した回数まで、幅広く使えます。

次の文のセットは、Interactive SQL から行うプロセスを示します。

```
CREATE VARIABLE propnum INT;  
CREATE VARIABLE propval INT;  
SET propnum = PROPERTY_NUMBER( 'CacheRead' );  
SET propval = DB_PROPERTY( propnum );
```

### 例

次に示す文は、変数 server\_name を現在のサーバ名に設定します。

```
SET server_name = PROPERTY( 'name' );
```

次に示すクエリは、現在の接続のユーザ ID を返します。

```
SELECT CONNECTION_PROPERTY( 'UserID' );
```

次に示すクエリは、現在のデータベースのルートファイル名を返します。

```
SELECT DB_PROPERTY( 'file' );
```

## 1.8.4.2.3.2 Windows パフォーマンスモニタ

Windows パフォーマンスモニタを使用して、データベースサーバをモニタリングすることもできます。

Windows パフォーマンスモニタ (PerfMon) では、ネットワーク通信の統計などのパフォーマンス統計が提供されます。Windows パフォーマンスモニタは、データベースサーバに対してクエリを実行するために共有メモリスキームを使用するので、統計値自体には影響を与えません。

データベースサーバのオプション、およびパフォーマンスモニタが `-ks`、`-ksc` および `-ksd` データベースサーバオプションを使用してモニタリングする接続またはデータベースの最大数を指定することができます。

Windows パフォーマンスモニタのインスタンスごとに 1 つのデータベースサーバのみをモニタリングすることができます。

## 1.8.4.2.4 プロシージャプロファイリング (システムプロシージャの場合)

プロファイラは、ストアード SQL オブジェクトのプロファイリングに適していますが、システムプロシージャを使用して、ストアードプロシージャ、ファンクション、イベント、システムトリガ、およびトリガのプロシージャプロファイリング情報を表示することもできます。

プロシージャプロファイリングを有効にし、以下のシステムプロシージャを実行して、プロシージャプロファイリングを行います。

### `sa_procedure_profile`

`sa_procedure_profile` システムプロシージャでは、各オブジェクト内の行の実行時間などの詳細なプロファイリング情報が表示されます。結果セット内の各行は、オブジェクト内の実行可能なコード行を表します。

### `sa_procedure_profile_summary`

`sa_procedure_profile_summary` システムプロシージャでは、各オブジェクトの合計実行時間が表示され、実行された全オブジェクトの概要を示します。結果セット内の各行は、1 つのオブジェクトの実行の詳細を表します。

これらのシステムプロシージャの結果では、表示されるオブジェクト数が、呼び出したオブジェクトよりも多い場合があります。これは、1 つのオブジェクトから別のオブジェクトが呼び出される場合があるからです。たとえば、トリガからストアードプロシージャが呼び出され、そのストアードプロシージャから別のストアードプロシージャが呼び出される場合があります。

このセクションの内容:

#### [プロシージャプロファイリングの有効化 \(SQL の場合\) \[1435 ページ\]](#)

`sa_server_option` システムプロシージャを使用して、プロシージャプロファイリングを有効にします。

#### [プロシージャプロファイリングのユーザ別のフィルタ \(SQL の場合\) \[1435 ページ\]](#)

他の接続によるデータベースの使用を妨害することなく、特定のユーザが使用しているプロシージャを確認します。

#### [プロシージャプロファイリングのリセット \(SQL の場合\) \[1436 ページ\]](#)

`sa_server_option` システムプロシージャを使用して、Interactive SQL からプロシージャプロファイリングをリセットします。

#### [プロシージャプロファイリングの無効化 \(SQL の場合\) \[1437 ページ\]](#)

`sa_server_option` システムプロシージャを使用して、Interactive SQL からプロシージャプロファイリングを無効にし、既存のデータをクリアします。

## 1.8.4.2.4.1 プロシージャプロファイリングの有効化 (SQL の場合)

sa\_server\_option システムプロシージャを使用して、プロシージャプロファイリングを有効にします。

### 前提条件

DIAGNOSTICS システムロールと MANAGE PROFILING システム権限が必要です。

### 手順

1. データベースに接続します。
2. sa\_server\_option システムプロシージャを呼び出して、ProcedureProfiling オプションを ON に設定します。

たとえば、次の CALL 文を実行します。

```
CALL sa_server_option( 'ProcedureProfiling' , 'ON' );
```

### 結果

プロシージャプロファイリングが有効になります。

## 1.8.4.2.4.2 プロシージャプロファイリングのユーザ別のフィルタ (SQL の場合)

他の接続によるデータベースの使用を妨害することなく、特定のユーザが使用しているプロシージャを確認します。

### 前提条件

DIAGNOSTICS システムロールと MANAGE PROFILING システム権限が必要です。

### コンテキスト

この機能は、接続がすでに存在する場合、または複数のユーザが同じユーザ ID で接続する場合に便利です。

## 手順

1. データベースに接続します。
2. 次のように sa\_server\_option システムプロシージャを呼び出します。

```
CALL sa_server_option( 'ProfileFilterUser' , 'userid' );
```

## 結果

userid の値は、モニタされるユーザを表します。

### 1.8.4.2.4.3 プロシージャプロファイリングのリセット (SQL の場合)

sa\_server\_option システムプロシージャを使用して、Interactive SQL からプロシージャプロファイリングをリセットします。

## 前提条件

DIAGNOSTICS システムロールと MANAGE PROFILING システム権限が必要です。

プロシージャプロファイリングが有効になっている必要があります。

## コンテキスト

プロファイリングをリセットすると、データベースは古い情報をクリアし、すぐにプロシージャ、ファンクション、イベント、トリガに関する新しい情報の収集を開始します。

## 手順

sa\_server\_option システムプロシージャを呼び出して、ProcedureProfiling オプションを RESET に設定します。

たとえば、次の CALL 文を実行します。

```
CALL sa_server_option( 'ProcedureProfiling' , 'RESET' );
```

## 結果

プロシージャプロファイリングがリセットされます。

### 1.8.4.2.4.4 プロシージャプロファイリングの無効化 (SQL の場合)

sa\_server\_option システムプロシージャを使用して、Interactive SQL からプロシージャプロファイリングを無効にし、既存のデータをクリアします。

## 前提条件

DIAGNOSTICS システムロールと MANAGE PROFILING システム権限が必要です。

## コンテキスト

プロファイリング情報の確認後に、プロファイリングを無効にするか、クリアします。プロファイリングを無効にすると、データベースはプロファイリング情報の収集を停止します。ただし、その時点までに収集された情報は、SQL Central の [プロファイル](#) タブに残ります。プロファイリングをクリアすると、データベースはプロファイリングをオフにして、SQL Central の [プロファイル](#) タブからプロファイリングデータをすべてクリアします。

## 手順

sa\_server\_option システムプロシージャを呼び出して、ProcedureProfiling オプションを OFF または CLEAR に設定します。

たとえば、次の CALL 文を実行してプロファイリングをオフにします。

```
CALL sa_server_option( 'ProcedureProfiling' , 'OFF' );
```

または次の CALL 文を実行してプロファイリングをクリアします。

```
CALL sa_server_option( 'ProcedureProfiling' , 'CLEAR' );
```

## 結果

オプション CLEAR を使用した場合、プロシージャプロファイリングは無効になり、プロファイリングデータはクリアされます。

## 1.8.4.2.5 診断トレーシング (廃止予定)

診断トレース機能は廃止予定です。SQL Anywhere プロファイラを使用して、データベースの問題を診断します。

データベースサーバによって生成された診断トレーシングのデータには、データベースサーバで処理された文のタイムスタンプと接続 ID が含まれる場合があります。クエリについては、独立性レベル、フェッチされたローの数、カーソルタイプ、クエリ実行プランも診断トレーシングのデータに含まれます。INSERT 文、UPDATE 文、DELETE 文については、影響を受けるロー数も含まれます。診断トレーシングを使用して、ロックとデッドロックに関する情報を記録し、パフォーマンス統計値を取得することもできます。

診断トレーシング中に収集されたデータを使用して、次の問題点の特定やトラブルシューティングなどの詳細なアプリケーションプロファイリングアクティビティを実行できます。

- 特定のパフォーマンスの問題
- 実行時間が異常に長い文
- 不正なオプションの設定
- オプティマイザで最適なプランが選択されない状況
- リソース (CPU、メモリ、ディスク I/O) の競合
- アプリケーションの論理の問題

トレーシングアーキテクチャは、堅牢かつ拡張性があります。要求ロギングで記録されるすべての情報だけでなく、個別要件に合った分析のための詳細情報も記録できます。

### i 注記

診断トレーシングでは、イベントは取得されませんが、イベント内の実行時間が長い文は取得されます。イベントを取得するには、プロシージャ内にイベントコードを埋め込み、イベントからプロシージャを呼び出します。プロシージャを実行する前に、詳細レベルの低いカスタムレベルを指定し、文の取得に条件を設定しないで、トレーシングを有効にします。

このセクションの内容:

#### [トレーシングセッションのデータ \(廃止予定\) \[1439 ページ\]](#)

診断トレーシングのデータは、トレーシングセッション中に収集されます。

#### [診断トレーシングの設定 \(廃止予定\) \[1439 ページ\]](#)

システムプロシージャを使用して診断トレーシングテーブルに格納されている設定を変更します。

#### [診断トレーシングのレベル \(廃止予定\) \[1440 ページ\]](#)

sa\_set\_tracing\_level システムプロシージャを使用して、トレーシングレベルを設定します。

#### [診断トレーシングの範囲 \(廃止予定\) \[1441 ページ\]](#)

範囲の値を使用すると、データベース内のアクティビティの特定の発生元だけをトレースできます。

#### [カスタマイズされた診断トレーシングレベル \(廃止予定\) \[1442 ページ\]](#)

診断トレーシングの設定は、複数のレベルに分類されていますが、これらのレベル内で設定をさらにカスタマイズして、診断トレーシングレベルとして参照することもできます。

#### [診断トレーシングのタイプ \(廃止予定\) \[1442 ページ\]](#)

診断トレーシングで選択できる診断トレーシングタイプがいくつかあります。

#### [診断トレーシングの条件 \(廃止予定\) \[1444 ページ\]](#)

特定の診断トレーシングタイプのトレーシングデータが記録されるために、条件が満たされる必要があります。

#### [現在の診断トレーシングの設定の確認 \(SQL の場合\) \(廃止予定\) \[1445 ページ\]](#)

sa\_diagnostic\_tracing\_level テーブルへのクエリで、現在有効になっている診断トレーシングの設定を取得します。

#### 診断トレーシングの設定 (廃止予定) [1446 ページ]

診断トレーシングの設定は、データベースに固有のもので、

#### 診断トレーシングセッションの作成 (SQL の場合) (廃止予定) [1447 ページ]

Interactive SQL で ATTACH TRACING 文を実行して、トレーシングセッションを開始します。

#### 診断トレーシング情報の分析 (廃止予定) [1448 ページ]

診断トレーシングのデータは、データベースサーバで発生し、診断トレーシングレベルとトレーシングセッションの設定に対応するすべてのアクティビティの記録です。

#### 外部トレーシングデータベースの作成 (コマンドラインの場合 (廃止予定)) [1448 ページ]

アンロードユーティリティ (dbunload) を使用して、トレーシングセッションのないトレーシングデータベースを手動で作成します。

### 1.8.4.2.5.1 トレーシングセッションのデータ (廃止予定)

診断トレーシングのデータは、トレーシングセッション中に収集されます。

ATTACH TRACING および DETACH TRACING 文を使用して、以下の方法でトレーシングセッションデータを取得します。

トレーシングセッションの実行中は、データベースサーバによって指定されたデータベースの診断情報が生成されます。生成されるトレーシングデータの量は、トレーシングの設定によって異なります。

プロファイリング対象のデータベースは、運用データベース、ソースデータベース、またはプロファイリング対象のデータベースと呼ばれます。トレーシングデータが格納されるデータベースは、トレーシングデータベースと呼ばれます。運用データベースとトレーシングデータベースは同じデータベースでもかまいません。ただし、運用データベースのサイズが増大するのを防ぐために、トレーシングデータを別のデータベースに格納します。データベースファイルのサイズは一度増大してしまうと、縮小することはできません。また、特に運用データベースが大きく、頻繁に使用される場合は、トレーシングデータの格納および管理にかかるオーバーヘッドを別のデータベースで処理した方が、運用データベースのパフォーマンスが向上します。

トレーシングデータが格納されるトレーシングデータベース内のテーブルは、診断トレーシングテーブルと呼ばれます。これらのテーブルの所有者は dbo です。

#### **i** 注記

トレーシング情報は、データベースのアンロードまたは再ロード操作の一環としてアンロードされません。トレーシング情報を別のデータベースに転送する場合は、sa\_diagnostic\_\* テーブルの内容を手動でコピーしてください。ただし、この方法はお奨めしません。

### 1.8.4.2.5.2 診断トレーシングの設定 (廃止予定)

システムプロシージャを使用して診断トレーシングテーブルに格納されている設定を変更します。

診断トレース機能は廃止予定です。SQL Anywhere プロファイラを使用して、データベースの問題を診断します。

トレーシングの設定は、sa\_diagnostic\_tracing\_level システムテーブルに格納されます。

SendingTracingTo と ReceivingTracingFrom の各データベースプロパティは、それぞれトレーシングデータベースと運用データベースを指定します。

### 1.8.4.2.5.3 診断トレーシングのレベル (廃止予定)

sa\_set\_tracing\_level システムプロシージャを使用して、トレーシングレベルを設定します。

現在のトレーシングレベルを参照するには、sa\_diagnostic\_tracing\_level テーブルを参照します。

パフォーマンスへの影響は、トレーシングデータが別のデータベースサーバ上にある (推奨) トレーシングデータベースに送信されるという前提で推定しています。

次の一覧は、診断トレーシングレベルを説明したものです。

#### レベル 0

このレベルでは、トレーシングセッションが実行されますが、トレーシングデータがトレーシングテーブルに送信されません。

#### レベル 1

パフォーマンスカウンタと実行された文のサンプリングが (5 秒に 1 回) 収集されます。このレベルでは、次の診断トレーシングタイプがあります。

- volatile\_statistics、1 秒ごとのサンプリング
- nonvolatile\_statistics、60 秒ごとのサンプリング

このレベルは、パフォーマンスにほとんど影響がありません。

#### レベル 2

このレベルでは、パフォーマンスカウンタと実行された文のサンプリングが (5 秒に 1 回) 収集され、実行された文がすべて記録されます。このレベルでは、次の診断トレーシングタイプがあります。

- volatile\_statistics、1 秒ごとのサンプリング
- nonvolatile\_statistics、60 秒ごとのサンプリング
- statements
- plans、5 秒ごとのサンプリング

このレベルは、パフォーマンスに中程度の影響があります (20% 以下のオーバーヘッドが発生します)。

#### レベル 3

このレベルでは、レベル 2 と同じ情報が記録されますが、プランのサンプリング頻度が高く (2 秒に 1 回)、ブロックとデッドロックの情報がより詳細です。このレベルには、次の診断トレーシングタイプがあります。

- volatile\_statistics、1 秒ごとのサンプリング
- nonvolatile\_statistics、60 秒ごとのサンプリング
- statements
- blocking
- deadlock
- statements\_with\_variables
- plans、2 秒ごとのサンプリング

このレベルは、パフォーマンスへの影響が最大です (20% を超えるオーバーヘッドが発生します)。

## 1.8.4.2.5.4 診断トレーシングの範囲 (廃止予定)

範囲の値を使用すると、データベース内のアクティビティの特定の発生元だけをトレースできます。

sa\_set\_tracing\_level システムプロシージャ値を使用します。

次は、診断トレーシングの範囲の一覧です。たとえば、特定の接続からの要求をトレースするように範囲を設定できます。範囲の値は、dbo.sa\_diagnostic\_tracing\_level 診断テーブルの scope カラムに格納されます。また、対応する引数 (通常はオブジェクト名やユーザ名などの識別子) が identifier カラムに格納されていることがあります。

scope カラムの値	説明
DATABASE	<p>データベース内で発生し、指定するレベルと条件に対応するすべてのイベントのトレーシングデータを記録します。高コストのクエリのソースを判断するために、バックグラウンドでのデータベースの長期のモニタリングや、短期の診断に使用します。</p> <p>DATABASE を指定するときに指定する識別子はありません。</p>
ORIGIN	<p>データベースの外部または内部から発生するクエリのトレーシングデータを記録します。</p> <p>範囲 ORIGIN を指定するときは、External または Internal のいずれかの識別子を指定できます。External は、データベースサーバ外で発生し、指定するレベルと条件に対応するクエリの文のテキストと、関連する詳細情報のログを取るよう指定します。Internal は、データベースサーバ内で発生し、指定するレベルと条件に対応するクエリに関する同じ情報のログを取るよう指定します。</p>
USER	<p>指定したユーザと、指定したユーザが作成した接続から発行されたクエリだけのトレーシングデータを記録します。特定のユーザに関連する問題のあるクエリを診断するために使用します。</p> <p>この範囲の識別子は、トレースを行う対象ユーザのユーザ ID です。</p>
CONNECTION_NAME または CONNECTION_NUMBER	<p>現在の接続で実行された文のトレーシングデータだけを記録します。これらの範囲は、ユーザに複数の接続があり、その 1 つで高コストの文が実行されている場合に使用します。</p> <p>この範囲の識別子は、それぞれ接続の名前または接続番号です。</p>

scope カラムの値	説明
FUNCTION、PROCEDURE、EVENT、TRIGGER、または TABLE	<p>指定するオブジェクトを使用する文のトレーシングデータが記録されます。オブジェクトが他のオブジェクトを参照する場合、参照先オブジェクトのデータもすべて記録されます。たとえば、イベントをトリガする関数を使用するプロシージャのトレースを行っている場合、3つのオブジェクトで、指定するレベルと条件に対応する文のログが取られません。特定のオブジェクトのコストが高い場合や、オブジェクトを参照する文が完了するまでの時間が異常に長い場合に使用します。</p> <p>TABLE スコープは、テーブル、マテリアライズドビュー、非マテリアライズドビューに使用します。</p> <p>このスコープの識別子は、オブジェクトの完全に修飾された名前です。</p>

### 1.8.4.2.5.5 カスタマイズされた診断トレーシングレベル (廃止予定)

診断トレーシングの設定は、複数のレベルに分類されていますが、これらのレベル内で設定をさらにカスタマイズして、診断トレーシングレベルとして参照することもできます。

診断トレーシングの設定をカスタマイズすると、診断トレーシングセッション内の不要なトレーシングデータの量を減らすことができます。たとえば、ユーザ AliceB のアプリケーションの実行は遅いが、他のユーザは同じ問題が発生していないとします。このとき必要なのは、AliceB のクエリがどのように実行されているかです。AliceB がアプリケーションで実行しているすべてのクエリとその他の文、および実行時間が長いクエリのクエリプランのリストを収集します。そのためには、診断トレーシングレベルを 3 に設定し、1 ~ 2 日のトレーシングデータを生成できます。ただし、このレベルは、他のユーザのパフォーマンスに大きく影響するので、AliceB のアクティビティだけをトレースすることをお奨めします。そのためには、診断トレーシングレベルを 3 に設定し、診断トレーシングの範囲を USER にカスタマイズし、ユーザ名として AliceB を指定します。診断トレーシングセッションを数時間実行し、結果を確認します。

sa\_set\_tracing\_level システムプロシージャを使用して診断トレーシング設定をカスタマイズします。

データの解釈が困難になるため、トレーシングセッションの実行中は診断トレーシングの設定を変更しないでください。

### 1.8.4.2.5.6 診断トレーシングのタイプ (廃止予定)

診断トレーシングで選択できる診断トレーシングタイプがいくつかあります。

各診断トレーシングタイプには、次に示すように、対応する条件が必要です。また、dbo.sa\_diagnostic\_tracing\_level 診断テーブルの trace\_type カラムに格納されます。対応する診断トレーシング条件が trace\_condition カラムに格納されていることがあります。

trace_type カラムの値	説明
VOLATILE_STATISTICS	<p>頻繁に変化するデータベースとサーバの統計値のサンプルを収集します。</p> <p>スコープと条件: この診断トレーシングタイプは、DATABASE スコープが必要で、SAMPLE_EVERY 条件をデータ収集の間隔として使用します。</p>
NONVOLATILE_STATISTICS	<p>頻繁に変化しないデータベースとサーバの統計値のサンプルを収集します。不揮発性の統計値は、揮発性の統計値よりも頻繁に収集できません。不揮発性の統計値を収集するには、揮発性の統計値を収集する必要があります。不揮発性の統計値のサンプリング間隔は、揮発性の統計値に指定する間隔の倍数である必要があります。</p> <p>スコープと条件: この診断トレーシングタイプは、DATABASE スコープが必要で、SAMPLE_EVERY 条件をデータ収集の間隔として使用します。</p>
CONNECTION_STATISTICS	<p>接続の統計値のサンプルを収集します。スコープがデータベースの場合は、データベースへのすべての接続の統計値が収集されます。スコープがユーザの場合は、指定するユーザのすべての接続の統計値が収集されます。スコープが CONNECTION_NAME または CONNECTION_NUMBER の場合、指定する接続の統計値だけが収集されます。CONNECTION_STATISTICS を収集するには、揮発性の統計値を収集する必要があります。サンプリング間隔は、VOLATILE_STATISTICS に指定する間隔の倍数である必要があります。</p> <p>スコープと条件: この診断トレーシングタイプは、DATABASE、USER、CONNECTION_NUMBER、CONNECTION_NAME スコープで使用できます。また、SAMPLE_EVERY 条件をデータ収集の間隔として使用します。</p>
BLOCKING	<p>指定するスコープと条件に従ってブロックに関する情報を収集します。スコープが CONNECTION_NAME または CONNECTION_NUMBER の場合、接続が別の接続をブロックしたとき、または別の接続によってブロックされたときにブロックを記録できます。</p> <p>スコープと条件: この診断トレーシングタイプはすべてのスコープで使用でき、NONE、NULL、SAMPLE_EVERY の収集条件のいずれかを使用できます。</p>
PLANS	<p>条件とスコープに従ってクエリ実行プランを収集します。</p> <p>スコープと条件: この診断トレーシングタイプはすべてのスコープで使用でき、NONE、NULL、SAMPLE_EVERY、ABSOLUTE_COST の収集条件のいずれかを使用できます。</p>

trace_type カラムの値	説明
PLANS_WITH_STATISTICS	<p>実行の統計値があるプランを収集します。プランはカーソルのクローズ時間に記録されます。RELATIVE_COST_DIFFERENCE 条件を指定した場合、出力内の統計値の一部は推測された統計値である可能性があります。</p> <p>スコープと条件: この診断トレーシングタイプはすべてのスコープで使用でき、収集条件のいずれかを受け入れます。</p>
STATEMENTS	<p>指定するスコープと条件の SQL 文を収集します。内部変数は、各プロセスが初めて実行されるときに収集されます。この診断トレーシングタイプは、STATEMENTS_WITH_VARIABLES、PLANS、PLANS_WITH_STATISTICS のいずれかの診断トレーシングタイプを指定した場合に自動的に追加されます。</p> <p>スコープと条件: この診断トレーシングタイプはすべてのスコープで使用でき、収集条件のいずれかを使用できます。</p>
STATEMENTS_WITH_VARIABLES	<p>SQL 文と文に付加された変数を収集します。各変数 (内部変数またはホスト変数) に割り当てられた値も収集されます。</p> <p>スコープと条件: この診断トレーシングタイプはすべてのスコープで使用でき、収集条件のいずれかを使用できます。</p>
OPTIMIZATION_LOGGING	<p>この診断トレースタイプは廃止予定です。このトレースタイプが指定された場合、ソフトウェア上では無視されます。</p>
OPTIMIZATION_LOGGING_WITH_PLANS	<p>この診断トレースタイプは廃止予定です。このトレースタイプが指定された場合、ソフトウェア上では無視されます。</p>

### 1.8.4.2.5.7 診断トレーシングの条件 (廃止予定)

特定の診断トレーシングタイプのトレーシングデータが記録されるために、条件が満たされる必要があります。

次の表は、設定可能な診断トレーシング条件を示します。次の表に示すように、ほとんどの条件には値が必要です。条件は、dbo.sa\_diagnostic\_tracing\_level 診断テーブルの trace\_condition カラムに格納されます。対応する値 (ミリ秒単位の時間) が value カラムに格納されていることがあります。

trace_condition カラムの値	説明
NONE または NULL	<p>レベルとスコープの条件を満たすトレーシングデータをすべて記録します。この条件を高コストの診断トレーシングレベル (たとえばプラン) と同時に長時間使用することはお奨めしません。</p>

trace_condition カラムの値	説明
SAMPLE_EVERY	最後のイベントが記録されてから指定された間隔以上の時間が経過した場合に、レベルとスコープの要件を満たすトレーシングデータを記録します。  値: この条件には、時間をミリ秒単位で表した正の整数を指定します。
ABSOLUTE_COST	実行コストが指定する値以上である文を記録します。  値: この条件には、ミリ秒単位のコスト値を指定します。
RELATIVE_COST_DIFFERENCE	予想実行時間と実際の実行時間の差が、指定する値以上である文を記録します。  値: この条件には、パーセント単位のコスト値を指定します。たとえば、予想よりも2倍以上低速の文をログに取るには、値 200 を指定します。

### 1.8.4.2.5.8 現在の診断トレーシングの設定の確認 (SQL の場合) (廃止予定)

sa\_diagnostic\_tracing\_level テーブルへのクエリで、現在有効になっている診断トレーシングの設定を取得します。

#### 前提条件

DIAGNOSTICS システムロールと MANAGE PROFILING システム権限が必要です。

#### 手順

1. データベースに接続します。
2. sa\_diagnostic\_tracing\_level テーブルに対して、enabled カラムが 1 のローを問い合わせます。

#### 結果

データベースサーバから、現在使用中の診断トレーシングの設定が返されます。enabled カラムが 1 の場合、設定が有効であることを示します。

## 例

次の文は、sa\_diagnostic\_tracing\_level 診断テーブルに問い合わせ、現在の診断トレーシングの設定を取得する方法を示します。

```
SELECT * FROM sa_diagnostic_tracing_level WHERE enabled = 1;
```

次の表は、クエリの結果セットの例を示します。

id	scope	identifier	trace_type	trace_condition	value	enabled
1	database	(NULL)	volatile_statistics	sample_every	1,000	1
2	database	(NULL)	nonvolatile_statistics	sample_every	60,000	1
3	database	(NULL)	connection_statistics	(NULL)	60,000	1
4	database	(NULL)	blocking	(NULL)	(NULL)	1
5	database	(NULL)	deadlock	(NULL)	(NULL)	1
6	database	(NULL)	plans_with_statistics	sample_every	2,000	1

## 1.8.4.2.5.9 診断トレーシングの設定 (廃止予定)

診断トレーシングの設定は、データベースに固有のものであります。

sa\_set\_tracing\_level システムプロシージャを使用して診断トレーシングレベルを変更します。このプロシージャでは、トレーシングセッションは開始されず、またトレーシングセッションが実行中だった場合には失敗します。

## 例

次の文は、sa\_set\_tracing\_level システムプロシージャを使用して診断トレーシングレベルを 1 に設定します。

```
CALL sa_set_tracing_level( 1 );
```

既存の設定は、診断トレーシングレベル 1 に関連付けられているデフォルトの設定で上書きされます。

## 1.8.4.2.5.10 診断トレーシングセッションの作成 (SQL の場合) (廃止予定)

Interactive SQL で ATTACH TRACING 文を実行して、トレーシングセッションを開始します。

### 前提条件

DIAGNOSTICS システムロールと MANAGE PROFILING システム権限が必要です。

### コンテキスト

トレーシングセッションを開始することをトレーシングの追加ともいいます。同様に、トレーシングセッションを停止することをトレーシングの分離ともいいます。トレーシングを開始、停止する SQL 文は、それぞれ ATTACH TRACING と DETACH TRACING です。

### 手順

1. データベースに接続します。
2. sa\_set\_tracing\_level システムプロシージャを使用してトレーシングレベルを設定します。次に例を示します。

```
CALL sa_set_tracing_level( 1 );
```

3. ATTACH TRACING 文を実行してトレースを開始します。
4. DETACH TRACING 文を実行してトレースを停止します。

### 結果

トレーシングセッションが作成され、完了します。

#### 例

この例は、現在のデータベースの診断トレーシングを開始し、トレーシングデータを別個のデータベースに格納し、格納するデータ量を 2 時間に制限する方法を示します。この例は、パスワード passwd と正しい権限を持っている DBA というユーザ ID が存在することを前提にしています。

```
ATTACH TRACING TO 'UID=DBA;PWD=passwd;Server=server47;DBN=tracing;Host=myhost '  
LIMIT HISTORY 2 HOURS;
```

この例は、現在のデータベースの診断トレーシングを開始し、トレーシングデータをローカルデータベースに格納し、格納するデータ量を 2 MB に制限する方法を示します。

```
ATTACH TRACING TO LOCAL DATABASE LIMIT SIZE 2 MB;
```

この例は、診断トレーシングを停止し、トレーシングセッション中に取得された診断データを保存する方法を示します。

```
DETACH TRACING WITH SAVE;
```

この例は、診断トレーシングを停止し、診断データを保存しない方法を示します。

```
DETACH TRACING WITHOUT SAVE;
```

## 1.8.4.2.5.11 診断トレーシング情報の分析 (廃止予定)

診断トレーシングのデータは、データベースサーバで発生し、診断トレーシングレベルとトレーシングセッションの設定に対応するすべてのアクティビティの記録です。

データを分析するときは、どのような設定だったかを考慮する必要があります。たとえば、予想していた文がトレーシングセッションになかった場合、文が実行されなかった可能性があります。高コストの文をトレースするように指定した条件を満たさなかった可能性もあります。

データベースサーバで実行されているアクティビティを詳細に調べる理由はさまざまです。パフォーマンスの問題のトラブルシューティング、リソースの使用量の予測による今後の負荷の計画、アプリケーションの論理のデバッグなどです。

### i 注記

診断トレーシングでは、イベントは取得されませんが、イベント内の実行時間が長い文は取得されます。イベントを取得するには、プロシージャ内にイベントコードを埋め込み、イベントからプロシージャを呼び出します。プロシージャを実行する前に、詳細レベルの低いカスタムレベルを指定し、文の取得に条件を設定しないで、トレーシングを有効にします。

## 1.8.4.2.5.12 外部トレーシングデータベースの作成 (コマンドラインの場合) (廃止予定)

アンロードユーティリティ (dbunload) を使用して、トレーシングセッションのないトレーシングデータベースを手動で作成します。

### 前提条件

DIAGNOSTICS システムロールと MANAGE PROFILING システム権限が必要です。

## 手順

1. データベースに接続します。
2. dbunload コマンドを実行して、スキーマを運用データベースから新しいトレーシング/データベースにアンロードします。

次に例を示します。

```
dbunload -c "UID=DBA;PWD=sql;Server=demo;DBN=demo" -an tracing.db -n -kd
```

この例は、-an オプションで指定した名前 (tracing.db) で新しいデータベースを作成します。-n オプションによって、プロファイリング対象のデータベース (この例では SQL Anywhere サンプルデータベース *demo.db*) から新しいトレーシングデータベースにスキーマがアンロードされます。-kd オプションによって、1 つの DB 領域ファイルのすべての DB 領域が配置されます。

3. トレーシングデータベースを別のコンピュータに保存する場合は、新しいロケーションにコピーします。

## 結果

分析データを格納する外部データベースが作成され、トレーシングセッションは作成されません。

### 1.8.4.2.6 データベースサーバのモニタリング (Windows パフォーマンスモニタ)

データベース、サーバ、または接続に関連したカウンタを表示するには、Windows パフォーマンスモニタを使用します。

#### 前提条件

Windows オペレーティングシステムのマニュアルを参照して、Windows パフォーマンスモニタを実行するのに必要なパーミッションと設定があることを確認します。

#### コンテキスト

Windows パフォーマンスモニタ (PerfMon) は、データベースサーバに対してクエリを実行するために共有メモリスキームを使用するので、統計値自体には影響を与えません。

-ksc および -ksd データベースサーバオプションを使用して、パフォーマンスモニタがモニタする接続またはデータベースの最大数を制御することができます。

-k および -ks データベースサーバオプションによって、Windows モニタがデータベースサーバをモニタできないようにします。

Windows パフォーマンスモニタのインスタンスごとに 1 つのデータベースサーバをモニタすることができます。

## 手順

1. モニタするデータベースサーバを起動し、共有メモリ接続を許容できるようにします。ログ統計が必要な場合、データベースサーバをサービスとして起動する必要があります。
2. パフォーマンスモニタを起動します。  
コマンドライン `perfmon` から次のコマンドを実行します。
3. ナビゲーションツリーで、**パフォーマンスモニタ**をクリックします。
4. カウンタを追加します。
  - a. 右ウィンドウ枠で、ツールバーのプラス記号ツール (+) をクリックします。
  - b. **パフォーマンスオブジェクト**リストから、次のいずれかを選択します。

### SQL Anywhere17 接続

このカウンタにより、単一の接続のパフォーマンスがモニタされます。この項目が表示されるためには、接続が存在する必要があります。

### SQL Anywhere17 データベース

このカウンタにより、単一のデータベースのパフォーマンスがモニタされます。

### SQL Anywhere17 サーバ

このカウンタにより、サーバ全体でパフォーマンスがモニタされます。

5. **カウンタ**リストから、閲覧する統計値をクリックします。  
*SQL Anywhere17 接続*または*SQL Anywhere17 データベース*をクリックした場合は、**インスタンス**ボックスから、モニタするデータベース接続またはデータベースを選択します。
6. カウンタを表示するには、**追加**をクリックします。
7. 表示するカウンタをすべて選択したら **[閉じる]** をクリックします。

## 結果

指定された統計情報が、Windows パフォーマンスモニタに表示されます。詳細な情報については、Windows パフォーマンスモニタのマニュアルを参照してください。

## 1.8.4.2.7 インデックスの断片化の識別と修正 (SQL の場合)

`sa_index_density` ストアドプロシージャを使用してインデックスの断片化を識別し、次にインデックスを再構築して問題を解決します。

## 前提条件

他にデータベースへの接続がないことを確認します。

sa\_index\_density システムプロシージャを実行するには、次のシステム権限に加え、そのシステムプロシージャに対する EXECUTE 権限が必要です。

```
MONITOR
MANAGE ANY STATISTICS
CREATE ANY INDEX
ALTER ANY INDEX
DROP ANY INDEX
CREATE ANY OBJECT
ALTER ANY OBJECT
DROP ANY OBJECT
```

テーブルのインデックスを変更するには、テーブルの所有者であるか、次のいずれかの権限を持っている必要があります。

```
そのテーブルに対する REFERENCES 権限
ALTER ANY INDEX システム権限
ALTER ANY OBJECT システム権限
```

## コンテキスト

運用データベースに断片化がないか定期的にチェックしてください。インデックスが作成されると、テーブルデータが読み込まれ、インデックスの値が論理順序に従ってインデックスページに記録されます。テーブル内のデータを変更した場合は、新しいインデックス値を既存の値の間に挿入できます。データベースサーバは、インデックス値の論理順序を維持するために、新しいインデックスページを作成し、移動された既存の値を調整する必要がある場合があります。この新しいページは、通常、その値が最初に格納されていたページとは隣接していません。このようなインデックスページの順序の崩れが累積することを、インデックスの断片化と呼びます。

インデックスの断片化が起こると、大きなローのブロックが継続的に挿入、更新、削除されるテーブルに対してよく実行されるクエリの実行時間が長くなるという現象が発生します。

## 手順

1. Interactive SQL でデータベースに接続し、以下と同様の文を実行して、特定のテーブルのインデックスの密度をテストします。

```
CALL sa_index_density( 'table-name' );
```

密度値の範囲は 0 ~ 1 です。値が 1 に近いほど、インデックスの断片化が少ないことを示しています。0.5 未満の値は、インデックスの断片化のレベルがパフォーマンスに影響を与える可能性があることを示しています。

2. 次の ALTER INDEX...REBUILD 文を実行して、インデックスの密度を向上させます。

```
ALTER INDEX PRIMARY KEY ON table-name REBUILD;
```

3. Interactive SQL を閉じます。

## 結果

Interactive SQL を使用して、インデックスの断片化が識別され、修正されました。

### 1.8.4.2.8 テーブルの断片化の識別と修正 (SQL の場合)

データベースにテーブルの断片化が発生しているかどうかを確認し、必要に応じて、テーブルの断片化を修正する方法を学習します。

#### 前提条件

他にサンプルデータベースへの接続がないことを確認します。

sa\_table\_fragmentation システムプロシージャを実行するには、MANAGE ANY STATISTICS または MONITOR システム権限に加え、そのシステムプロシージャに対する EXECUTE 権限が必要です。

REORGANIZE TABLE 文を実行するには、テーブルの所有者であるか、REORGANIZE ANY OBJECT システム権限を持っている必要があります。

#### コンテキスト

インデックスの断片化が起こると、大きなローのブロックが継続的に挿入、更新、削除されるテーブルに対してよく実行されるクエリの実行時間が長くなるという現象が発生します。

テーブルデータはデータベースページに格納されています。INSERT、UPDATE、DELETE などのデータ操作をテーブルに対して実行すると、ローが連続して格納されなかったり、ローが複数のページに渡って分割される場合があります。CPU のアクティビティが高い場合でも、テーブルの断片化は、テーブルのスキャンを必要とするクエリのパフォーマンスに悪影響を与える可能性があります。

#### 手順

1. Interactive SQL でデータベースに接続し、以下と同様の文を実行して、テーブルの断片化をテストします。

```
CALL sa_table_fragmentation( 'table-name' );
```

segs\_per\_row (ローごとのセグメント数) カラムの値が 1.1 よりも多い場合は、テーブルの断片化が存在します。断片化の程度が高いほど、パフォーマンスに悪影響を与える可能性があります。

2. REORGANIZE TABLE 文を実行して、テーブルの断片化を削減します。

```
REORGANIZE TABLE table-name;
```

3. Interactive SQL を閉じます。

## 結果

テーブルの断片化を診断し、テーブルを再編成して断片化を修正しました。

## 1.9 ユーザおよびデータベースのセキュリティ

データへの不正なアクセスとデータベース上での不正なアクティビティを防ぐために、多くの機能が提供されています。

このセクションの内容:

### [ユーザのセキュリティ \(ロールと権限\) \[1453 ページ\]](#)

権限付き操作を実行するためにロールベースのアクセス制御モデルが用意されています。

### [データのセキュリティ \[1594 ページ\]](#)

データベースには機密情報や個人情報などが含まれている場合があるので、データベースや格納されるデータのセキュリティを考慮した設計になっていることが重要です。

### [トランスポートレイヤセキュリティ \[1640 ページ\]](#)

トランスポートレイヤセキュリティ (TLS) は IETF 標準プロトコルであり、デジタル証明書とパブリックキー暗号方式を使用して、クライアント/サーバ通信をセキュリティ保護します。

### 1.9.1 ユーザのセキュリティ (ロールと権限)

権限付き操作を実行するためにロールベースのアクセス制御モデルが用意されています。

ロールベースのセキュリティモデルは、ユーザに付与する権限の完全な制御と詳細度を提供します。ユーザがデータベースで実行できる権限付き操作のそれぞれに、1 つ以上のシステム権限またはオブジェクトレベル権限が必要となります。

システム権限は、許可されたデータベースタスクを実行する権限です。たとえば、CREATE TABLE システム権限では、ユーザが自己所有テーブルを作成できます。

オブジェクトレベル権限は、指定したオブジェクトで許可されたタスクを実行する権限です。たとえば、TableA での ALTER 権限を持っていれば、ユーザはそのテーブルを変更できますが、他のテーブルは変更できません。

ロールは 1 つ以上のシステム権限、オブジェクトレベル権限、またはロールのコレクションです。ロールを他のロールに付与し、ロール階層を作成することができます。ロールをユーザに付与することは、ロールの基本となるシステム権限をユーザに付与することに相当します。

新しい、または移行された各データベースには、使用開始に利用できるロールの事前定義済みセットが含まれます。これらのシステムロールは、ロールベースのセキュリティを実装するための出発点として動作します。

このセクションの内容:

## [ロール \[1455 ページ\]](#)

ロールベースのセキュリティモデルには 3 種類のロールがあります。システムロール、ユーザ定義ロール (ユーザ拡張ロールを含む)、および 互換ロールです。

## [権限 \[1496 ページ\]](#)

権限とは、システムで権限付き操作を実行するための権利です。

## [ユーザ \[1530 ページ\]](#)

すべての新規ユーザには、自動的に PUBLIC システムロールが付与されます。

## [グループ \[1536 ページ\]](#)

グループとは、グループ内の他のすべてのユーザに共通のロールおよび権限のセットを所有するユーザのセットです。

## [ロールと権限の継承 \[1537 ページ\]](#)

ロールまたはユーザが持つ権限は、複数のカテゴリにグループ化できます。

## [同一化 \[1547 ページ\]](#)

ユーザは、一時的にデータベースの別のユーザの ID を仮定します。これは同一化とも呼ばれ、同一化しているユーザの権限のスーパーセットを持つ場合に操作を実行できます。

## [ロールベースのセキュリティ階層のプランと実装 \[1551 ページ\]](#)

ロールベースのセキュリティ階層を計画し実装する必要があります。

## [ネストされたオブジェクトの所有権 \[1552 ページ\]](#)

ビューとプロシージャは、さまざまなユーザが所有する基本のオブジェクトにアクセスできます。

## [パスワードおよびユーザ ID の制限事項と考慮事項 \[1554 ページ\]](#)

ユーザは、データベースに接続可能なパスワードを持っている必要があります。

## [チュートリアル: ロールと権限の付与 \(SQL Central の場合\) \[1559 ページ\]](#)

ロールベースのセキュリティモデルでは、ユーザが特定のデータベース操作を実行するには、正しい権限およびロールを持っている必要があります。

## [チュートリアル: ロールと権限の付与 \(SQL の場合\) \[1567 ページ\]](#)

ロールベースのセキュリティモデルでは、ユーザが特定のデータベース操作を実行するには、正しい権限およびロールを持っている必要があります。

## [権限ベースのセキュリティからロールベースのセキュリティへのアップグレード \[1575 ページ\]](#)

データベースを、権限ベースのセキュリティからロールベースのセキュリティにアップグレードすることができます。

## 関連情報

[権限ベースのセキュリティからロールベースのセキュリティへのアップグレード \[1575 ページ\]](#)

## 1.9.1.1    ロール

ロールベースのセキュリティモデルには 3 種類のロールがあります。システムロール、ユーザ定義ロール (ユーザ拡張ロールを含む)、および 互換ロールです。

ユーザをクリックし、表示される詳細を確認して、ユーザが持つ SQL Central のロールと権限を表示します。また、sp\_displayroles システムプロシージャを使用して詳細を取得することもできます。

このセクションの内容:

### [システムロール \[1456 ページ\]](#)

システムロールには、さらに複雑な操作に必要な権限のセットが含まれます。

### [互換ロール \[1459 ページ\]](#)

互換ロールは、権限の論理グループを含む開始用のロールのようなものと考えられます。

### [ユーザ定義ロール \[1470 ページ\]](#)

ユーザ定義ロールは作成可能なシステム権限、オブジェクトレベル権限、およびロールのコレクションであり、主に特定のタスクまたはタスクセットに関連するグループ権限に対して作成されます。

### [ユーザ拡張ロール \[1475 ページ\]](#)

ユーザ拡張ロールはロールに拡張されたユーザであり、ユーザ定義ロールの 1 つのタイプです。

### [ロール管理者 \[1481 ページ\]](#)

ロール管理者は、ユーザおよび他のロールに対してユーザ定義ロールを付与したり取り消したりする役割を持ちます。

### [アップグレード時に付与されるシステム権限 \[1484 ページ\]](#)

ソフトウェアの新しいリリースによって、新しいシステム権限が導入される場合があります。

### [ロールにロールと権限を設定する \(SQL Central の場合\) \[1486 ページ\]](#)

任意のロールにロールや権限を付与したり、ロールや権限を取り消したり、そのロールに付与されたロールや権限に対してロールが持つ管理権限を設定したりします。

### [ロールにロールと権限を設定する \(SQL の場合\) \[1487 ページ\]](#)

任意のロールにロールや権限を付与したり、ロールや権限を取り消したり、そのロールに付与されたロールや権限に対してロールが持つ管理権限を設定したりします。

### [ロールの付与 \(SQL Central の場合\) \[1489 ページ\]](#)

ユーザまたは別のロールにロールを付与します。

### [ロールの付与 \(SQL の場合\) \[1490 ページ\]](#)

ユーザまたは別のロールにロールを付与します。

### [ロールの取り消し \(SQL Central の場合\) \[1491 ページ\]](#)

ユーザ、ユーザ拡張ロール、またはユーザ定義ロールからロールを取り消します。

### [ロールの取り消し \(SQL の場合\) \[1493 ページ\]](#)

ユーザ、ユーザ拡張ロール、またはユーザ定義ロールからロールを取り消します。

### [ロールの削除 \(SQL Central の場合\) \[1494 ページ\]](#)

ユーザ定義、ユーザ拡張、および互換の各ロールを削除します。

### [ロールの削除 \(SQL の場合\) \[1495 ページ\]](#)

ユーザ定義、ユーザ拡張、および互換の各ロールを削除します。

## 関連情報

[権限ベースのセキュリティからロールベースのセキュリティへのアップグレード \[1575 ページ\]](#)

### 1.9.1.1.1 システムロール

システムロールには、さらに複雑な操作に必要な権限のセットが含まれます。

システムロールについて覚えておくべき点がいくつかあります。

- システムロールは削除できません。
- システムロールを付与するときに管理権限 (WITH ADMIN OPTION または WITH ADMIN ONLY OPTION) を付与することはできません。これらのシステムロールでの管理権限は、MANAGE ROLES システム権限とのみ存在します。
- システムロールがユーザ拡張ロールに付与されるとき、ユーザ拡張ロールの被付与者はシステムロールも同様に継承します。
- SYS ロールという例外はありますが、can grant/revoke 追加の権限やロールをシステムロールに追加したり、システムロールから取り消したりできます。ただし、追加や取り消しを実行する権限やロールに対する管理権限がある場合に限りです。
- SYS、dbo、および rs\_systabgroup ロールを除き、システムロールはオブジェクトを所有しません。

システムロール	説明
dbo	このロールは、Ultra Light と Mobile Link 用に使用される多数のシステムストアプロシージャ、ビュー、テーブルを所有しており、SYS_AUTH_DBA_ROLE ロールと SYS_AUTH_RESOURCE_ROLE ロールの被付与者です。このロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。
DIAGNOSTICS	このロールは、SELECT、INSERT、UPDATE、DELETE、ALTER の権限を診断テーブルとビューに付与します。このロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。
PUBLIC	このロールには、システムテーブルに対する SELECT 権限があります。また、PUBLIC ロールは SYS ロールと dbo ロールの被付与者であり、一部のシステムテーブルとビューの読み込みアクセスを保有しているため、ユーザはデータベーススキーマに関する情報を検索できます。
rs_systabgroup	このロールは、Replication Server に必要なテーブルとシステムプロシージャを所有しています。このロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。
SA_DEBUG	このロールは、SQL Anywhere デバッグ用に必要です。このロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。
SYS	このロールは、すべてのデータベースオブジェクトとすべてのユーザ ID をはじめ、データベーススキーマの詳細を含むカタログを所有しています。このロールに追加の権限を付与したり、このロールから追加の権限を取り消したりはできません。このロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。

システムロール	説明
SYS_RUN_PROFILER_ROLE	このロールは、SQL Anywhere プロファイラ デバッグ用に必要です。また、このロールはステートメントパフォーマンスサマリを表示するためにも必要です。このアプリケーションにおける診断やトラブルシューティングに必要な権限を含みます。このロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。
SYS_REPLICATION_ADMIN_ROLE	このロールは、レプリケーションロールの付与、パブリケーションの管理、サブスクリプション、ユーザとプロファイルの同期、メッセージタイプの管理、レプリケーション関連のオプションの設定など、レプリケーションに関連する管理タスクを実行するために必要です。このロールを管理できるのは、MANAGE REPLICATION システム権限を持つユーザのみです。
SYS_RUN_REPLICATION_ROLE	このロールは、dbremote ユーティリティを使用してレプリケーションを実行したり、dbmsync ユーティリティを使用して同期を実行したりするために必要です。このロールを管理できるのは、SYS_REPLICATION_ADMIN_ROLE システムロールを持つユーザのみです。
SYS_SAMONITOR_ADMIN_ROLE	このロールは、SQL Anywhere モニタ用に必要です。
SYS_SPATIAL_ADMIN_ROLE	このロールでは、ユーザは空間参照系と空間測定単位を作成、変更、または削除できます。このロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。

このセクションの内容:

[レプリケーションに関連するシステムロール \[1457 ページ\]](#)

レプリケーションに関連するシステムロールには 2 種類あります。SYS\_RUN\_REPLICATION\_ROLE と SYS\_REPLICATION\_ADMIN\_ROLE です。

### 1.9.1.1.1.1 レプリケーションに関連するシステムロール

レプリケーションに関連するシステムロールには 2 種類あります。SYS\_RUN\_REPLICATION\_ROLE と SYS\_REPLICATION\_ADMIN\_ROLE です。

このセクションの内容:

[SYS\\_REPLICATION\\_ADMIN\\_ROLE システムロール \[1458 ページ\]](#)

SYS\_REPLICATION\_ADMIN\_ROLE システムロールは、レプリケーションロールの付与、パブリケーションの管理、サブスクリプション、ユーザとプロファイルの同期、メッセージタイプの管理、レプリケーション関連のオプションの設定など、レプリケーションに関連する管理タスクを実行するために必要です。

[SYS\\_RUN\\_REPLICATION\\_ROLE システムロール \[1458 ページ\]](#)

SYS\_RUN\_REPLICATION\_ROLE システムロールは、dbremote ユーティリティを使用してレプリケーションを実行したり、dbmsync ユーティリティを使用して同期を実行したりするために必要です。

## 関連情報

[互換ロール \[1459 ページ\]](#)

### 1.9.1.1.1.1 SYS\_REPLICATION\_ADMIN\_ROLE システムロール

SYS\_REPLICATION\_ADMIN\_ROLE システムロールは、レプリケーションロールの付与、パブリケーションの管理、サブスクリプション、ユーザとプロファイルの同期、メッセージタイプの管理、レプリケーション関連のオプションの設定など、レプリケーションに関連する管理タスクを実行するために必要です。

SYS\_REPLICATION\_ADMIN\_ROLE に付与されるシステム権限は次のとおりです。

- MANAGE REPLICATION
- SET ANY SYSTEM OPTION
- SET ANY PUBLIC OPTION
- SET ANY USER DEFINED OPTION
- SELECT ANY TABLE
- CREATE ANY PROCEDURE
- DROP ANY PROCEDURE
- MANAGE ANY WEB SERVICE
- CREATE ANY TABLE
- DROP ANY TABLE
- SERVER OPERATOR
- MANAGE ANY USER
- MANAGE ROLES
- MANAGE ANY OBJECT PRIVILEGE

これらは、この 2 つのロールに必要な最小限の (かつ、取り消すことができない) 権限ですが、追加の権限を付与することはできません。

上記権限に関連付けられた文の他に、SYS\_REPLICATION\_ADMIN\_ROLE システムロールを持つユーザは、次の文を実行できます。

- SYNCHRONIZE PROFILE 文
- REMOTE RESET 文
- LOCK FEATURE 文

### 1.9.1.1.1.2 SYS\_RUN\_REPLICATION\_ROLE システムロール

SYS\_RUN\_REPLICATION\_ROLE システムロールは、dbremote ユーティリティを使用してレプリケーションを実行したり、dbmlsync ユーティリティを使用して同期を実行したりするために必要です。

SYS\_RUN\_REPLICATION\_ROLE に付与されたシステム権限は、次のとおりです。

- MONITOR

- BACKUP DATABASE
- DROP CONNECTION
- SELECT ANY TABLE
- SET ANY SYSTEM OPTION
- SET ANY USER DEFINED OPTION
- ACCESS USER PASSWORD

また、SYS\_RUN\_REPLICATION\_ROLE システムロールは、SYS\_AUTH\_DBA\_ROLE 互換性ロールに付与されたシステム権限を継承します。

上記権限に関連付けられた文の他に、SYS\_RUN\_REPLICATION\_ROLE システムロールを持つユーザは、SYNCHRONIZE 文を実行できます。

Mobile Link では、接続が SQL Anywhere 同期クライアント (dbmlsync) ユーティリティから行われた場合は、SYS\_RUN\_REPLICATION\_ROLE システムロールによって、dbmlsync がデータベースにフルアクセスできるようになります。同じユーザ ID を使用する他の接続には、特別な権限は付与されません。

SQL Remote では、接続が Message Agent から行われた場合は、SYS\_RUN\_REPLICATION\_ROLE システムロールによって、Message Agent がデータベースにフルアクセスしてメッセージの内容を変更できるようになります。同じユーザ ID を使用する他の接続には、特別な権限は付与されません。

その他の方法でデータベースへの接続が行われた場合、ユーザはこのロールに関連付けられているすべての権限を行使できません。たとえば、ユーザは SELECT ANY TABLE システム権限を使用できません。

SYS\_RUN\_REPLICATION\_ROLE システム権限は、ユーザ ID に完全な DBA 権限を付与しなくて済むようにして、DBA のユーザ ID やパスワードの配布に関連するセキュリティ問題を回避します。

たとえば、SYS\_RUN\_REPLICATION\_ROLE システムロールを持つ SQL Remote のユーザ ID は、Message Agent 以外からの接続に対して特別な権限を持ちません。このユーザのユーザ ID とパスワードが公開されていても、セキュリティ上の問題は発生しません。そのデータベースで CONNECT より上のパーミッションが付与されたユーザ ID を使用しないかぎり、データベース内のデータにアクセスすることはできません。

## 1.9.1.1.2 互換ロール

互換ロールは、権限の論理グループを含む開始用のロールのようなものと考えられます。

また、互換ロールは、権限ベースのセキュリティをサポートする SQL Anywhere の以前のバージョンとの下位互換性を確保するためにも存在します。

廃止予定の GRANT および REVOKE 構文を使用すれば、権限を付与および取り消すことができます。ただし、データベースサーバにより、付与や取り消しが相当する互換ロールに変換されます。さらに、16.0 より前のバージョンの継承不可であった権限 (DBA、REMOTE DBA、BACKUP、RESOURCE および VALIDATE) の場合、相当する互換ロールはデフォルトで継承可能です。

互換ロールの基本となるシステム権限は変更できません。ただし、それらをユーザ定義ロールに移行してから、権限を変更することができます。互換ロールを移行すると、代わりに互換ロールの被付与者すべてに自動的にユーザ定義ロールが付与されます。互換ロールをユーザ定義ロールに移行すると、互換ロールが自動的に削除されます。ただし、互換ロールはいつでもリストアできます。

次の表では、各ロールに相当する 16.0 より前のバージョンの権限を含むすべての互換ロールを説明しています。

互換ロール	説明	16.0 より前のバージョンの権限
SYS_AUTH_DBA_ROLE	このロールは、SYS_AUTH_DBA_ROLE および SYS_AUTH_SA_ROLE 互換ロールを付与されていることで、ソフトウェア内のすべての付与可能権限を網羅します。	DBA
SYS_AUTH_SA_ROLE	このロールは、SYS_AUTH_DBA_ROLE 互換性ロールが保有するデータベース管理権限と、そのロールにリンクされているデータベース管理権限のすべてを網羅しています。このロールに SYS_AUTH_SSO_ROLE を加えると、SYS_AUTH_DBA_ROLE 互換性ロールのすべての権限と同等になります。  このロールは移行、削除、リストアできますが、SYS_AUTH_DBA_ROLE に対してそれらの操作を実行する一部としてのみ実行できます。	-
SYS_AUTH_SSO_ROLE	このロールは、SYS_AUTH_DBA_ROLE 互換性ロールが保有するすべてのセキュリティおよびアクセス関連の権限を網羅しています。このロールに SYS_AUTH_SA_ROLE を加えると、SYS_AUTH_DBA_ROLE 互換性ロールのすべての権限と同等になります。  このロールは移行、削除、リストアできますが、SYS_AUTH_DBA_ROLE に対してそれらの操作を実行する一部としてのみ実行できます。	-
SYS_AUTH_BACKUP_ROLE	このロールでは、ユーザは BACKUP DATABASE 文または dbbackup ユーティリティを使用して、データベースとトランザクションログをアーカイブバックアップまたはイメージバックアップへバックアップできます。	BACKUP
SYS_AUTH_PROFILE_ROLE	このロールでは、ユーザはプロファイリング、トレーシング、診断の各操作を実行できます。	PROFILE
SYS_AUTH_READCLIENTFILE_ROLE	このロールでは、ユーザはクライアントコンピュータ上のファイルを読み込みます。たとえば、クライアントコンピュータ上のファイルからデータをロードできます。	READCLIENTFILE
SYS_AUTH_READFILE_ROLE	このロールでは、ユーザは SELECT 文で OPENSTRING 句を使用してファイルを読み込みます。	READFILE

互換ロール	説明	16.0 より前のバージョンの権限
SYS_AUTH_RESOURCE_ROLE	このロールでは、ユーザはテーブル、ビュー、ストアドプロシージャ、トリガなどのデータベースオブジェクトを作成できます。	RESOURCE
SYS_AUTH_VALIDATE_ROLE	このロールでは、ユーザは VALIDATE 文または dbvalid ユーティリティを使用して、データベース、テーブル、インデックス、チェックサムの検証を実行できます。	VALIDATE
SYS_AUTH_WRITEFILE_ROLE	このロールでユーザは、xp_write_file システムプロシージャを実行できます。	WRITEFILE
SYS_AUTH_WRITECLIENTFILE_ROLE	このロールでは、ユーザはクライアントコンピュータ上にファイルを書き込みます。たとえば、UNLOAD TABLE 文を使用してクライアントコンピュータにデータを書き込みます。	WRITECLIENTFILE

このセクションの内容:

[互換ロールをユーザ定義ロールに移行する \(SQL Central の場合\) \[1462 ページ\]](#)

互換ロールをユーザ定義ロールに移行してから、他のロールおよびユーザに新しいロールを付与します。

[互換ロールをユーザ定義ロールに移行する \(SQL の場合\) \[1463 ページ\]](#)

互換ロールをユーザ定義ロールに移行してから、他のロールおよびユーザに新しいロールを付与します。

[互換ロールの付与 \(SQL Central の場合\) \[1465 ページ\]](#)

ユーザまたはロールに互換ロールを付与します。

[互換ロールの付与 \(SQL の場合\) \[1466 ページ\]](#)

ユーザまたはユーザ拡張ロールに互換ロールを付与します。

[互換ロールのリストア \(SQL Central の場合\) \[1468 ページ\]](#)

移行または削除された互換ロールをリストアします。

[互換ロールのリストア \(SQL の場合\) \[1469 ページ\]](#)

移行または削除された互換ロールをリストアします。

## 関連情報

[権限 \(authority\) は互換ロールに \[1577 ページ\]](#)

[ロールとなる一部の権限の継承動作の変更 \[1592 ページ\]](#)

## 1.9.1.1.2.1 互換ロールをユーザ定義ロールに移行する (SQL Central の場合)

互換ロールをユーザ定義ロールに移行してから、他のロールおよびユーザに新しいロールを付与します。

### 前提条件

移行するロールに関して `MANAGE ANY USER` システム権限、`MANAGE ROLES` システム権限、および管理権限が必要です。

### コンテキスト

互換ロールを移行すると、互換ロールの被付与者すべてに自動的にユーザ定義ロールが付与され、互換ロールは削除されます。

互換ロールは開始用のロールのようなものです。互換ロールの基本となるシステム権限は変更できません。ただし、それらをユーザ定義ロールに移行してから、権限を変更することができます。

新しいロールに付ける名前には、プレフィクス `SYS_` を先頭に付けることも、サフィックス `_ROLE` を末尾に付けることもできません。たとえば、`SYS_MyBackup_ROLE` をユーザ定義ロールの名前にすることはできません。新しいロールに付けた名前は永続的です。後で変更することはできません。

`SYS_AUTH_SA_ROLE` および `SYS_AUTH_SSO_ROLE` 互換性ロールは個別に移行することはできません。ただし、`SYS_AUTH_DBA_ROLE` 互換性ロールを移行するとき、`SYS_AUTH_SA_ROLE` および `SYS_AUTH_SSO_ROLE` は同様に新しいユーザ定義ロールに自動的に移行されます (デフォルトでは、`sa_role` および `sso_role`)。

システム権限の継承を無効にしたロール (`SYS_AUTH_VALIDATE_ROLE` など) を移行する場合、新しく作成されたユーザ定義ロールの権限は、継承ツリー内の被付与者すべてに付与されます。

### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、**ロール** をダブルクリックして、互換ロールを選択します。
3. 互換ロールを右クリックし、**[ユーザ定義ロールへ移行]** をクリックします。
4. **名前** フィールドで新しいユーザ定義ロールの名前を新たに指定するか、デフォルトの名前を受け入れます。
5. **OK** をクリックします。

## 結果

互換ロールの被付与者にはすべて、自動的にユーザ定義ロールが付与されます。互換ロールは削除されます。

## 次のステップ

ユーザ定義ロールのシステム権限を変更し、そのロールを他のユーザに付与します。

## 関連情報

[互換ロールをユーザ定義ロールに移行する \(SQL の場合\) \[1463 ページ\]](#)

[互換ロールのリストア \(SQL の場合\) \[1469 ページ\]](#)

[ロールにロールと権限を設定する \(SQL Central の場合\) \[1486 ページ\]](#)

### 1.9.1.1.2.2 互換ロールをユーザ定義ロールに移行する (SQL の場合)

互換ロールをユーザ定義ロールに移行してから、他のロールおよびユーザに新しいロールを付与します。

## 前提条件

移行するロールに対する MANAGE ROLES システム権限および管理権限が必要です。

## コンテキスト

互換ロールを移行すると、互換ロールの被付与者すべてに自動的にユーザ定義ロールが付与され、互換ロールは削除されます。

互換ロールは開始用のロールのようなものです。互換ロールの基本となるシステム権限は変更できません。ただし、それらをユーザ定義ロールに移行してから、権限を変更することができます。

新しいロールに付ける名前には、プレフィクス SYS\_ を先頭に付けることも、サフィックス \_ROLE を末尾に付けることもできません。たとえば、SYS\_MyBackup\_ROLE をユーザ定義ロールの名前にすることはできません。新しいロールに付けた名前は永続的です。後で変更することはできません。

システム権限の継承を無効にしたロール (SYS\_AUTH\_VALIDATE\_ROLE など) を移行する場合、新しく作成されたユーザ定義ロールの権限は、継承ツリー内の被付与者すべてに付与されます。

SYS\_AUTH\_SA\_ROLE および SYS\_AUTH\_SSO\_ROLE 互換性ロールは個別に移行することはできません。ただし、SYS\_AUTH\_DBA\_ROLE 互換性ロールを移行するとき、SYS\_AUTH\_SA\_ROLE および SYS\_AUTH\_SSO\_ROLE は、自動的に新しいユーザ定義ロールに含まれます。

## 手順

互換ロールを移行するには、次のような ALTER ROLE 文を実行します。

```
ALTER ROLE compatibility-role MIGRATE TO new-role-name;
```

SYS\_AUTH\_DBA\_ROLE 互換性ロールを移行する場合、自動的に移行される SYS\_AUTH\_SA\_ROLE および SYS\_AUTH\_SSO\_ROLE 互換性ロールの名前 (つまり、MIGRATE TO new-role-name, new-sa-role-name, new-sso-role-name) を忘れずに含めるようにします。

## 結果

互換ロールの被付与者にはすべて、自動的にユーザ定義ロールが付与されます。互換ロールは削除されます。

### 例

次の文は、SYS\_AUTH\_BACKUP\_ROLE 互換ロールを新しいユーザ定義ロールである my\_BACKUP に移行します。以前 SYS\_AUTH\_BACKUP\_ROLE 互換ロールを付与されたすべてのユーザには、互換ロールに対するあらゆる管理権限をはじめ、my\_BACKUP ロールが付与されます。

```
ALTER ROLE SYS_AUTH_BACKUP_ROLE MIGRATE TO my_BACKUP;
```

次の文は、SYS\_AUTH\_DBA\_ROLE、SYS\_AUTH\_SA\_ROLE、および SYS\_AUTH\_SSO\_ROLE 互換性ロールを、それぞれ my\_DBA、my\_DBA\_Administration および my\_DBA\_Security というユーザ定義ロールに移行します。すべてのユーザ、基本となるシステム権限、および元のロールに付与されたロールは、自動的に新しいロールに移行されます。最後に、SYS\_AUTH\_DBA\_ROLE、SYS\_AUTH\_SA\_ROLE、および SYS\_AUTH\_SSO\_ROLE 互換性ロールが削除されます。

```
ALTER ROLE SYS_AUTH_DBA_ROLE  
MIGRATE TO my_DBA, my_DBA_Administration, my_DBA_Security
```

## 次のステップ

新しいユーザ定義ロールのシステム権限を変更し、そのロールを他のユーザに付与します。

互換ロールは移行された後自動的に削除されますが、CREATE ROLE 文を使用すれば、後から使用する目的でリストアできます。

## 関連情報

[互換ロールをユーザ定義ロールに移行する \(SQL Central の場合\) \[1462 ページ\]](#)

[互換ロールのリストア \(SQL の場合\) \[1469 ページ\]](#)

### 1.9.1.1.2.3 互換ロールの付与 (SQL Central の場合)

ユーザまたはロールに互換ロールを付与します。

#### 前提条件

付与している互換ロールに対する管理権限が必要です。

#### コンテキスト

ロールに互換ロールが付与されている場合は、そのロールとその被付与者の両方が、その互換ロールの権限を使用できません。ただし、次の互換ロールに対するシステム権限の継承は無効にすることができます。

- SYS\_AUTH\_DBA\_ROLE
- SYS\_AUTH\_RESOURCE\_ROLE
- SYS\_AUTH\_BACKUP\_ROLE
- SYS\_AUTH\_VALIDATE\_ROLE

SQL Central で、これらの互換ロールのいずれかをユーザ拡張ロールまたはシステムロールに付与するとき、デフォルトでシステム権限の継承が無効になっています。互換ロールに対するシステム権限の継承を無効にすると、バージョン 12 以前のデータベースの継承不可能な権限の動作と同様になります。

#### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. **ロール**フォルダで付与したい互換ロールをクリックします。
3. 右ウィンドウ枠で、**被付与者**をクリックします。
4. 右ウィンドウ枠の任意の場所を右クリックして、**新規** > **被付与者** を選択します。
5. 互換ロールを付与するユーザまたはロールをクリックし、**[OK]** をクリックします。

新しいローが **[被付与者]** に追加されます。

互換ロールのシステム権限の継承が無効になっている場合、**オプション** カラムには **継承が無効** が含まれています。システム権限の継承を有効にするには、ローを右クリックして **[オプションの設定]** をクリックし、**[システム権限の継承を有効にする]** を選択します。

SYS\_AUTH\_DBA\_ROLE 互換ロールの付与を除き、**[継承が無効]** が選択されていると管理権限を付与できません。

6. **▶ ファイル ▶ 保存 ▶** をクリックします。

## 結果

互換ロールは、ユーザまたはロールに付与されます。

## 関連情報

[権限 \(authority\) は互換ロールに \[1577 ページ\]](#)

[互換ロールをユーザ定義ロールに移行する \(SQL Central の場合\) \[1462 ページ\]](#)

## 1.9.1.1.2.4 互換ロールの付与 (SQL の場合)

ユーザまたはユーザ拡張ロールに互換ロールを付与します。

## 前提条件

付与しているロールに対する管理権限が必要です。

## コンテキスト

ユーザ拡張ロールではないユーザ定義ロールには、互換ロールを付与できません。

ロールに互換ロールが付与されている場合は、そのロールとその被付与者の両方が、その互換ロールの権限を使用できません。ただし、次の互換ロールに対するシステム権限の継承は無効にすることができます。

- SYS\_AUTH\_DBA\_ROLE
- SYS\_AUTH\_RESOURCE\_ROLE
- SYS\_AUTH\_BACKUP\_ROLE
- SYS\_AUTH\_VALIDATE\_ROLE

互換ロールのシステム権限の継承を無効にした場合は、被付与者でなく、ロールのみがシステム権限を使用できます。互換ロールに対するシステム権限の継承を無効にすると、バージョン 12 以前のデータベースの継承不可能な権限の動作と同様になります。

## 手順

GRANT ROLE 文を実行します

オプション	アクション
他のユーザまたはロールにロールを付与したり、ロールを取り消したりする機能とともに、ロールを行使する機能を付与します。	<pre>GRANT ROLE SYS_AUTH_DBA_ROLE TO user-or-role-name WITH ADMIN OPTION;</pre>
他のユーザまたはロールにロールを付与したり、ロールを取り消したりする機能を付与しますが、ロールは行使しません。	<pre>GRANT ROLE SYS_AUTH_DBA_ROLE TO user-or-role-name WITH ADMIN ONLY OPTION;</pre>
ロールを行使する機能を付与しますが、他のユーザまたはロールにロールを付与したり、ロールを取り消したりはしません。	<pre>GRANT ROLE SYS_AUTH_DBA_ROLE TO user-or-role-name WITH NO ADMIN OPTION;</pre>
互換ロールを行使する機能を付与し、ロールのシステム権限継承を無効にします。	<pre>GRANT ROLE SYS_AUTH_DBA_ROLE TO role-name WITH NO SYSTEM PRIVILEGE INHERITANCE;</pre>

## 結果

互換ロールは、ユーザまたはロールに付与されます。

### 例

この例では、SYS\_AUTH\_BACKUP\_ROLE をロール R\_HumanResources に付与します。

```
GRANT ROLE SYS_AUTH_BACKUP_ROLE TO R_HumanResources
WITH NO SYSTEM PRIVILEGE INHERITANCE
```

## 関連情報

[権限 \(authority\) は互換ロールに \[1577 ページ\]](#)

[互換ロールをユーザ定義ロールに移行する \(SQL の場合\) \[1463 ページ\]](#)

## 1.9.1.1.2.5 互換ロールのリストア (SQL Central の場合)

移行または削除された互換ロールをリストアします。

### 前提条件

MANAGE ROLES システム権限、およびリストアしているロールに付与されたすべてのシステム権限に対する管理権限が必要です。たとえば、SYS\_AUTH\_BACKUP\_ROLE 互換ロールをリストアするには、MANAGE ROLES システム権限、および BACKUP DATABASE システム権限に対する管理権限が必要です。

### コンテキスト

データベース内に存在しなくなった互換ロールをリストアできます。これらのロールのいずれかをリストアすると、元のロールが持っていた権限を思い出したり、それらの権限をユーザ定義ロールに移行したりする場合に役立ちます。

互換ロールを移行した場合 (互換ロールは削除されます)、そのロールの被付与者には自動的に、移行先の新しいユーザ定義ロールが付与されます。互換ロールをリストアしても、被付与者が互換性ロールにリストアされるわけではありません。作成したユーザ定義ロールの被付与者のままとなります。

### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、**[ロール]** をダブルクリックします。
3. 右ウィンドウ枠の任意の場所で右クリックし、**新規 > ロール** をクリックします。
4. **ロール作成ウィザード**の **[1 つ以上の互換ロールのリストア]** をクリックし、表示される指示に従います。

### 結果

ロールがリストアされます。リストアされたロールに対する管理者権限は、まだデータベース内にある場合は、自動的に SYS\_AUTH\_DBA\_ROLE 互換性ロールに付与されます。

SYS\_AUTH\_DBA\_ROLE 互換性ロールをリストアすると、SYS\_AUTH\_SA\_ROLE および SYS\_AUTH\_SSO\_ROLE 互換性ロールも自動的にリストアされます。

### 関連情報

[互換ロールをユーザ定義ロールに移行する \(SQL Central の場合\) \[1462 ページ\]](#)

## 1.9.1.1.2.6 互換ロールのリストア (SQL の場合)

移行または削除された互換ロールをリストアします。

### 前提条件

MANAGE ROLES システム権限、およびリストアしているロールに付与されたすべてのシステム権限に対する管理権限が必要です。たとえば、SYS\_AUTH\_BACKUP\_ROLE 互換ロールをリストアするには、MANAGE ROLES システム権限、および BACKUP DATABASE システム権限に対する管理権限が必要です。

### コンテキスト

データベース内に存在しなくなった互換ロールをリストアできます。これらのロールのいずれかをリストアすると、元のロールが持っていた権限を思い出したり、それらの権限をユーザ定義ロールに移行したりする場合に役立ちます。

互換ロールを移行した場合 (互換ロールは削除されます)、そのロールの被付与者には自動的に、移行先の新しいユーザ定義ロールが付与されます。互換ロールをリストアしても、被付与者が互換性ロールにリストアされるわけではありません。作成したユーザ定義ロールの被付与者のままとなります。

### 手順

次のような CREATE ROLE 文を実行します。ここで `role-name` はロールの名前です。

```
CREATE ROLE role-name;
```

### 結果

ロールがリストアされます。リストアされたロールに対する管理者権限は、まだデータベース内にある場合は、自動的に SYS\_AUTH\_DBA\_ROLE 互換性ロールに付与されます。

SYS\_AUTH\_DBA\_ROLE 互換性ロールをリストアすると、SYS\_AUTH\_SA\_ROLE および SYS\_AUTH\_SSO\_ROLE 互換性ロールも自動的にリストアされます。

## 例

次の例では、SYS\_AUTH\_BACKUP\_ROLE ロールをリストアし、ユーザ TomW に他のユーザまたはロールにロールを付与したり、ロールを取り消したりする機能を付与しますが、ロールは行使しません。

```
GRANT ROLE SYS_AUTH_BACKUP_ROLE TO TomW WITH ADMIN ONLY OPTION;
```

次の例では、SYS\_AUTH\_DBA\_ROLE、SYS\_AUTH\_SA\_ROLE、および SYS\_AUTH\_SSO\_ROLE 互換性ロールをリストアします。

```
CREATE ROLE SYS_AUTH_DBA_ROLE
```

## 関連情報

[互換ロールをユーザ定義ロールに移行する \(SQL の場合\) \[1463 ページ\]](#)

### 1.9.1.1.3 ユーザ定義ロール

ユーザ定義ロールは作成可能なシステム権限、オブジェクトレベル権限、およびロールのコレクションであり、主に特定のタスクまたはタスクセットに関連するグループ権限に対して作成されます。

ユーザ定義ロールは組織のニーズに合わせて作成できます。

ユーザ定義ロールはオブジェクトを所有できます。

ユーザ定義ロールが管理権限とともに付与されるとき、ユーザはロールの基本となる任意のシステム権限を使用できるだけでなく、ロールを管理 (付与、取り消し、削除) できます。管理権限のみが付与された場合、ユーザはロールを管理できますが、その基本となるシステム権限を使用することはできません。管理権限が付与されない場合、ユーザは基本となるシステム権限を使用することだけできます。

ユーザ定義ロールの特別な形式の 1 つに、ユーザ拡張ロールがあります。これは、ロールになるように拡張されたユーザで、他のユーザに付与することができます。

このセクションの内容:

[COCKPIT\\_ROLE ユーザ定義ロール \[1471 ページ\]](#)

COCKPIT\_ROLE はコックピットを使用するときに必要なユーザ定義ロールです。

[ユーザ定義ロールの作成 \(SQL Central の場合\) \[1472 ページ\]](#)

ロールを作成し、そこにロールおよび権限を付与します。

[ユーザ定義ロールの作成 \(SQL の場合\) \[1474 ページ\]](#)

ロールを作成し、そこに権限およびロールを付与します。

## 関連情報

[ユーザ拡張ロール \[1475 ページ\]](#)

[互換ロールをユーザ定義ロールに移行する \(SQL の場合\) \[1463 ページ\]](#)

### 1.9.1.1.3.1 COCKPIT\_ROLE ユーザ定義ロール

COCKPIT\_ROLE はコックピットを使用するときに必要なユーザ定義ロールです。

#### デフォルト

COCKPIT\_ROLE は、データベースの作成または更新時に作成されるユーザ定義ロールです。デフォルトでは、このロールの管理権限は MANAGE ROLES システムロールに付与されます。

このロールの行使権限は、ユーザに明示的に付与する必要があります。新規に作成されたデータベースでは、COCKPIT\_ROLE の行使権限は初期ユーザを含めどのユーザにも付与されません。バージョン 16 のデータベースには COCKPIT\_ROLE が存在しないため、作成してユーザに付与する必要があります。

利便性のために、COCKPIT\_ROLE には以下のシステム権限が格納されています。COCKPIT\_ROLE と、これらのシステム権限を持つユーザは、コックピットの機能のすべてにアクセスすることができます。

- MONITOR システム権限
- DROP CONNECTION システム権限
- BACKUP DATABASE システム権限
- SERVER OPERATOR システム権限
- ACCESS DISK INFORMATION システム権限

#### 備考

COCKPIT\_ROLE ユーザ定義ロールはコックピットへの接続に必要です。このロールはデータベースサーバレベルのロールです。

データベースに COCKPIT\_ROLE がなく、コックピットを使用する必要がある場合は、作成してユーザに付与します。

ユーザ定義ロールであるため、必要であればデータベースから削除して、後から再度作成することができます。たとえば、すべてのデータベースユーザがコックピットに接続できないようにするには、COCKPIT\_ROLE を削除してください。

コックピットにすべてのデフォルトの権限を付与または取り消すことができます。デフォルトの権限を取り消すと、ユーザはコックピットに接続することはできるものの、取り消した権限が必要な任意のアクションは無効になります。たとえば、COCKPIT\_ROLE のみを持つユーザがいるとします。COCKPIT\_ROLE で BACKUP DATABASE システム権限を取り消すと、ユーザはコックピットまたは他のツールやユーティリティを使用してデータベースのバックアップを行うことができなくなります。

コックピットはデータベースサーバ上で稼働し、当該データベースサーバ上で実行中のすべてのデータベースの情報を表示します。データベースサーバ上で実行中のデータベースの COCKPIT\_ROLE を持つすべてのデータベースユーザは、コックピットを使用することができます。ただし、バックアップなどのアクションをデータベース上で行うには、そのデータベースのユーザにクレデンシャルを与える必要があります。たとえば、同一のサーバ上で2つのデータベース、データベース A とデータベース B が実行中だとします。この場合、デフォルト権限を含む COCKPIT\_ROLE を持つデータベース A のユーザのクレデンシャルを使用して、コックピットに接続します。データベース A をバックアップする権限はありますが、データベース B をバックアップするには、データベースのバックアップに必要なシステム権限を持つデータベース B のユーザに、クレデンシャルを与える必要があります。

#### 例

COCKPIT\_ROLE を再作成するには、次の SQL ステートメントを実行します。

```
CREATE ROLE COCKPIT_ROLE;  
GRANT MONITOR, DROP_CONNECTION, BACKUP DATABASE, SERVER OPERATOR, ACCESS DISK  
INFORMATION TO COCKPIT_ROLE;
```

たとえば、COCKPIT\_ROLE をユーザ JohnDoe に付与するには、次の文を実行します。

```
GRANT ROLE COCKPIT_ROLE TO JohnDoe;
```

## 関連情報

[SQL Anywhere コックピット \[1349 ページ\]](#)

[コックピットのセキュリティ \[1359 ページ\]](#)

[コックピットの起動とコックピットへの接続 \(sa\\_server\\_option システムプロシージャ\) \[1353 ページ\]](#)

[コックピットの起動とコックピットへの接続 \(SQL Central\) \[1351 ページ\]](#)

### 1.9.1.1.3.2 ユーザ定義ロールの作成 (SQL Central の場合)

ロールを作成し、そこにロールおよび権限を付与します。

#### 前提条件

MANAGE ROLES システム権限が必要です。

新しいロールに付ける名前には、プレフィクス SYS\_ を先頭に付けることも、サフィックス \_ROLE を末尾に付けることもできません。たとえば、SYS\_MyBackup\_ROLE をユーザ定義ロールの名前にすることはできません。

## コンテキスト

また、既存のユーザをユーザ拡張ロールに変換してから、他のユーザにユーザ拡張ロールを付与することもできます。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で **ロール** を右クリックし、**▶ 新規 ▶ ロール ▶** をクリックします。
3. ウィザードの指示に従います。

## 結果

新しいロールが作成されます。デフォルトでは、作成者がロールの所有者になっていて、そのロールは `MANAGE ROLES` システム権限に付与されます。

## 次のステップ

このロールにシステム権限を付与し、このロールを他のユーザに付与します。

## 関連情報

[ロール](#) [1455 ページ]

[ロール管理者](#) [1481 ページ]

[ロールと権限の継承](#) [1537 ページ]

[ロールの付与 \(SQL Central の場合\)](#) [1489 ページ]

[ユーザをユーザ拡張ロールに変換 \(SQL Central の場合\)](#) [1476 ページ]

### 1.9.1.1.3.3 ユーザ定義ロールの作成 (SQL の場合)

ロールを作成し、そこに権限およびロールを付与します。

#### 前提条件

MANAGE ROLES システム権限が必要です。

新しいロールに付ける名前には、プレフィクス SYS\_ を先頭に付けることも、サフィックス \_ROLE を末尾に付けることもできません。たとえば、SYS\_MyBackup\_ROLE をユーザ定義ロールの名前にすることはできません。

#### コンテキスト

また、既存のユーザをユーザ拡張ロールに変換してから、他のユーザにユーザ拡張ロールを付与することもできます。

#### 手順

1. データベースに接続します。
2. CREATE ROLE 文を実行します。次に例を示します。

```
CREATE ROLE role-name
```

3. システム権限とロールを新しいロールに付与します。

#### 結果

新しいロールが作成されます。

#### 例

次の文は Sales という名前のロールを作成します。管理者が指定されていないため、グローバル管理者 (MANAGE ROLES システム権限を持つユーザ) がロールを管理できます。

```
CREATE ROLE Sales;  
GRANT SELECT, UPDATE ON GROUPO.SalesOrders TO Sales;  
GRANT SELECT, UPDATE ON GROUPO.SalesOrderItems TO Sales;
```

## 次のステップ

このロールにシステム権限を付与し、このロールを他のユーザに付与します。

## 関連情報

[ロール \[1455 ページ\]](#)

[ロール管理者 \[1481 ページ\]](#)

[ロールと権限の継承 \[1537 ページ\]](#)

[ユーザ定義ロールの作成 \(SQL Central の場合\) \[1472 ページ\]](#)

[ユーザをユーザ拡張ロールに変換 \(SQL Central の場合\) \[1476 ページ\]](#)

[ロールの付与 \(SQL Central の場合\) \[1489 ページ\]](#)

### 1.9.1.1.4 ユーザ拡張ロール

ユーザ拡張ロールはロールに拡張されたユーザであり、ユーザ定義ロールの 1 つのタイプです。

他のロールに付与可能なロールとして動作するように拡張されたユーザ ID です。ユーザ拡張ロールは、16.0 より前のリリースの SQL Anywhere におけるグループと同等です。

ユーザ拡張ロールをユーザまたは別のロールに付与すると、被付与者は、すべての管理権限を含め、ユーザ拡張ロールが持つシステム権限とオブジェクトレベル権限をすべて継承します。

データベースオブジェクトの所有権が単一のユーザ ID に関連付けられているため、所有者がユーザ拡張ロールであるとき、データベースオブジェクトの所有権はその被付与者には継承されません。付与された権限のみを継承します。

ユーザ拡張ロールは、あるユーザが持つシステム権限のセットを別のユーザにも付与したい場合に便利です。ユーザ拡張ロールになったユーザは、新しいロールを管理 (他のユーザへの付与および他のユーザからの取り消し) できます。ただし、この権限が明示的に削除されている場合を除きます。

ユーザ拡張ロールを作成するとき、MANAGE ROLES システム権限が、管理権限のみを持つロールに自動的に付与されません。

ロールの拡張を取り消したり、被付与者から取り消す権限を制御したりできます。

ユーザは、ユーザ拡張ロールであるとき、そのユーザ ID を使用して引き続きログインできます。

ユーザをユーザ拡張ロールに変換するには、CREATE ROLE 文を使用します。ユーザ拡張ロールをユーザに戻すには、DROP ROLE 文を使用します。

#### 例

- 次の文は user1 を作成し、BACKUP DATABASE 権限を user1 に付与し、user1 をロールとして拡張し、さらに user1 ロールを user2 と user3 に付与します。

```
CREATE USER "user1" IDENTIFIED BY 'sql';
GRANT BACKUP DATABASE TO "user1";
CREATE ROLE FOR USER "user1";
```

```
GRANT ROLE "user1" TO "user2";
GRANT ROLE "user1" TO "user3";
```

次の文は user1 ロールを user2 から取り消します。

```
REVOKE ROLE "user1" FROM "user2";
```

次の文は、user1 ロールをユーザに変更します。結果として、user1 ロールは user3 から取り消されます。

```
DROP ROLE FROM USER "user1" WITH REVOKE;
```

このセクションの内容:

[ユーザをユーザ拡張ロールに変換 \(SQL Central の場合\) \[1476 ページ\]](#)

ユーザをユーザ拡張ロールに変更します。このユーザ拡張ロールは、他のユーザやロールに付与できます。

[ユーザをユーザ拡張ロールに変換する \(SQL の場合\) \[1477 ページ\]](#)

ユーザをユーザ拡張ロールに変更します。このユーザ拡張ロールは、他のユーザやロールに付与できます。

[ユーザ拡張ロールからユーザへの逆変換 \(SQL Central の場合\) \[1479 ページ\]](#)

ユーザ拡張ロールを変更して、通常のユーザに戻します。

[ユーザ拡張ロールをユーザに戻す \(SQL の場合\) \[1480 ページ\]](#)

ユーザ拡張ロールを変更して、通常のユーザに戻します。

## 1.9.1.1.4.1 ユーザをユーザ拡張ロールに変換 (SQL Central の場合)

ユーザをユーザ拡張ロールに変更します。このユーザ拡張ロールは、他のユーザやロールに付与できます。

### 前提条件

MANAGE ROLES 権限が必要です。

### コンテキスト

ユーザ拡張ロールは、特定のユーザが持つシステム権限のセットを別のユーザにも付与したい場合に便利です。

ユーザ拡張ロールになったユーザは、新しいロールを管理 (他のユーザへの付与および他のユーザからの取り消し) できます。ただし、この権限が明示的に削除されている場合を除きます。また、MANAGE ROLES システム権限を持つユーザなら誰でもこのロールを管理できます。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[ユーザ](#) をダブルクリックして、ユーザを選択します。
3. ユーザを右クリックして、[\[ユーザ拡張ロールに変更\]](#) をクリックします。

## 結果

ユーザはロールへと拡張されます。[被付与者](#) タブが右ウィンドウ枠に表示され、左ウィンドウ枠のロールのリストには新しいロールが表示されます。

## 次のステップ

新しいロールは、他のユーザやロールに付与できます。ユーザ拡張ロールをユーザまたは別のロールに付与すると、被付与者は、すべての管理権限を含め、ユーザ拡張ロールが持つシステム権限とオブジェクトレベル権限をすべて継承します。

## 関連情報

[ユーザ拡張ロールからユーザへの逆変換 \(SQL Central の場合\)](#) [1479 ページ]

[ユーザをユーザ拡張ロールに変換する \(SQL の場合\)](#) [1477 ページ]

### 1.9.1.1.4.2 ユーザをユーザ拡張ロールに変換する (SQL の場合)

ユーザをユーザ拡張ロールに変更します。このユーザ拡張ロールは、他のユーザやロールに付与できます。

## 前提条件

MANAGE ROLES システム権限が必要です。

## コンテキスト

ユーザ拡張ロールは、特定のユーザが持つシステム権限のセットを別のユーザにも付与したい場合に便利です。

変換時には、ロールの管理者を指定できます。管理者を指定しない場合、デフォルトでは MANAGE ROLE システム権限を持つユーザなら誰でもロールを管理できます。

ユーザ拡張ロールに拡張されるユーザを管理者として指定することはできず、変換の完了後にロールに対する管理権限を付与することもできません。ただし、拡張されるユーザが MANAGE ROLES システム権限を持ち、変換時に管理者が指定されていない場合は、そのユーザがロールを管理できます。

## 手順

1. データベースに接続します。
2. 次のような CREATE ROLE 文を実行します。

オプション	アクション
MANAGE ROLES システム権限を持つユーザなら誰でも管理できるロールにユーザを拡張します。	<pre>CREATE ROLE FOR USER userid;</pre>
<code>list-of-administrator-ids</code> のユーザなら誰でも管理できるロールにユーザを拡張します。	<pre>CREATE ROLE FOR USER userid WITH ADMIN list-of- administrator-ids;</pre>

## 結果

ユーザはロールへと拡張されます。

### 例

次の文は、ユーザ JaneSmith を拡張して、別のユーザに割り当てることができるロールにします。

```
CREATE ROLE FOR USER JaneSmith;
```

## 次のステップ

新しいロールは、他のユーザやロールに付与できます。ユーザ拡張ロールをユーザまたは別のロールに付与すると、被付与者は、すべての管理権限を含め、ユーザ拡張ロールが持つシステム権限とオブジェクトレベル権限をすべて継承します。

## 関連情報

[ユーザ拡張ロールをユーザに戻す \(SQL の場合\) \[1480 ページ\]](#)

[ユーザをユーザ拡張ロールに変換 \(SQL Central の場合\) \[1476 ページ\]](#)

### 1.9.1.1.4.3 ユーザ拡張ロールからユーザへの逆変換 (SQL Central の場合)

ユーザ拡張ロールを変更して、通常のユーザに戻します。

#### 前提条件

MANAGE ROLES システム権限、または、ロールに対する管理権限が必要です。

#### コンテキスト

ユーザ拡張ロールは、変換して、通常のユーザに戻すことができます。ただし、依存するすべてのロールが、min\_role\_admin データベースオプションで設定された、アクティブなパスワードを持つ管理者ユーザの最小数の要件を満たす必要があります。

#### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[\[ロール\]](#) をダブルクリックして、ユーザ拡張ロールを選択します。
3. ユーザを右クリックしてから、[\[ユーザに変更\]](#) をクリックします。

#### 結果

ロールがユーザに再変換されます。被付与者タブが右ウィンドウ枠に表示されなくなり、左ウィンドウ枠のロールのリストからそのロールが消えます。ユーザ拡張ロールが所有していたオブジェクトはいずれも、変換されたユーザに残ります。以前にユーザ拡張ロールを持っていたユーザまたはロールにはいずれも、変換されたユーザが持っている権限がなくなります。

#### 次のステップ

ユーザ拡張ロールをユーザに変更してロールを削除する場合、そのユーザを削除できます。

## 関連情報

[min\\_role\\_admins オプション \[737 ページ\]](#)

### 1.9.1.1.4.4 ユーザ拡張ロールをユーザに戻す (SQL の場合)

ユーザ拡張ロールを変更して、通常のユーザに戻します。

#### 前提条件

MANAGE ROLES システム権限、または、ロールに対する管理権限が必要です。

#### コンテキスト

ユーザ拡張ロールは、変換して、通常のユーザに戻すことができます。ただし、依存するすべてのロールが、min\_role\_admin データベースオプションで設定された、アクティブなパスワードを持つ管理者ユーザの最小数の要件を満たす必要があります。

#### 手順

1. データベースに接続します。
2. 次のような文を実行します。

オプション	文
ロールが任意のロールまたはユーザに付与されていない場合、ロールをユーザに戻します	<pre>DROP ROLE FROM USER userid;</pre>
ロールをユーザに戻し、基本となる権限をユーザ拡張ロールが付与されたユーザから取り消します	<pre>DROP ROLE FROM USER userid WITH REVOKE;</pre>

## 結果

ロールがユーザに再変換されます。ユーザ拡張ロールが所有していたオブジェクトはいずれも、変換されたユーザに残りません。以前にユーザ拡張ロールを持っていたユーザまたはロールにはいずれも、変換されたユーザが持っている権限がなくなります。

### 例

次の文は、Joe という名前のユーザ拡張ロールを通常のユーザに戻します。ユーザ拡張ロールが所有するオブジェクトは、通常のユーザ、Joe の所有となります。Joe を付与されたユーザまたはロールは、このロールに関連付けられた、基本となる権限を保持します。

```
DROP ROLE FROM USER Joe;
```

次の文は、Sam という名前のユーザ拡張ロールを通常のユーザに戻します。Sam を付与されたユーザおよびロールによって、Jack の権限が取り消されます。

```
DROP ROLE FROM USER Sam WITH REVOKE;
```

次の文は、drops a role named Marketing2 という名前のロールを削除し、このロールが所有していたオブジェクトを削除して、このロールを付与されていたユーザの、ロールの基本となるシステム権限を取り消します。

```
DROP ROLE Marketing2 WITH REVOKE WITH DROP OBJECTS;
```

## 次のステップ

ユーザ拡張ロールをユーザに変更してそのユーザを削除する場合、そのユーザを削除できるようになります。

## 関連情報

[ユーザ拡張ロールからユーザへの逆変換 \(SQL Central の場合\) \[1479 ページ\]](#)

[ユーザをユーザ拡張ロールに変換する \(SQL の場合\) \[1477 ページ\]](#)

[min\\_role\\_admins オプション \[737 ページ\]](#)

### 1.9.1.1.5 ロール管理者

ロール管理者は、ユーザおよび他のロールに対してユーザ定義ロールを付与したり取り消したりする役割を持ちます。

ロール管理者は、必要に応じて追加および削除できます。単一のロールを付与できるロール管理者の数に制限はありません。ただし、min\_role\_admins データベースオプションで指定されるとおり、最少数には制限があります。

また、MANAGE ROLES システム権限を持つユーザであるグローバルロール管理者もいます。グローバルロール管理者は、管理者が指定されていないユーザ定義ロールを管理できます。

グローバルロール管理者は、システムロールの唯一の管理者です。

## 管理者の指定

ロール作成時、またはロールの作成後にロール管理者を指定できます。

この例では、myRole というロールが作成され、MaryM と TomS が管理権限を持つロールに付与されます。作成後、MaryM と TomS だけがロールの付与または取り消しを行えます。

```
CREATE ROLE myRole WITH ADMIN MaryM, TomS;
```

この例では、myRole というロールが作成され、そこに権限が追加され、さらに PeriW というユーザ ID が管理者として (のみ) 追加されます。

```
CREATE USER PeriW IDENTIFIED BY PeriW;  
CREATE ROLE myRole;  
GRANT SELECT ANY TABLE to myRole;  
GRANT ROLE myRole to PeriW WITH ADMIN ONLY OPTION;
```

## グローバル管理者と MANAGE ROLES システム権限

管理者を指定せずにロールを作成する場合、MANAGE ROLES システム権限は、新しいロールに管理権限 (のみ) が自動的に付与されます。このロールは、MANAGE ROLES システム権限を持つユーザなら誰でも管理できます。

MANAGE ROLES は付与されたり (WITH ADMIN ONLY OPTION) 取り消したりするロールを持つ唯一のシステム権限であるという点でユニークです。

ロール作成時に管理者が指定されない場合 (自動的に MANAGE ROLES に付与されることになり) 後でそのロールに管理者が指定され、さらに MANAGE ROLES システム権限を持つユーザはロールを管理できます。

ロールを作成し、作成時に管理者を指定する場合、その管理者だけがロールを管理できます。指定した管理者の 1 人が MANAGE ROLES でなければ、グローバル管理者はそのロールを管理できません。ロールを作成し、管理者の 1 人として MANAGE ROLES を含めるには、次のような文を実行します。MANAGE ROLES システム権限を指定するとき、SYS\_MANAGE\_ROLES\_ROLE という内部表現を使用する必要があります。

MANAGE ROLES システム権限にロールを付与するときには、管理権限 (WITH ADMIN ONLY) で付与する必要があります。それ以外の場合、エラーが返されます。

```
CREATE ROLE myRole WITH ADMIN ONLY MaryM, TomS, SYS_MANAGE_ROLES_ROLE;
```

MaryM、TomS、および MANAGE ROLES システム権限を持つ任意のユーザなら、このロールを管理できます。ただし、MaryM と TomS はロールを行使できません。

### i 注記

ロールを作成するとき、管理者に関連するオプションは次の通りです。WITH ADMIN、および WITH ADMIN ONLY。ロールを付与するとき、管理者に関連するオプションは次の通りです。WITH ADMIN OPTION、WITH NO ADMIN OPTION、および WITH ADMIN ONLY OPTION。

また、ロールを作成した後に次のような文を実行し、MANAGE ROLES システム権限に管理者権限を付与することもできます。必ず WITH ADMIN ONLY OPTION を指定する必要があります。

```
GRANT ROLE myRole TO SYS_MANAGE_ROLES_ROLE WITH ADMIN ONLY OPTION;
```

ロールを作成し、グローバル管理者の管理機能を削除したい場合、次のような文を実行してロールから MANAGE ROLES を取り消します。

```
REVOKE ROLE myRole FROM SYS_MANAGE_ROLES_ROLE;
```

## 作成後にロール管理者を変更または削除する

すでに管理者が割り当てられているロールを持ち、その管理者を別のユーザ ID に変更したい場合、ロールを新しい管理者に付与 (WITH ADMIN または WITH ADMIN ONLY を指定) してから最初の管理者から管理者権限を取り消します。

### i 注記

ロールに管理者のロールを追加したいだけの場合は、CREATE OR REPLACE ROLE 文ではなく、WITH ADMIN [ONLY] OPTION 句とともに GRANT ROLE 文を使用します。

min\_role\_admins データベースオプションは、ロールが持つべき管理者の最小数を制御します。ユーザから管理権限を取り消そうとすると、min\_role\_admins オプションで設定された値より管理者が少なくなってしまう場合に文が失敗することがあります。ただし、グローバルロール管理権限 (MANAGE ROLE システム権限) を持つ他のユーザがいれば問題にはなりません。このグローバルロール管理者は、管理者が指定されていないロールを管理することができるためです。

## ロール作成時に管理者を指定する

ロールを作成するとき、CREATE ROLE 文の一部として管理者を設定します。設定する場合は、その管理者に MANAGE ROLES システム権限を付与することを検討します。たとえば、次の文では myRole というロールが作成され、管理者として SallyG と IrisM が設定されます。

```
CREATE ROLE myRole WITH ADMIN SallyG, IrisM, SYS_MANAGE_ROLES_ROLE;
```

このセクションの内容:

### [スーパユーザ \[1484 ページ\]](#)

スーパユーザとは、システム権限を行使し、あらゆるロールを管理するユーザであり、データベース内であらゆる権限付きの操作を実行できるユーザです。

## 関連情報

[min\\_role\\_admins オプション \[737 ページ\]](#)

## 1.9.1.1.5.1 スーパーユーザ

スーパーユーザとは、システム権限を行使し、あらゆるロールを管理するユーザであり、データベース内であらゆる権限付きの操作を実行できるユーザです。

新規のデータベースの作成時に作成された初期ユーザは、データベーススーパーユーザであり、完全な管理権限があり、すべてのシステム権限で行使権を持っています。このユーザは、新規作成したデータベースであらゆる権限付き操作を実行できます。このユーザはしばしば DBA ユーザと呼ばれます。

データベースサーバスーパーユーザは、COCKPIT\_ROLE ユーザ定義ロールに関するすべての行使権限を持つデータベーススーパーユーザです。

新しいロールを作成して、ロール作成の一部として管理者を割り当てた場合は、管理を行うのは管理権限を与えられた被付与者に限定されます。したがって、新しいロールの管理権限をデータベーススーパーユーザに持たせる場合は、該当のユーザに管理権限を付与する必要があります。さらに、初期ユーザを削除し、独自のスーパーユーザを作成することができます。

データベーススーパーユーザを作成するには、ユーザを作成して SYS\_AUTH\_SA\_ROLE および SYS\_AUTH\_SSO\_ROLE システムロールを付与します。それにより、アップグレードで導入された新しい権限を含む、システムで可能な権限をすべて付与します。

### 関連情報

[互換ロール \[1459 ページ\]](#)

## 1.9.1.1.6 アップグレード時に付与されるシステム権限

ソフトウェアの新しいリリースによって、新しいシステム権限が導入される場合があります。

新しいデータベースでは、これらの権限は許可する権限付き操作のタイプによって、自動的に SYS\_AUTH\_SA\_ROLE または SYS\_AUTH\_SSO\_ROLE 互換性ロールに追加されます。

アップグレードされたデータベースでは、これらの権限は UPGRADE ROLE システム権限 (のみ) に追加されます。この動作では、新しい権限を付与するロールとユーザを決定する必要があります。新しい権限を付与した後、UPGRADE ROLE システム権限からそれらを取り消す必要があります。

### i 注記

UPGRADE ROLE はシステム権限ですが、付与された権限を持てるという点でロールに似ています。

このセクションの内容:

[アップグレード後 UPGRADE ROLE システム権限に付与された権限を配布する \[1485 ページ\]](#)

UPGRADE ROLE システム権限に追加された権限を他のロールやユーザに付与し、その後 UPGRADE ROLE から権限を取り消します。

## 1.9.1.1.6.1 アップグレード後 UPGRADE ROLE システム権限に付与された権限を配布する

UPGRADE ROLE システム権限に追加された権限を他のロールやユーザに付与し、その後 UPGRADE ROLE から権限を取り消します。

### 前提条件

UPGRADE ROLE システム権限の行使権限と管理権限が必要です。

### コンテキスト

アップグレード後に新しい権限が UPGRADE ROLE システム権限に追加されてから、このプロシージャを実行します。リリースに新しい権限が追加されているかを判断するには、SQL Central の UPGRADE ROLE を表示します。これは、ロールまたはユーザのロールと権限を表示するプロセスと同じです。また、リリースの新機能のリストで、新規に追加した権限についての記載を参照することもできます。

### 手順

1. Interactive SQL では、UPGRADE ROLE システム権限の管理および行使権限を持ったユーザとしてログインし、sp\_displayroles システムプロシージャを呼び出す文を実行して、UPGRADE ROLE システム権限に付与されている権限を表示します。この文を実行するときには、UPGRADE ROLE システム権限の内部表現である SYS\_UPGRADE\_ROLE\_ROLE を使用する必要があることに注意してください。

```
CALL sp_displayroles ( 'SYS_UPGRADE_ROLE_ROLE', 'expand_down' );
```

2. 他のロールやユーザにリストされている権限を付与して、管理権限が少なくとも 1 つ以上の他のロールまたはユーザに付与されているようにします。
3. 付与した各権限について、その権限の管理権限をもつユーザとしてログインし、REVOKE 文を実行して UPGRADE ROLE システム権限から権限を取り消します (ここでも、内部表現 SYS\_UPGRADE\_ROLE\_ROLE を使用します)。次に例を示します。

```
REVOKE new-privilege-name FROM SYS_UPGRADE_ROLE_ROLE;
```

### 結果

すべての新しい権限が他のユーザおよびロールに付与されて、UPGRADE ROLE には何も権限が付与されていない状態になります。

## 次のステップ

なし。

## 関連情報

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

[システム権限の付与 \(SQL Central の場合\) \[1520 ページ\]](#)

[システム権限の付与 \(SQL の場合\) \[1521 ページ\]](#)

[システム権限の取り消し \(SQL Central の場合\) \[1525 ページ\]](#)

[システム権限の取り消し \(SQL の場合\) \[1527 ページ\]](#)

### 1.9.1.1.7 ロールにロールと権限を設定する (SQL Central の場合)

任意のロールにロールや権限を付与したり、ロールや権限を取り消したり、そのロールに付与されたロールや権限に対してロールが持つ管理権限を設定したりします。

## 前提条件

システムロールを付与するか、取り消す場合は、MANAGE ROLES システム権限が必要です。

互換ロールを付与するか、取り消す場合は、そのロールの管理権限が必要です。

SYS\_RUN\_REPLICATION\_ROLE を付与するか、取り消す場合は、SYS\_REPLICATION\_ADMIN\_ROLE システムロールが必要です。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[\[ロール\]](#) をダブルクリックして、変更するロールを選択します。

オプション	アクション
権限の付与	<ol style="list-style-type: none"><li>1. 右ウィンドウ枠で、該当する権限タブをクリックします。</li><li>2. 右ウィンドウ枠の任意の場所を右クリックして、<b>新規</b> &gt; <b>権限</b> をクリックします。</li><li>3. 1つ以上の権限をリストから選択し、<a href="#">[OK]</a> をクリックします。</li></ol>

オプション	アクション
権限の取り消し	<ol style="list-style-type: none"> <li>1. 右ウィンドウ枠で、該当する権限タブをクリックします。</li> <li>2. 取り消す権限を右クリックし、<b>[削除]</b>をクリックします。</li> </ol>
ロールの付与	<ol style="list-style-type: none"> <li>1. 右ウィンドウ枠で、<b>ロール</b>タブをクリックします。</li> <li>2. 右ウィンドウ枠の任意の場所を右クリックして、<b>新規 ▶ 付与されたロール</b> をクリックします。</li> <li>3. 1つ以上の付与するロールを選択し、<b>[OK]</b>をクリックします。</li> </ol>
ロールの取り消し	<ol style="list-style-type: none"> <li>1. 右ウィンドウ枠で、<b>ロール</b>タブをクリックします。</li> <li>2. 取り消すロールを右クリックし、<b>[削除]</b>をクリックします。</li> </ol>

3. **ファイル ▶ 保存** をクリックします。

## 結果

ロールが設定されます。

## 関連情報

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

### 1.9.1.1.8 ロールにロールと権限を設定する (SQL の場合)

任意のロールにロールや権限を付与したり、ロールや権限を取り消したり、そのロールに付与されたロールや権限に対してロールが持つ管理権限を設定したりします。

## 前提条件

システムロールを付与するか、取り消す場合は、MANAGE ROLES システム権限が必要です。

互換ロールを付与するか、取り消す場合は、そのロールの管理権限が必要です。

SYS\_RUN\_REPLICATION\_ROLE を付与するか、取り消す場合は、SYS\_REPLICATION\_ADMIN\_ROLE システムロールが必要です。

## 手順

1. データベースに接続します。
2. 変更したい内容に従って、次のような文を実行します。

オプション	アクション
権限の付与	<pre>GRANT privilege-name TO target-role-name;</pre>
権限の取り消し	<pre>REVOKE privilege-name FROM target-role-name;</pre>
ロールの付与	<pre>GRANT ROLE role-name TO target-role-name;</pre>
ロールの取り消し	<pre>REVOKE ROLE role-name FROM target-role-name;</pre>

## 結果

ロールが設定されます。

### 例

RoleA に管理権限を与えずに、CREATE ANY OBJECT システム権限をロール RoleA に付与するには、次の文を実行します。

```
GRANT CREATE ANY OBJECT TO RoleA;
```

他のユーザやロールに対してシステム権限の付与や取り消しを実行する機能とともに、CREATE ANY OBJECT システム権限を RoleA に付与するには、次の文を実行します。

```
GRANT CREATE ANY OBJECT TO RoleA WITH ADMIN OPTION;
```

RoleB を管理権限とともにユーザ Jane に付与するには、次の文を実行します。

```
GRANT ROLE RoleB TO Jane WITH ADMIN OPTION;
```

ユーザ John に RoleB に対する管理権限を付与するが、RoleB を使用できないようにするには、次の文を実行します。

```
GRANT ROLE RoleB TO John WITH ADMIN ONLY OPTION;
```

## 関連情報

[ロールにロールと権限を設定する \(SQL Central の場合\) \[1486 ページ\]](#)

## 1.9.1.1.9 ロールの付与 (SQL Central の場合)

ユーザまたは別のロールにロールを付与します。

### 前提条件

付与しているロールに対する管理権限が必要です。

### コンテキスト

システムロールには管理権限を付与できません。システムロールに対して管理権限を持っているのは、MANAGE ROLES システム権限を持つユーザのみです。

### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. ユーザまたはロールをダブルクリックし、ロールを付与するユーザまたはロールを選択します。
3. 右ウィンドウ枠で、被付与者をクリックします。
4. 右ウィンドウ枠の任意の場所を右クリックして、**新規** ▶ **被付与者** を選択します。
5. ロールを付与するユーザまたはロールをクリックし、**[OK]** をクリックします。
6. (省略可) **管理**を使用します。(管理権限) ロールの管理権限を変更するためのカラム。空のカラムは、ユーザまたはロールがそのロールに対する管理権限を持たないことを示します。カラムのチェックマークは、ユーザまたはロールがロールを管理 (付与または取り消し) できることを意味します。システムロールには管理権限を付与できません。システムロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。オプションを切り替えるには、カラムをクリックします。  
  
(省略可) **発動**を使用します。(行使権限) ユーザまたはロールの行使権限を変更するためのカラム。カラムのチェックマークは、ユーザまたはロールがロールを行使できることを意味します。行使権限はデフォルトで付与されます。オプションを切り替えるには、カラムをクリックします。
7. **ファイル** ▶ **保存** をクリックします。

### 結果

ロールは、ユーザまたはロールに付与されます。

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[GRANT 文の構文の変更 \[1582 ページ\]](#)

[オブジェクトレベル権限の付与 \(SQL Central の場合\) \[1523 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

### 1.9.1.1.10 ロールの付与 (SQL の場合)

ユーザまたは別のロールにロールを付与します。

#### 前提条件

付与しているロールに対する管理権限が必要です。

#### コンテキスト

システムロールには管理権限を付与できません。システムロールを管理できるのは、MANAGE ROLES システム権限を持つユーザのみです。

#### 手順

1. データベースに接続します。
2. GRANT ROLE 文を実行します。次に例を示します。

オプション	アクション
管理権限を持たないロールを付与します。	<pre>GRANT ROLE role-name TO userid;</pre>
管理権限を持つロールを付与します。	<pre>GRANT ROLE role-name TO userid WITH ADMIN OPTION;</pre>
管理権限のみのロールを付与します。	<pre>GRANT ROLE role-name TO userid WITH ADMIN ONLY OPTION;</pre>

## 結果

ロールは、ユーザまたはロールに付与されます。

### 例

RoleB を管理権限とともにユーザ Jane に付与するには、次の文を実行します。

```
GRANT ROLE RoleB TO Jane WITH ADMIN OPTION
```

ユーザ John に RoleB に対する管理権限を付与するが、RoleB を使用できないようにするには、次の文を実行します。

```
GRANT ROLE RoleB TO John WITH ADMIN ONLY OPTION
```

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[GRANT 文の構文の変更 \[1582 ページ\]](#)

[オブジェクトレベル権限の付与 \(SQL Central の場合\) \[1523 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

### 1.9.1.1.11 ロールの取り消し (SQL Central の場合)

ユーザ、ユーザ拡張ロール、またはユーザ定義ロールからロールを取り消します。

## 前提条件

ユーザ定義ロールを取り消すには、そのロールの管理権限が必要です。システムロールを取り消すには、MANAGE ROLES システム権限が必要です。

SYS ロールからロールを取り消すことはできません。その他のシステムロールからは、そのデフォルトロールでない場合に限り、ロールを取り消すことができます。

ロールを取り消すことができるのは、依存するすべてのロールが、min\_role\_admins データベースオプションで設定された、アクティブなパスワードを持つ管理者ユーザの最小数の要件を満たす場合のみです。

## 手順

1. SQL Central で、[SQL Anywhere17 プラグイン](#)を使用してデータベースに接続します。

2. 左ウィンドウ枠で **ユーザ** または **ロール** のいずれかをクリックしてから、特定のユーザまたはロールをクリックします。
3. 次のオプションのうちの 1 つを選択してください。

管理オプション	アクション
このロールが付与されているユーザまたはロールからロールを取り消します。	<ol style="list-style-type: none"> <li>1. <b>被付与者</b> タブをクリックします。</li> <li>2. ユーザまたはロールからロールを取り消すには、そのユーザまたはロールを右クリックし、<b>[削除]</b> をクリックします。</li> </ol>
ロールの行使権限または管理権限を取り消します。	<ol style="list-style-type: none"> <li>1. <b>ロール</b> タブをクリックします。</li> <li>2. ロールを選択し、右クリックして次を選択します。 <ul style="list-style-type: none"> <li>○ <b>取り消し</b> &gt; <b>行使権限</b> 。これは、次の文を実行することと同義です。 <pre>REVOKE ROLE "role-name" FROM "user/role-name"</pre> </li> <li>○ <b>取り消し</b> &gt; <b>管理権限</b> 。これは、次の文を実行することと同義です。 <pre>REVOKE ADMIN OPTION FOR ROLE "role-name" FROM "user/role-name"</pre> </li> </ul> </li> </ol>

#### ➔ ヒント

変更を元に戻すには、1 つ以上のローを選択して、**編集** > **元に戻す** をクリックします。

**凡例** をオンまたはオフにするには、**ファイル** > **凡例を表示** をクリックします。

4. **ファイル** > **保存** をクリックします。

## 結果

ロールは、指定されたユーザまたはロールから取り消されます。

## 関連情報

[互換ロールをユーザ定義ロールに移行する \(SQL Central の場合\) \[1462 ページ\]](#)

## 1.9.1.1.12 ロールの取り消し (SQL の場合)

ユーザ、ユーザ拡張ロール、またはユーザ定義ロールからロールを取り消します。

### 前提条件

ユーザ定義ロールを取り消すには、そのロールの管理権限が必要です。システムロールを取り消すには、MANAGE ROLES システム権限が必要です。

SYS ロールからロールを取り消すことはできません。その他のシステムロールからは、そのデフォルトロールでない場合に限り、ロールを取り消すことができます。

ロールを取り消すことができるのは、依存するすべてのロールが、min\_role\_admins データベースオプションで設定された、アクティブなパスワードを持つ管理者ユーザの最小数の要件を満たす場合のみです。

### 手順

1. データベースに接続します。
2. REVOKE ROLE 文を実行します。次に例を示します。

オプション	アクション
ロールの取り消し	<pre>REVOKE ROLE role-name FROM userid;</pre>
ロールの管理権限を取り消しますが、ユーザは引き続きロールを行使できます	<pre>REVOKE ADMIN OPTION FOR ROLE role-name FROM userid;</pre>

### 結果

ロールは、指定されたユーザまたはロールから取り消されます。

#### 例

SYS\_AUTH\_RESOURCE\_ROLE 互換ロールをユーザ Jim から取り消します。

```
REVOKE ROLE SYS_AUTH_RESOURCE_ROLE FROM Jim;
```

myRole ロールに対する管理権限のみをユーザ AnnW から取り消します。

```
REVOKE ADMIN OPTION FOR ROLE myRole FROM AnnW;
```

## 関連情報

[ロールの取り消し \(SQL Central の場合\) \[1491 ページ\]](#)

[互換ロールをユーザ定義ロールに移行する \(SQL Central の場合\) \[1462 ページ\]](#)

### 1.9.1.1.13 ロールの削除 (SQL Central の場合)

ユーザ定義、ユーザ拡張、および互換の各ロールを削除します。

#### 前提条件

削除しようとしているロールに対する管理権限が必要です。

削除するロールがデータベースオブジェクトを所有してはいけません。

ロールを削除できるのは、依存するすべてのロールが、`min_role_admins` データベースオプションで設定された、アクティブなパスワードを持つ管理者ユーザの最小数の要件を満たす場合のみです。

#### コンテキスト

システムロールは削除できません。

#### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. **ロール**をダブルクリックして、削除するロールを選択します。
3. ロールを右クリックして、**[削除]**をクリックします。

#### 結果

ロールが削除され、そのロールに付与されていたユーザまたはロールが取り消されます。MANAGE ROLES システム権限で管理可能なロールを削除するとき、そのロールも MANAGE ROLES システム権限から取り消されます。

## 関連情報

[ユーザ拡張ロールからユーザへの逆変換 \(SQL Central の場合\) \[1479 ページ\]](#)

[min\\_role\\_admins オプション \[737 ページ\]](#)

### 1.9.1.1.14 ロールの削除 (SQL の場合)

ユーザ定義、ユーザ拡張、および互換の各ロールを削除します。

#### 前提条件

削除しようとしているロールに対する管理権限が必要です。

ロールを削除できるのは、依存するすべてのロールが、min\_role\_admins データベースオプションで設定された、アクティブなパスワードを持つ管理者ユーザの最小数の要件を満たす場合のみです。

#### コンテキスト

システムロールは削除できません。

#### 手順

DROP ROLE 文を実行します。次に例を示します。

```
DROP ROLE role-name;
```

ロールがユーザに割り当てられている場合、ロールを削除するには WITH REVOKE 句を指定する必要があります。

```
DROP ROLE role-name WITH REVOKE;
```

ロールがオブジェクトを所有している場合、ロールを削除するには WITH DROP OBJECTS 句を指定する必要があります。

```
DROP ROLE role-name WITH DROP OBJECTS;
```

#### 結果

ロールが削除されます。

## 例

組織内で誰も SYS\_AUTH\_VALIDATE\_ROLE 互換ロールを付与されていないとします。次の文を実行すれば、削除することができます。

```
DROP ROLE SYS_AUTH_VALIDATE_ROLE;
```

SYS\_AUTH\_VALIDATE\_ROLE 互換ロールを付与されているユーザがいる場合、次の文を実行すれば削除することができます。

```
DROP ROLE SYS_AUTH_VALIDATE_ROLE WITH REVOKE;
```

次の文では、SYS\_AUTH\_DBA\_ROLE システムロールが削除されます。この操作は、付随するすべてのシステム権限が他のロールを管理し、min\_role\_admins データベースオプションで指定されているように、これらのロールの管理者数が適切である場合に可能です。ロールが他のユーザに割り当てられている可能性があるため、WITH REVOKE 句が追加されます。

```
DROP ROLE SYS_AUTH_DBA_ROLE WITH REVOKE;
```

## 関連情報

[互換ロール \[1459 ページ\]](#)

[min\\_role\\_admins オプション \[737 ページ\]](#)

## 1.9.1.2 権限

権限とは、システムで権限付き操作を実行するための権利です。

たとえば、テーブルを変更することは、変更の種類によっては権限付き操作です。権限には、システム権限とオブジェクトレベル権限の 2 種類があります。システム権限は権限付け操作を実行するための一般的な権利を付与しますが、オブジェクトレベル権限は特定オブジェクトでの操作の実行を制限します。たとえば、ALTER ANY TABLE システム権限では、システムの任意のテーブルを変更できます。この権限を持っていない場合、作成したテーブルか、ALTER TABLE オブジェクトレベル権限を持つテーブルだけを編集できます。

システム権限はデータベースに組み込まれ、付与や取り消しが可能ですが、作成や削除はできません。MANAGE ROLES 権限と UPGRADE ROLE 権限を除き、システム権限は付与されたり、取り消されたりするシステム権限を持ってません。システム権限のそれぞれは、SET USER システム権限を除き、デフォルトで SYS\_AUTH\_SA\_ROLE または SYS\_AUTH\_SSO\_ROLE 互換性ロールのいずれかに付与されますが、両方には付与されていません。SET USER システム権限は、両方のロールに付与されます (WITH ADMIN OPTION は SYS\_AUTH\_SSO\_ROLE に、WITH NO ADMIN OPTION は SYS\_AUTH\_SA\_ROLE に)。

GRANT 文および REVOKE 文を使用して、システム権限とオブジェクトレベル権限を付与および取り消します。

このセクションの内容:

[オブジェクトレベル権限 \[1498 ページ\]](#)

付与することができるオブジェクトレベル権限は複数あります。

#### [システム権限 \[1502 ページ\]](#)

付与することができるシステム権限が多数あります。

#### [ユーザ権限の評価方法 \[1520 ページ\]](#)

ロールが原因で、個々のユーザの権限が複雑になります。

#### [システム権限の付与 \(SQL Central の場合\) \[1520 ページ\]](#)

システム権限を付与して、ユーザまたはロールに、データベースでの操作の実行を許可します。

#### [システム権限の付与 \(SQL の場合\) \[1521 ページ\]](#)

システム権限を付与して、ユーザまたはロールに、データベースでの操作の実行を許可します。

#### [オブジェクトレベル権限の付与 \(SQL Central の場合\) \[1523 ページ\]](#)

オブジェクトレベル権限を付与して、テーブル、ビュー、プロシージャ、シーケンス、または DB 領域などの特定のオブジェクトへのアクセスを許可します。

#### [オブジェクトレベル権限の付与 \(SQL の場合\) \[1524 ページ\]](#)

オブジェクトレベル権限を付与して、テーブル、ビュー、プロシージャ、シーケンス、または DB 領域などの特定のオブジェクトへのアクセスを許可します。

#### [システム権限の取り消し \(SQL Central の場合\) \[1525 ページ\]](#)

ユーザ、ユーザ拡張ロール、またはユーザ定義ロールからシステム権限を取り消します。

#### [システム権限の取り消し \(SQL の場合\) \[1527 ページ\]](#)

ユーザ、ユーザ拡張ロール、またはユーザ定義ロールからシステム権限を取り消します。

#### [オブジェクトレベル権限の取り消し \(SQL Central の場合\) \[1528 ページ\]](#)

ユーザまたはロールに付与されているオブジェクトレベル権限を取り消して、テーブル、ビュー、プロシージャ、シーケンス、または DB 領域などの特定のオブジェクトへのアクセスを制限します。

#### [オブジェクトレベル権限の取り消し \(SQL の場合\) \[1529 ページ\]](#)

ユーザまたはロールに付与されているオブジェクトレベル権限を取り消して、テーブル、ビュー、プロシージャ、シーケンス、または DB 領域などの特定のオブジェクトへのアクセスを制限します。

## 関連情報

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

## 1.9.1.2.1 オブジェクトレベル権限

付与することができるオブジェクトレベル権限は複数あります。

権限名	データベースオブジェクトでのサポート対象	グループメンバーシップを通して継承可能か	付与時に WITH GRANT OPTION が許可されるか	説明
ALL	テーブル、ビュー	○	○	データベースオブジェクトに関連するすべてのタスクを実行することをユーザに許可します。この権限によって、テーブルに次の権限が付与されます。ALTER、DELETE、INSERT、REFERENCES、SELECT、および UPDATE。この権限によって、ビューに次の権限が付与されます。DELETE、INSERT、SELECT、および UPDATE。ALL を付与すると、LOAD および TRUNCATE 権限も付与されます。
ALTER	テーブル	○	○	テーブルの構造を変更することや、テーブルにトリガを作成することをユーザに許可します。この権限は、データベーススキーマを変更する権限をユーザに付与するものなので、ほとんどのユーザには付与する必要はありません。
CONNECT	データベース	×	×	現在のデータベースに接続することをユーザに許可します。
CONSOLIDATE	ユーザ	×	×	SQL Remote で統合データベースを特定することをユーザに許可します。

権限名	データベースオブジェクトでのサポート対象	グループメンバーシップを通して継承可能か	付与時に WITH GRANT OPTION が許可されるか	説明
CREATE ON	DB 領域	○	○	DB 領域で作成することをユーザに許可します。必要となる追加の権限は、作成されるオブジェクトによって異なります。たとえば、テーブルを作成するには、CREATE TABLE、CREATE ANY TABLE、または CREATE ANY OBJECT のいずれかが必要です。
DELETE	テーブル、ビュー	×	○	テーブルまたはビューからローを削除することをユーザに許可します。
EXECUTE	プロシージャ、ユーザ定義関数	×	×	プロシージャまたはユーザ定義関数を実行することをユーザに許可します。
INSERT	テーブル、ビュー	○	○	テーブルまたはビューにローを挿入することをユーザに許可します。
INTEGRATED LOGIN	データベース	×	×	統合化ログインを使用して現在のデータベースに接続することをユーザに許可します。
KERBEROS LOGIN	データベース	×	×	Kerberos ログインを使用して現在のデータベースに接続することをユーザに許可します。
LOAD	テーブル	○	○	-gl データベースオプションが NONE 以外のものに設定されている場合、テーブルをロードすることをユーザに許可します。
PUBLISH	ユーザ	×	×	SQL Remote のデータベースのユーザ ID を特定します。

権限名	データベースオブジェクトでのサポート対象	グループメンバーシップを通して継承可能か	付与時に WITH GRANT OPTION が許可されるか	説明
REFERENCES	テーブル	○	○	<p>テーブルへのインデックスの作成とテーブルを参照する外部キーの作成をユーザに許可します。また、この権限は、テーブル内の個別のカラムに対しても付与できます。この権限は、データベーススキーマを変更することをユーザに許可するものなので、ほとんどのユーザには付与する必要はありません。</p> <p>カラム名を指定する場合、ユーザはテーブル内のそれらのインデックスのみを参照できます。</p>
REMOTE	ユーザ	×	×	<p>SQL Remote および Mobile Link でリモートデータベースを特定することをユーザに許可します。</p>
SELECT	テーブル、ビュー	○	○	<p>テーブルまたはビューへの問い合わせをユーザに許可します。また、この権限は、テーブル内の個別のカラムに対しても付与できます。</p> <p>カラム名を指定する場合、ユーザはテーブル内のそれらのカラムのみを参照できます。</p>
TRUNCATE	テーブル、実体化ビュー	○	○	<p>テーブルまたは実体化ビューのトランケートをユーザに許可します。</p>
UPDATE	テーブル、ビュー	○	○	<p>テーブルまたはビューでローを更新することをユーザに許可します。また、この権限は、テーブル内の個別のカラムに対しても付与できます。</p>

権限名	データベースオブジェクトでのサポート対象	グループメンバーシップを通して継承可能か	付与時に WITH GRANT OPTION が許可されるか	説明
USAGE	シーケンスジェネレータ	×	○	シーケンス内の現在の値または次の値を評価することをユーザに許可します。

## 追加の注意事項

### WITH GRANT OPTION はグループメンバーには継承されません

一部のオブジェクトレベル権限は継承可能ですが、WITH GRANT OPTION は継承不可です。たとえば、ユーザ拡張ロールの UserA がテーブルで SELECT を付与され、WITH GRANT OPTION が指定されている場合、UserA はテーブルで選択し、そのテーブルでの選択を他のユーザに付与できます。そのユーザ拡張ロールの被付与者は、テーブルから選択する機能を継承しますが、SELECT 権限を他のユーザに付与する機能は継承しません。

### ビュー

無効にされたビューに対する権限を付与できます。無効なビューに対する権限はデータベースに保存され、ビューが有効になると使用できるようになります。

### プロシージャ

実行するには、ユーザがプロシージャを所有しているか、EXECUTE 権限を持っている必要があります。

さらに、プロシージャ定義に SQL SECURITY DEFINER が含まれている場合、プロシージャは、ユーザのプロシージャ呼び出しではなく、プロシージャ所有者の権限で実行されます。プロシージャ定義に SQL SECURITY INVOKER が含まれている場合、プロシージャは呼び出し側が持つ権限で実行されます。

### DB 領域

DB 領域での CREATE ON 権限が付与されているユーザは、作成したいオブジェクトを作成するための適切な CREATE 権限も持っている必要があります。

### シーケンス

USAGE 権限は、シーケンスジェネレータに付与される唯一の権限です。この権限は、ユーザがシーケンス内の現在の値または次の値を評価できるようにします。シーケンスがテーブルの DEFAULT 句の一部である場合、テーブルにローを挿入するすべてのユーザにシーケンスに対する権限が必要です。

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[Windows 統合化ログイン \[124 ページ\]](#)

[Kerberos ユーザ認証 \[149 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

[オブジェクトレベル権限の付与 \(SQL Central の場合\) \[1523 ページ\]](#)

## 1.9.1.2.2 システム権限

付与することができるシステム権限が多数あります。

ユーザをクリックし、表示される詳細を確認すれば、ユーザが持つ SQL Central のロールと権限を表示できます。また、sp\_displayroles システムプロシージャを使用して詳細を取得することもできます。

### i 注記

デフォルトでは、すべてのシステム権限が、SYS\_RUN\_REPLICATION\_ROLE システムロールを持つユーザによって継承されます。SYS\_AUTH\_DBA\_ROLE 互換性ロールが SYS\_RUN\_REPLICATION\_ROLE に付与されているため、この設定が必要です。継承されないようにするには、SYS\_RUN\_REPLICATION\_ROLE から SYS\_AUTH\_DBA\_ROLE 互換性ロールを取り消し、レプリケーションに必要な権限のみを SYS\_RUN\_REPLICATION\_ROLE システムロールに付与します。ただし、SYS\_RUN\_REPLICATION\_ROLE システムロールを付与されたユーザは、リモート接続でこのロールの権限のみを行使できます。

## システム権限のリスト

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
データベース		
ALTER DATABASE ...	次のことをユーザに許可します。 <ul style="list-style-type: none"><li>データベースを更新します。</li><li>コストモデル調整を実行します。</li><li>データベース統計をロードします。</li><li>トランザクションログを変更します (SERVER OPERATOR システム権限も必要)。</li></ul>	<ul style="list-style-type: none"><li>SYS_AUTH_DBA_ROLE</li><li>SYS_AUTH_SA_ROLE</li><li>SYS_RUN_REPLICATION_ROLE</li></ul>
BACKUP DATABASE	ユーザはデータベースをバックアップできます。	<ul style="list-style-type: none"><li>DIAGNOSTICS</li><li>SYS_AUTH_BACKUP_ROLE</li><li>SYS_AUTH_DBA_ROLE</li><li>SYS_AUTH_PROFILE_ROLE</li><li>SYS_AUTH_SA_ROLE</li><li>SYS_RUN_REPLICATION_ROLE</li><li>SYS_SAMONITOR_ADMIN_ROLE</li></ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
CHECKPOINT	ユーザはデータベースサーバにチェックポイントを強制的に実行させることができます。	<ul style="list-style-type: none"> <li>● SYS_AUTH_DBA_ROLE</li> <li>● SYS_AUTH_SA_ROLE</li> <li>● SYS_RUN_REPLICATION_ROLE</li> <li>● SYS_SAMONITOR_ADMIN_ROLE</li> </ul>
DROP CONNECTION	ユーザはデータベースへの任意の接続を削除できます。	<ul style="list-style-type: none"> <li>● SYS_AUTH_DBA_ROLE</li> <li>● SYS_AUTH_SSO_ROLE</li> <li>● SYS_RUN_REPLICATION_ROLE</li> </ul>
MANAGE PROFILING	ユーザは、データベースサーバトレーシングを管理できません。	<ul style="list-style-type: none"> <li>● SYS_AUTH_DBA_ROLE</li> <li>● SYS_AUTH_PROFILE_ROLE</li> <li>● SYS_AUTH_SA_ROLE</li> <li>● SYS_RUN_REPLICATION_ROLE</li> <li>● SYS_RUN_PROFILER_ROLE</li> </ul>
MONITOR	ユーザは、権限のある統計、接続されたユーザ、ロックへのアクセスを含め、データベースをモニタできます。	<ul style="list-style-type: none"> <li>● SYS_AUTH_DBA_ROLE</li> <li>● SYS_AUTH_SA_ROLE</li> <li>● SYS_RUN_REPLICATION_ROLE</li> <li>● SYS_SAMONITOR_ADMIN_ROLE</li> <li>● SYS_RUN_PROFILER_ROLE</li> </ul>
データベースオプション		
SET ANY PUBLIC OPTION	ユーザは、SET ANY SECURITY OPTION システム権限または SET ANY SYSTEM OPTION システム権限がない PUBLIC データベースオプションを設定できます。	<ul style="list-style-type: none"> <li>● SYS_AUTH_DBA_ROLE</li> <li>● SYS_AUTH_SA_ROLE</li> <li>● SYS_RUN_REPLICATION_ROLE</li> <li>● SYS_REPLICATION_ADMIN_ROLE</li> </ul>
SET ANY SECURITY OPTION	ユーザは、任意の PUBLIC セキュリティデータベースオプションを設定できます。	<ul style="list-style-type: none"> <li>● SYS_AUTH_DBA_ROLE</li> <li>● SYS_AUTH_SSO_ROLE</li> <li>● SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
SET ANY SYSTEM OPTION	ユーザは、PUBLIC システムデータベースオプションを設定できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
SET ANY USER DEFINED OPTION	ユーザは、ユーザ定義データベースオプションを設定できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> </ul>
データ型		
ALTER DATATYPE	ユーザはデータ型を変更できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE DATATYPE	ユーザはデータ型を作成できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_RESOURCE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
DROP DATATYPE	ユーザはデータ型を削除できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
DB 領域		
MANAGE ANY DBSPACE	<p>次のことをユーザに許可します。</p> <ul style="list-style-type: none"> <li>• DB 領域での作成、変更、削除、コメント。</li> <li>• CREATE 権限を DB 領域に付与したり、DB 領域から取り消したりします。</li> <li>• データを任意の DB 領域に移動します。</li> <li>• データベース削除ファイル関数を実行します。</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_SAMONITOR_ADMIN_ROLE</li> </ul>
デバッグ		

オブジェクトまたは機能に応じたシステム権限	説明	ユーザー/ロールは次から継承可能
DEBUG ANY PROCEDURE	ユーザは、プロシージャ、関数、トリガ、イベント、イベントをデバッグできます。	<ul style="list-style-type: none"> <li>• SA_DEBUG</li> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
イベント		
MANAGE ANY EVENT	ユーザはイベントの作成、変更、削除、トリガ、コメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_SAMONITOR_ADMIN_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
外部環境		
CREATE EXTERNAL REFERENCE	<p>データベースで外部参照を作成することをユーザに許可します。</p> <p>外部参照を作成する前に、特定のデータベースオブジェクトを作成するときに必要なシステム権限が必要です。</p> <p>たとえば、外項ブレーカーおよび事前フィルタを使用する自己所有テキスト設定オブジェクトを作成するには、CREATE TEXT CONFIGURATION と CREATE EXTERNAL REFERENCE 両方のシステム権限が必要です。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_SAMONITOR_ADMIN_ROLE</li> </ul>
MANAGE ANY EXTERNAL ENVIRONMENT	ユーザは、外部環境の変更、コメントの追加、起動、停止を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
MANAGE ANY EXTERNAL OBJECT	ユーザは、外部環境オブジェクトのインストール、コメントの追加、削除を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
ファイル、ディレクトリ、およびディスクドライブ		
ACCESS DISK INFORMATION	sp_disk_info システムプロシージャを使用して、合計のディスクサイズおよび残りのディスク領域についての情報にアクセスできます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザーロールは次から継承可能
READ CLIENT FILE	ユーザはクライアントコンピュータ上のファイルを読み込みます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_READCLIENTFILE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
READ FILE	ユーザはデータベースサーバコンピュータ上のファイルを読み込みます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_READFILE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
WRITE CLIENT FILE	ユーザはクライアントコンピュータ上にファイルを書き込みます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_AUTH_WRITECLIENTFILE_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
WRITE FILE	ユーザはデータベースサーバコンピュータ上にファイルを書き込みます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_AUTH_WRITEFILE_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
<b>インデックス</b>		
ALTER ANY INDEX	ユーザは、任意のユーザが所有するテーブルとビューでインデックスとテキストインデックスに対して変更とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• DIAGNOSTICS</li> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_PROFILE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE ANY INDEX	ユーザは、任意のユーザが所有するテーブルとビューでインデックスとテキストインデックスの作成とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
DROP ANY INDEX	ユーザは、任意のユーザが所有するテーブルとビューのインデックスとテキストインデックスを削除できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
<b>LDAP</b>		
MANAGE ANY LDAP SERVER	ユーザは LDAP サーバの作成、変更、削除、検証、コメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
<b>OData</b>		
MANAGE ODATA	ユーザは OData プロデューサオブジェクトを作成、変更、および削除することができます。	<ul style="list-style-type: none"> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> </ul>
VERIFY ODATA	ユーザは CREATE ODATA PRODUCER および ALTER ODATA PRODUCER 文の AUTHENTICATION USER または ADMIN USER 句を参照することができます。この権限のないユーザが参照すると、OData プロデューサはエラーを返します。	<ul style="list-style-type: none"> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> </ul>
<b>マテリアライズドビュー</b>		
ALTER ANY MATERIALIZED VIEW	ユーザは、任意のユーザが所有するマテリアライズドビューに対して変更とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE ANY MATERIALIZED VIEW	ユーザは、任意のユーザが所有するマテリアライズドビューの作成とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE MATERIALIZED VIEW	ユーザは、自己所有のマテリアライズドビューを作成したりコメントを加えたりできます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_RESOURCE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
DROP ANY MATERIALIZED VIEW	ユーザは、任意のユーザが所有するマテリアライズドビューを削除できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
<b>Messages</b>		
CREATE MESSAGE	ユーザはメッセージを作成できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_RESOURCE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザー/ロールは次から継承可能
DROP MESSAGE	ユーザはメッセージを削除できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
その他		
ALTER ANY OBJECT	<p>ユーザは、任意のユーザが所有する次のタイプのオブジェクトに対して変更とコメントの追加を実行できます。</p> <ul style="list-style-type: none"> <li>• データ型</li> <li>• イベント</li> <li>• 関数</li> <li>• インデックス</li> <li>• マテリアライズドビュー</li> <li>• Messages</li> <li>• プロシージャ</li> <li>• シーケンスジェネレータ</li> <li>• 空間参照システム</li> <li>• 空間計測単位</li> <li>• 統計情報</li> <li>• テーブル</li> <li>• テキスト設定オブジェクト</li> <li>• テキストインデックス</li> <li>• トリガ</li> <li>• ビュー</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
ALTER ANY OBJECT OWNER	ユーザは任意のタイプのオブジェクトの所有者を変更できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
COMMENT ANY OBJECT	CREATE ANY OBJECT システム権限を使用して作成可能なあらゆるタイプのオブジェクトにコメントを付けることをユーザに許可します。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
CREATE ANY OBJECT	<p>ユーザは、任意のユーザが所有する次のタイプのオブジェクトを作成し、コメントを追加できます。</p> <ul style="list-style-type: none"> <li>• データ型</li> <li>• イベント</li> <li>• 関数</li> <li>• インデックス</li> <li>• マテリアライズドビュー</li> <li>• Messages</li> <li>• プロシージャ</li> <li>• シーケンスジェネレータ</li> <li>• 空間参照システム</li> <li>• 空間計測単位</li> <li>• 統計情報</li> <li>• テーブル</li> <li>• テキスト設定オブジェクト</li> <li>• テキストインデックス</li> <li>• トリガ</li> <li>• ビュー</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
DROP ANY OBJECT	<p>ユーザは、任意のユーザが所有する次のタイプのオブジェクトを削除できます。</p> <ul style="list-style-type: none"> <li>• データ型</li> <li>• イベント</li> <li>• 関数</li> <li>• インデックス</li> <li>• マテリアライズドビュー</li> <li>• Messages</li> <li>• プロシージャ</li> <li>• シーケンスジェネレータ</li> <li>• 空間参照システム</li> <li>• 空間計測単位</li> <li>• 統計情報</li> <li>• テーブル</li> <li>• テキスト設定オブジェクト</li> <li>• テキストインデックス</li> <li>• トリガ</li> <li>• ビュー</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザーロールは次から継承可能
MANAGE ANY OBJECT PRIVILEGE	<p>次のことをユーザに許可します。</p> <ul style="list-style-type: none"> <li>任意のユーザが所有するテーブルに対する SELECT、INSERT、DELETE、UPDATE、ALTER、REFERENCES、LOAD、および TRUNCATE の各権限を付与したり取り消したりします。</li> <li>任意のユーザが所有するビューに対する SELECT、INSERT、DELETE、および UPDATE の各権限を付与したり取り消したりします。</li> <li>任意のユーザが所有するプロシージャおよび関数に対する EXECUTE 権限を付与したり取り消したりします。</li> <li>任意のユーザが所有するシーケンスジェネレータに対する USAGE 権限を付与したり取り消したりします。</li> <li>CREATE 権限を DB 領域に付与したり、DB 領域から取り消したりします。</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_SAMONITOR_ADMIN_ROLE</li> </ul>
MANAGE CACHED PLANS	ユーザが任意の接続についてのキャッシュされたプランに関する統計にアクセスできます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
MANAGE CERTIFICATES	ユーザは証明書の作成、変更、削除、コメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
MANAGE TIME ZONE	ユーザは、シミュレートされたタイムゾーンを作成、変更、および削除できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
REORGANIZE ANY OBJECT	ユーザはテーブルとマテリアライズドビューを再編成できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
VALIDATE ANY OBJECT	ユーザは、テーブル、マテリアライズドビュー、インデックス、テキストインデックスを検証できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_AUTH_VALIDATE_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
ミラーサーバ		

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
MANAGE ANY MIRROR SERVER	次のことをユーザに許可します。 <ul style="list-style-type: none"> <li>ミラーサーバの作成、変更、削除、コメントの追加。</li> <li>ミラーサーバのパラメータの変更。</li> <li>ミラーサーバオプションの設定。</li> </ul>	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> </ul>
ミューテックスおよびセマフォ		<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> </ul>
CREATE ANY MUTEX SEMAPHORE	ユーザは、任意のユーザが所有するミューテックスまたはセマフォの作成を実行できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> </ul>
UPDATE ANY MUTEX SEMAPHORE	ユーザは、任意のユーザが所有するミューテックスまたはセマフォの更新を実行できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> </ul>
DROP ANY MUTEX SEMAPHORE	ユーザは、任意のユーザが所有するミューテックスまたはセマフォの削除を実行できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> </ul>
プロシージャ		
ALTER ANY PROCEDURE	ユーザは、任意のユーザが所有するプロシージャと関数に対して変更とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> <li>SYS_SAMONITOR_ADMIN_ROLE</li> <li>SYS_RUN_PROFILER_ROLE</li> </ul>
CREATE ANY PROCEDURE	ユーザは、任意のユーザが所有するプロシージャと関数の作成とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> <li>SYS_REPLICATION_ADMIN_ROLE</li> <li>SYS_SAMONITOR_ADMIN_ROLE</li> <li>SYS_RUN_PROFILER_ROLE</li> </ul>
CREATE PROCEDURE	ユーザは、自己所有のプロシージャと関数を作成できません。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_RESOURCE_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
DROP ANY PROCEDURE	ユーザは、任意のユーザが所有するプロシージャと関数を削除できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
EXECUTE ANY PROCEDURE	ユーザは、任意のユーザが所有するプロシージャと関数を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
<b>監査</b>		
MANAGE AUDITING	<p>次の監査関連のシステムプロシージャを実行することをユーザに許可します。</p> <ul style="list-style-type: none"> <li>• sa_audit_string</li> <li>• sp_trace_events (include_audit_events ビットセットを含む)</li> <li>• sp_trace_event_fields (include_audit_events ビットセットを含む)</li> <li>• sp_trace_event_session_events (include_audit_events ビットセットを含む)</li> <li>• sp_trace_event_session_targets (include_audit_events ビットセットを含む)</li> <li>• sp_trace_event_session_target_options (include_audit_events ビットセットを含む)</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
<b>電子メール</b>		
SEND EMAIL	<p>次の電子メール関連のシステムプロシージャを実行することをユーザに許可します。</p> <ul style="list-style-type: none"> <li>• xp_get_mail_error_code</li> <li>• xp_get_mail_error_text</li> <li>• xp_startmail</li> <li>• xp_stopmail</li> <li>• xp_startsmtp</li> <li>• xp_stopsmtp</li> <li>• xp_sendmail</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
<b>レプリケーション</b>		

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
MANAGE REPLICATION	<p>次のことをユーザに許可します。</p> <ul style="list-style-type: none"> <li>• パブリケーションの作成、変更、削除、コメント。</li> <li>• Mobile Link ユーザの作成、変更、削除。</li> <li>• SQL Remote および同期サブスクリプションの作成、変更、削除、起動、停止、および同期。</li> <li>• 同期プロファイルの作成、変更、削除、コメント。</li> <li>• SQL Remote メッセージタイプの作成、変更、削除、コメント。</li> <li>• リモートオプションの設定。</li> <li>• スキーマ同期の開始と停止。</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
ロール		
MANAGE ROLES	<p>新しいロールを作成し、新しいロールと既存のロールに対してグローバル管理者として動作することをユーザに許可します。デフォルトでは、新規作成された各ロールに対して MANAGE ROLES が付与される管理権限です。</p> <p>また、ロールの管理はロールの作成中または作成後のいずれかに、ユーザに対して直接付与することもできます。ユーザに直接付与された場合、ユーザにはロールを管理するための MANAGE ROLES システム権限は必要ありません。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
UPGRADE ROLE	<p>ユーザが、SQL Anywhere データベースを 16.0 からアップグレードするときに導入されるシステム権限のデフォルトの管理者になることを許可します。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
シーケンス		
ALTER ANY SEQUENCE	<p>ユーザは、任意のユーザが所有するシーケンスジェネレータを変更できます。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE ANY SEQUENCE	<p>ユーザは、所有者に関係なくシーケンスジェネレータを作成できます。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_RESOURCE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
DROP ANY SEQUENCE	<p>ユーザは、任意のユーザが所有するシーケンスジェネレータを削除できます。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
USE ANY SEQUENCE	ユーザは、任意のユーザが所有するシーケンスジェネレータを使用できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
サーバオペレータ		
MANAGE ANY PROPERTY HISTORY	ユーザは、データベースサーバプロパティ値の追跡をオンにして設定することができます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
MANAGE LISTENERS	ユーザはネットワークリスナーを開始および停止できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
SERVER OPERATOR	<p>次のことをユーザに許可します。</p> <ul style="list-style-type: none"> <li>• データベースの作成、リストア、削除、起動、および停止、データベース所有権の変更、カタログ (のみ) のリストア。</li> <li>• リモートサーバおよびディレクトリアクセスサーバの作成、変更、および削除。</li> <li>• サーバキャッシュの管理。</li> <li>• データベースサーバの起動および停止。</li> <li>• 暗号化および複合化データベースとファイルの作成。</li> <li>• データベーストランケーションログの変更。</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_SAMONITOR_ADMIN_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
空間オブジェクト		
MANAGE ANY SPATIAL OBJECT	ユーザは、空間参照系と空間測定単位の作成、変更、削除、コメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_SPATIAL_ADMIN_ROLE</li> </ul>
統計情報		
MANAGE ANY STATISTICS	ユーザは、任意のテーブルのデータベース統計の作成、変更、削除、更新を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
テーブル		

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
ALTER ANY TABLE	<p>次のことをユーザに許可します。</p> <ul style="list-style-type: none"> <li>任意のユーザが所有するテーブル (プロキシテーブルを含む) の変更とコメントの追加。</li> <li>任意のユーザが所有するテーブル (プロキシテーブルを含む) のカラムへのコメント。</li> </ul>	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE ANY TABLE	<p>次のことをユーザに許可します。</p> <ul style="list-style-type: none"> <li>任意のユーザが所有するテーブル (プロキシテーブルを含む) の作成とコメントの追加。</li> <li>任意のユーザが所有するテーブル (プロキシテーブルを含む) のカラムへのコメント。</li> </ul>	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_REPLICATION_ADMIN_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> <li>SYS_SAMONITOR_ADMIN_ROLE</li> <li>SYS_RUN_PROFILER_ROLE</li> </ul>
CREATE PROXY TABLE	ユーザは、自己所有のプロキシテーブルを作成できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_RESOURCE_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE TABLE	ユーザは、自己所有のテーブルを作成できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_RESOURCE_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> </ul>
DELETE ANY TABLE	ユーザは、任意のユーザが所有するテーブルとビューのローを削除できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> <li>SYS_RUN_PROFILER_ROLE</li> </ul>
DROP ANY TABLE	ユーザは、任意のユーザが所有するテーブル (プロキシテーブルを含む) を削除できます。	<ul style="list-style-type: none"> <li>SYS_AUTH_DBA_ROLE</li> <li>SYS_AUTH_SA_ROLE</li> <li>SYS_REPLICATION_ADMIN_ROLE</li> <li>SYS_RUN_REPLICATION_ROLE</li> <li>SYS_RUN_PROFILER_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
INSERT ANY TABLE	ユーザは、任意のユーザが所有するテーブルとビューにローを挿入できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
LOAD ANY TABLE	ユーザは、任意のユーザが所有するテーブルにデータをロードできます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
SELECT ANY TABLE	ユーザは、任意のユーザが所有するテーブルとマテリアライズドビューに問い合わせることができます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
TRUNCATE ANY TABLE	ユーザは、任意のユーザが所有するテーブルとマテリアライズドビューのデータをトランケートできます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
UPDATE ANY TABLE	ユーザは、任意のユーザが所有するテーブルとビューのローを更新できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
テキスト設定オブジェクト		
ALTER ANY TEXT CONFIGURATION	ユーザは、任意のユーザが所有するテキスト設定オブジェクトに対して変更とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE ANY TEXT CONFIGURATION	ユーザは、任意のユーザが所有するテキスト設定オブジェクトの作成とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE TEXT CONFIGURATION	ユーザは、自己所有のテキスト設定オブジェクトを作成できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_RESOURCE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
DROP ANY TEXT CONFIGURATION	ユーザは、任意のユーザが所有するテキスト設定オブジェクトを削除できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
トレース		
MANAGE ANY TRACE SESSION	ユーザは、ユーザトレースイベントのトリガを除き、すべてのイベントトレーシング関連の操作を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
NOTIFY TRACE EVENT	ユーザはユーザトレースイベントをトリガできます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_RUN_PROFILER_ROLE</li> </ul>
トリガ		
ALTER ANY TRIGGER	ユーザは、テーブルおよびビューに対して変更とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_RESOURCE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE ANY TRIGGER	ユーザは、テーブルおよびビューに対して作成とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_RESOURCE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
ユーザおよびログインの管理		
ACCESS USER PASSWORD	ユーザは、パスワードハッシュを含むビューにアクセスし、データベースのアンロード、抽出、比較など、パスワードへのアクセスを伴う操作を実行できるようになります。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
CHANGE PASSWORD	<p>ユーザは任意のユーザのユーザパスワードを管理できません。</p> <p>このシステム権限は、すべてのユーザに適用したり、特定のユーザセットあるいは1つ以上の指定ロールを付与されたユーザに限定したりできます。</p> <p>このシステム権限は、ユーザ所有のパスワードの変更には必要ありません。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
MANAGE ANY LOGIN POLICY	<p>ユーザはログインポリシーの作成、変更、削除、コメントの追加を実行できます。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_SAMONITOR_ADMIN_ROLE</li> </ul>
MANAGE ANY USER	<p>次のことをユーザに許可します。</p> <ul style="list-style-type: none"> <li>• データベースユーザの作成、変更、削除、コメントの追加（初期パスワードの割り当てを含む）。</li> <li>• 任意のユーザの次回ログイン時でのパスワード変更の強制。</li> <li>• 任意のユーザのログインポリシーの割り当てとリセット。</li> <li>• 統合ログインと Kerberos ログインの作成、削除、コメントの追加。</li> <li>• 外部ログインの作成と削除。</li> </ul>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_SAMONITOR_ADMIN_ROLE</li> </ul>
SET USER	<p>別のユーザのロールおよび権限を一時的に仮定することをユーザに許可します。</p> <p>このシステム権限は、すべてのユーザに適用したり、特定のユーザセットあるいは1つ以上の指定ロールを付与されたユーザに限定したりできます。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_AUTH_SSO_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
変数		
CREATE DATABASE VARIABLE	<p>ユーザは、自分が所有するデータベーススコープ変数の作成、選択、更新および削除ができます。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
MANAGE ANY DATABASE VARIABLE	<p>ユーザは、自分または PUBLIC が所有するデータベーススコープ変数の作成および削除ができます。</p>	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>

オブジェクトまたは機能に応じたシステム権限	説明	ユーザ/ロールは次から継承可能
SELECT PUBLIC DATABASE VARIABLE	ユーザは、PUBLIC が所有するデータベーススコープ変数の値を選択することができます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
UPDATE PUBLIC DATABASE VARIABLE	ユーザは、自分が所有するデータベーススコープ変数を更新をすることができます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
<b>ビュー</b>		
ALTER ANY VIEW	ユーザは、任意のユーザが所有するビューに対して変更とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE ANY VIEW	ユーザは、任意のユーザが所有するビューの作成とコメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
CREATE VIEW	ユーザは、自己所有のビューを作成できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_RESOURCE_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
DROP ANY VIEW	ユーザは、任意のユーザが所有するビューを削除できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> </ul>
<b>Web サービス</b>		
MANAGE ANY WEB SERVICE	ユーザは Web サービスの作成、変更、削除、コメントの追加を実行できます。	<ul style="list-style-type: none"> <li>• SYS_AUTH_DBA_ROLE</li> <li>• SYS_AUTH_SA_ROLE</li> <li>• SYS_RUN_REPLICATION_ROLE</li> <li>• SYS_REPLICATION_ADMIN_ROLE</li> </ul>

## 関連情報

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

[アルファベット順のデータベースオプションリスト \[632 ページ\]](#)

### 1.9.1.2.3 ユーザ権限の評価方法

ロールが原因で、個々のユーザの権限が複雑になります。

たとえば、ユーザ M\_Haneef が、あるテーブルに対して SELECT と UPDATE の権限を所有しており、2 つのロールのメンバーでもあるとします。一方のロールではテーブルに対する権限をまったく持たず、もう一方のロールでは SELECT 権限だけを持つとします。この場合、このユーザの権限は一体どうなるでしょうか。

ユーザ ID にその作業を実行する適切な権限が明示的に付与されていれば、その作業は実行できます。

それ以外の場合、作業を実行する権限はメンバーが属する各ロールの権限によって異なります。いずれかのグループがその作業を実行する権限を持っていると、そのユーザ ID もロールのメンバーとして権限を持っていることになり、その作業を実行できます。

### 1.9.1.2.4 システム権限の付与 (SQL Central の場合)

システム権限を付与して、ユーザまたはロールに、データベースでの操作の実行を許可します。

#### 前提条件

付与するシステム権限に対して管理権限を持っている必要があります。

#### 手順

1. SQL Central で、*SQL Anywhere17* プラグインを使用してデータベースに接続します。
2. **システム権限** をダブルクリックし、システム権限を選択します。
3. 右ウィンドウ枠で、**[被付与者]** タブをクリックします。
4. ウィンドウ枠の任意の場所を右クリックして、**新規 ▶ 被付与者** を選択します。
5. システム権限を付与するユーザまたはロールを選択します。
6. **OK** をクリックします。

**システム権限** テーブルに新しいローが追加されます。

7. (省略可) **管理** を使用します。(管理権限) システム権限の管理権限を変更するためのカラム。カラムのチェックマークは、ユーザまたはロールがシステム権限を管理 (付与または取り消し) できることを意味します。空のカラムは、ユーザまたはロールがそのシステム権限に対する管理権限を持たないことを示します。オプションを切り替えるには、カラムをクリックします。  
(省略可) **発動** を使用します。(行使権限) ユーザまたはロールの行使権限を変更するためのカラム。カラムのチェックマークは、ユーザまたはロールがシステム権限を行使できることを意味します。行使権限はデフォルトで付与されます。オプションを切り替えるには、カラムをクリックします。
8. (省略可) CHANGE PASSWORD または SET USER システム権限を付与した場合、**[オプション]** カラムには **[すべてのユーザ]** が含まれます。**オプション** カラムはこれら 2 つのシステム権限にのみ適用されます。

- CHANGE PASSWORD システム権限を付与した場合、デフォルトでは、ユーザまたはロールは任意のユーザのパスワードを変更できます。変更できるパスワードを所有するユーザを制限するには、ローを右クリックし、[\[オプションの設定\]](#) をクリックします。
- SET USER システム権限を付与した場合、デフォルトでは、ユーザまたはロールは任意のユーザを同一化できます。同一化できるユーザを制限するには、ローを右クリックし、[\[オプションの設定\]](#) をクリックします。[オプションカラム](#) は CHANGE PASSWORD および SET USER システム権限にのみ適用されます。

9. [▶ ファイル ▶ 保存](#) をクリックします。

## 結果

ユーザまたはロールにシステム権限が付与されます。ロールにシステム権限を付与した場合、そのロールを付与されたユーザまたはロールはいずれも、新しいシステム権限を継承します。

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[GRANT 文の構文の変更 \[1582 ページ\]](#)

[オブジェクトレベル権限の付与 \(SQL Central の場合\) \[1523 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

### 1.9.1.2.5 システム権限の付与 (SQL の場合)

システム権限を付与して、ユーザまたはロールに、データベースでの操作の実行を許可します。

## 前提条件

付与するシステム権限に対して管理権限を持っている必要があります。

## 手順

1. データベースに接続します。
2. 次のような文を実行します。

オプション	アクション
管理権限を持たない権限を付与します。	<pre>GRANT privilege-name TO userid;</pre>
管理権限を持つ権限を付与します。	<pre>GRANT privilege-name TO userid WITH ADMIN OPTION;</pre>
管理権限のみの権限を付与します。	<pre>GRANT privilege-name TO userid WITH ADMIN ONLY OPTION;</pre>

## 結果

ユーザまたはロールにシステム権限が付与されます。ロールにシステム権限を付与した場合、そのロールを付与されたユーザまたはロールはいずれも、新しいシステム権限を継承します。

### 例

RoleA に管理権限を与えずに、CREATE ANY OBJECT システム権限をロール RoleA に付与するには、次の文を実行します。

```
GRANT CREATE ANY OBJECT TO RoleA
```

他のユーザやロールに対してシステム権限の付与や取り消しを実行する機能とともに、CREATE ANY OBJECT システム権限を RoleA に付与するには、次の文を実行します。

```
GRANT CREATE ANY OBJECT TO RoleA WITH ADMIN OPTION
```

RoleA 管理権限を BACKUP DATABASE システム権限に与えるが、BACKUP DATABASE 権限を使えないようにするには、次の分を実行します。

```
GRANT BACKUP DATABASE TO RoleA WITH ADMIN ONLY OPTION
```

RoleA 管理権限 (のみ) を BACKUP DATABASE および CHECKPOINT システム権限に付与するには、次の文を実行します。

```
GRANT BACKUP DATABASE, CHECKPOINT TO RoleA WITH ADMIN ONLY OPTION;
```

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[GRANT 文の構文の変更 \[1582 ページ\]](#)

[オブジェクトレベル権限の付与 \(SQL Central の場合\) \[1523 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

## 1.9.1.2.6 オブジェクトレベル権限の付与 (SQL Central の場合)

オブジェクトレベル権限を付与して、テーブル、ビュー、プロシージャ、シーケンス、または DB 領域などの特定のオブジェクトへのアクセスを許可します。

### 前提条件

オブジェクトレベル権限を付与するには、次のいずれかの条件に該当する必要があります。

- 権限を付与しているデータベースオブジェクトの所有者である。
- 権限を付与されており、管理権限がある。
- MANAGE ANY OBJECT PRIVILEGE システム権限がある。

### コンテキスト

権限は、ロールに付与すると、ロールのメンバーによって継承されます。

### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、[ユーザ](#) をダブルクリックして、ユーザを選択します。
3. 右ウィンドウ枠で、適切な権限のタブをクリックします。たとえば、テーブルへの権限を付与するには、[テーブル権限](#) タブをクリックします。

タブには、明示的にユーザに付与された権限が一覧表示されています。

4. ウィンドウ枠の任意の場所を右クリックして、**新規** > **権限** をクリックします。
5. 権限を付与するデータベースオブジェクトを選択し、**[OK]** をクリックします。

データベースオブジェクトはタブのローとして追加されました。デフォルトでは、すべてのデータベースオブジェクトの権限が付与されています。

6. (省略可) データベースオブジェクトでユーザに付与される権限を変更するには、該当する [権限](#) タブに表示されるテーブルの関連セルをクリックします (セル定義の凡例を参照)。

チェックマークにより、権限がユーザに付与されます。プラス記号 (+) 付きのチェックマークは、ユーザに権限を付与し、ユーザに他のユーザへの同じ権限を付与する権限を付与します。プラス記号 (+) 付きのチェックマークは、WITH GRANT OPTION 句付きの GRANT 文を実行することと同等です。

[テーブル権限](#) タブでテーブルを選択すると、対応する [カラム権限](#) テーブルが下に表示されます。カラム権限を変更すると、付与されている、または付与オプション付きで付与されている [カラム権限](#) アイコンが [テーブル権限](#) テーブルに表示されます。テーブル権限の変更は、特定の [カラム権限](#) を付与できるかどうかに影響します。

## → ヒント

- 変更を取り消すには、**権限** タブにある 1 つ以上のローを選択して、**編集 > 元に戻す** をクリックします。
- **凡例** をオンまたはオフにするには、**ファイル > 凡例を表示** をクリックします。

7. **ファイル > 保存** をクリックします。

## 結果

指定したオブジェクトレベル権限がユーザに付与されます。

## 関連情報

[オブジェクトレベル権限 \[1498 ページ\]](#)

[オブジェクトレベル権限の取り消し \(SQL Central の場合\) \[1528 ページ\]](#)

[オブジェクトレベル権限の付与 \(SQL の場合\) \[1524 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

[12.0.1 GRANT statement \(12.0.1 GRANT 文\)](#)

## 1.9.1.2.7 オブジェクトレベル権限の付与 (SQL の場合)

オブジェクトレベル権限を付与して、テーブル、ビュー、プロシージャ、シーケンス、または DB 領域などの特定のオブジェクトへのアクセスを許可します。

## 前提条件

オブジェクトレベル権限を付与するには、次のいずれかの条件に該当する必要があります。

- 権限を付与しているデータベースオブジェクトの所有者である。
- 権限を付与されており、管理権限がある。
- MANAGE ANY OBJECT PRIVILEGE システム権限がある。

## コンテキスト

権限は、ロールに付与すると、ロールのメンバーによって継承されます。

## 手順

1. データベースに接続します。
2. GRANT 文を実行します。

## 結果

指定したオブジェクトレベル権限がユーザに付与されます。

### 例

次の文では、myTable というテーブルに対する SELECT 権限がユーザ AnnW に付与されます。

```
GRANT SELECT ON myTable TO AnnW;
```

次の文では、myTable というテーブルに対する SELECT 権限がユーザ AnnW に付与され、その権限の管理権限が与えられます。

```
GRANT SELECT ON myTable TO AnnW WITH GRANT OPTION;
```

## 関連情報

[オブジェクトレベル権限 \[1498 ページ\]](#)

[オブジェクトレベル権限の付与 \(SQL Central の場合\) \[1523 ページ\]](#)

[オブジェクトレベル権限の取り消し \(SQL の場合\) \[1529 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

### 1.9.1.2.8 システム権限の取り消し (SQL Central の場合)

ユーザ、ユーザ拡張ロール、またはユーザ定義ロールからシステム権限を取り消します。

## 前提条件

システム権限の管理権限が必要です。

SYS ロールから権限を取り消すことはできません。その他のシステムロールからは、そのデフォルト権限でない場合に限り、権限を取り消すことができます。

## コンテキスト

権限の行使権限と管理権限がユーザに付与されている場合は、管理権限、または行使権限と管理権限の両方を取り消すことができます。行使権限 (のみ) を取り消して管理権限を残すことはできません。これを行うには、システム権限全体を取り消して、管理権限のみのシステム権限を再度付与します。

互換ロールからシステム権限を取り消すことはできません。ただし、互換ロールをユーザ定義ロールに移行してから、ユーザ定義ロールから権限を取り消すことはできます。

## 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で **[システム権限]** をクリックし、さらにシステム権限をクリックします。
3. **被付与者** タブをクリックします。
4. 次のうちの 1 つを選択してください。

オプション	アクション
管理権限をはじめとした権限の取り消し	ユーザまたはロールを右クリックして、 <b>[削除]</b> をクリックします。
管理権限のみの取り消し	<b>管理</b> をクリックします。(管理権限) チェックボックスをオフにするカラム。
権限のみの取り消し	<b>発動</b> をクリックします。(行使権限) チェックボックスをオフにするカラム。

### ➔ ヒント

- 変更を元に戻すには、1 つ以上のローを選択して、**▶ 編集 ▶ 元に戻す** をクリックします。
- **凡例** をオンまたはオフにするには、**▶ ファイル ▶ 凡例を表示** をクリックします。

5. **▶ ファイル ▶ 保存** をクリックします。

## 結果

システム権限、管理権限、またはその両方が、ユーザまたはロールから取り消されます。

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[互換ロールをユーザ定義ロールに移行する \(SQL Central の場合\) \[1462 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

## 1.9.1.2.9 システム権限の取り消し (SQL の場合)

ユーザ、ユーザ拡張ロール、またはユーザ定義ロールからシステム権限を取り消します。

### 前提条件

システム権限の管理権限が必要です。

SYS ロールから権限を取り消すことはできません。その他のシステムロールからは、そのデフォルト権限でない場合に限り、権限を取り消すことができます。

### コンテキスト

権限の行使権限と管理権限がユーザに付与されている場合は、管理権限、または行使権限と管理権限の両方を取り消すことができます。行使権限 (のみ) を取り消して管理権限を残すことはできません。これを行うには、システム権限全体を取り消して、管理権限のみのシステム権限を再度付与します。

互換ロールからシステム権限を取り消すことはできません。ただし、互換ロールをユーザ定義ロールに移行してから、ユーザ定義ロールから権限を取り消すことはできます。

### 手順

1. データベースに接続します。
2. 次のような REVOKE 文を実行します。

```
REVOKE privilege FROM grantee;
```

### 結果

システム権限、管理権限、またはその両方が、ユーザまたはロールから取り消されます。

### 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[互換ロールをユーザ定義ロールに移行する \(SQL Central の場合\) \[1462 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

## 1.9.1.2.10 オブジェクトレベル権限の取り消し (SQL Central の場合)

ユーザまたはロールに付与されているオブジェクトレベル権限を取り消して、テーブル、ビュー、プロシージャ、シーケンス、または DB 領域などの特定のオブジェクトへのアクセスを制限します。

### 前提条件

オブジェクトレベル権限を取り消すには、次のいずれかの条件に該当する必要があります。

- 権限を取り消しているデータベースオブジェクトの所有者である。
- 権限を付与されており、管理権限がある。
- MANAGE ANY OBJECT PRIVILEGE システム権限がある。

### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で、ユーザ権限を取り消すデータベースオブジェクトをクリックするか、[\[ユーザ\]](#) または [\[ロール\]](#) をクリックしてから権限を取り消すユーザまたはロールをクリックします。

オプション	アクション
ユーザまたはロールごとのオブジェクトレベル権限の取り消し	<ol style="list-style-type: none"><li>1. 右ウィンドウ枠で、権限を取り消すデータベースオブジェクトに対応する権限のタブをクリックします。たとえば、テーブルへの権限を取り消すには、<a href="#">[テーブル権限]</a> タブをクリックします。</li><li>2. 権限を右クリックし、<a href="#">[取り消し]</a> をクリックしてから、<a href="#">[すべての権限]</a> をクリックするか、取り消す特定の権限をクリックします。</li></ol>
データベースオブジェクトごとのオブジェクトレベル権限の取り消し	<ol style="list-style-type: none"><li>1. 右ウィンドウ枠で、<a href="#">[権限]</a> タブをクリックします。</li><li>2. ユーザまたはロールを右クリックし、<a href="#">[取り消し]</a> をクリックしてから、<a href="#">[すべての権限]</a> をクリックするか、取り消す特定の権限をクリックします。</li></ol>

3. [▶ ファイル ▶ 保存 ▶](#) をクリックします。

### 結果

権限は、指定されたユーザまたはロールから取り消されます。

### 関連情報

[オブジェクトレベル権限 \[1498 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

## 1.9.1.2.11 オブジェクトレベル権限の取り消し (SQL の場合)

ユーザまたはロールに付与されているオブジェクトレベル権限を取り消して、テーブル、ビュー、プロシージャ、シーケンス、または DB 領域などの特定のオブジェクトへのアクセスを制限します。

### 前提条件

オブジェクトレベル権限を取り消すには、次のいずれかの条件に該当する必要があります。

- 権限を取り消しているデータベースオブジェクトの所有者である。
- 権限を付与されており、管理権限がある。
- MANAGE ANY OBJECT PRIVILEGE システム権限がある。

### 手順

1. データベースに接続します。
2. 次のような REVOKE 文を実行します。

```
REVOKE privilege-name ON object-name FROM userid;
```

### 結果

権限は、指定されたユーザまたはロールから取り消されます。

### 関連情報

[オブジェクトレベル権限 \[1498 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

### 1.9.1.3 ユーザ

すべての新規ユーザには、自動的に PUBLIC システムロールが付与されます。

PUBLIC ロールを含む権限を編集することで、新規ユーザに自動的に付与される権限を制御できます。

デフォルトで新しいユーザに割り当てられる権限には以下が含まれます。

- データベースへの接続 (ユーザのパスワードが指定されていることが前提)
- システムビューに格納されているデータの表示
- 大部分のシステムストアプロシージャの実行

新しいユーザを作成すると、次を実行できます。

- ユーザへシステムロールおよびシステム権限の付与します。
- ユーザのオブジェクトレベル権限を設定します。
- そのユーザにログインポリシーを割り当てます。デフォルトでは、ユーザはルートログインポリシーに割り当てられます。
- SQL Remote システムで使用するために、パブリッシャとしてまたはデータベースのリモートユーザとしてユーザを設定します。

このセクションの内容:

#### [ユーザの作成 \(SQL Central の場合\) \[1531 ページ\]](#)

SQL Central のユーザ作成ウィザードを使用してユーザを作成します。

#### [ユーザの作成 \(SQL の場合\) \[1532 ページ\]](#)

CREATE USER 文を使用して、ユーザを作成します。

#### [ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

ユーザおよびロールに付与されたロールと権限を表示します。

#### [ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

継承しているロールを含むロールと権限を表示します。

#### [ユーザの削除 \(SQL Central の場合\) \[1535 ページ\]](#)

ユーザを削除し、ユーザが所有するすべてのデータベースオブジェクト (テーブルなど) とユーザの外部ログインも削除します。

## 関連情報

[スーパユーザ \[1484 ページ\]](#)

## 1.9.1.3.1 ユーザの作成 (SQL Central の場合)

SQL Central の[ユーザ作成ウィザード](#)を使用してユーザを作成します。

### 前提条件

MANAGE ANY USER システム権限が必要です。

### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. 左ウィンドウ枠で [ユーザ](#) を右クリックし、**▶ 新規 ▶ ユーザ ▶** をクリックします。
3. ウィザードの指示に従います。

### 結果

ユーザが作成されます。

### 次のステップ

ロールまたは権限をユーザに付与します。

### 関連情報

[パスワードおよびユーザ ID の制限事項と考慮事項 \[1554 ページ\]](#)

[ログインポリシー \[613 ページ\]](#)

[ユーザをユーザ拡張ロールに変換 \(SQL Central の場合\) \[1476 ページ\]](#)

[ロールの付与 \(SQL Central の場合\) \[1489 ページ\]](#)

[接続されたユーザの管理 \(SQL Central\) \[15 ページ\]](#)

## 1.9.1.3.2 ユーザの作成 (SQL の場合)

CREATE USER 文を使用して、ユーザを作成します。

### 前提条件

MANAGE ANY USER システム権限が必要です。

### 手順

1. データベースに接続します。
2. IDENTIFIED BY 句を含む CREATE USER 文を実行します。

デフォルトでは、ユーザにルートログインポリシーが割り当てられています。ユーザに別のログインポリシーを割り当てる必要がある場合は、CREATE USER 文の LOGIN POLICY 句を使用して指定します。

### 結果

ユーザが作成され、指定したログインポリシーがそのユーザに割り当てられます。ログインポリシーが指定されない場合、ルートログインポリシーが適用されます。

#### 例

次の例は、ユーザ ID M\_Haneef とパスワード Welcome を使用して、データベースに新しいユーザを追加します。

```
CREATE USER M_Haneef  
IDENTIFIED BY Welcome;
```

### 次のステップ

ロールまたは権限をユーザに付与します。

### 関連情報

[パスワードおよびユーザ ID の制限事項と考慮事項 \[1554 ページ\]](#)

[ログインポリシー \[613 ページ\]](#)

### 1.9.1.3.3 ユーザまたはロールのロールと権限を表示する (SQL Central の場合)

ユーザおよびロールに付与されたロールと権限を表示します。

#### 前提条件

PUBLIC ロールが必要です。

#### 手順

1. SQL Central で、[SQL Anywhere17 プラグイン](#)を使用してデータベースに接続します。
2. 次のオプションのうちの 1 つを選択してください。

オプション	アクション
ユーザのロールと権限を表示	<ol style="list-style-type: none"><li>1. 左ウィンドウ枠で、<b>ユーザ</b>をダブルクリックします。</li><li>2. 表示したいユーザをクリックします。</li><li>3. ユーザのロールとシステム権限を表示するには、右ウィンドウ枠で、<b>ロールとシステム権限</b>をそれぞれクリックします。</li><li>4. ユーザのオブジェクトレベル権限を表示するには、残っている権限タブのいずれか (<b>テーブル権限</b>、<b>表示権限</b>など)をクリックします。</li></ol>
ロールのロールと権限を表示	<ol style="list-style-type: none"><li>1. 左ウィンドウ枠で、<b>[ロール]</b>をダブルクリックします。</li><li>2. 表示したいロールをクリックします。</li><li>3. ロールとシステム権限を表示するには、右ウィンドウ枠で、<b>ロールとシステム権限</b>をそれぞれクリックします。</li><li>4. ロールのオブジェクトレベル権限を表示するには、残っている権限タブのいずれか (<b>テーブル権限</b>、<b>表示権限</b>など)をクリックします。</li></ol>

3. 必要に応じて、**継承の表示**を選択して、継承したロールと権限を含めます。

#### 結果

右ウィンドウ枠に、指定したユーザまたはロールに付与された、あるいは指定したユーザまたはロールが継承したロールと権限が表示されます。

## 関連情報

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

### 1.9.1.3.4 ユーザまたはロールのロールと権限を表示する (SQL の場合)

継承しているロールを含むロールと権限を表示します。

#### 前提条件

- `sp_displayroles` システムプロシージャこのプロシージャを自分自身に実行する場合、権限は不要です。ただし、別のユーザ ID またはロールのシステム権限またはロールを返すには、`MANAGE ROLES` システム権限が必要です。
- `sp_objectpermission` システム権限このプロシージャを、自分自身または自分が所有するオブジェクトに実行する場合、権限は不要です。ただし、このプロシージャを別のユーザ ID、または別のユーザ ID が所有するオブジェクトに対して呼び出す場合は、`MANAGE ANY OBJECT PRIVILEGE` システム権限が必要です。

#### コンテキスト

また、このタスクを利用して、指定したロールのロールと権限も表示できます。

#### 手順

1. データベースに接続します。
2. ユーザの持つロールおよびシステム権限を表示するには、`sp_displayroles` システムプロシージャを呼び出す次のような文を実行します。ここで、`userid` はユーザのユーザ ID です。

```
SELECT * FROM sp_displayroles( 'userid', 'expand_down');
```

3. ユーザの持つオブジェクトレベル権限を表示するには、`sp_objectpermission` を呼び出す次のような文を実行します。

```
SELECT * FROM sp_objectpermission('userid');
```

#### 結果

`sp_displayroles` の結果の `role_name` カラムには、ユーザに明示的に付与されたロールやシステム権限だけでなく、継承したのも含まれます。ロールやシステム権限が別のロールから継承されている場合、そのロールの名前は

parent\_role\_name カラムに表示されます。grant\_type カラムでは、ロールまたはシステム権限に対してユーザが管理権限を持っているかがわかります。role\_level カラムは、別のロールのメンバーであるロールのメンバーになることで継承が発生することがあるため、継承の階層を伝えます。このことは、ユーザからロールまたは権限を取り消そうとしても、そのユーザが依然としてロールや権限を使用できることがわかったときのトラブルシューティングに役立ちます。

sp\_objectpermission の結果には、ユーザに明示的に付与された権限だけでなく、継承されたオブジェクトレベル権限も含まれます。被付与者カラムを使用して、オブジェクトレベル権限の継承元を理解します。付与者カラムでは、実際の付与を行った人がわかります。付与可能カラムでは、権限に対してユーザが管理権限を持っているかがわかります。

## 関連情報

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

### 1.9.1.3.5 ユーザの削除 (SQL Central の場合)

ユーザを削除し、ユーザが所有するすべてのデータベースオブジェクト (テーブルなど) とユーザの外部ログインも削除します。

## 前提条件

MANAGE ANY USER システム権限が必要です。

この手順の実行中は、対象ユーザはデータベースに接続できません。

## 手順

1. SQL Central で、[SQL Anywhere17 プラグイン](#)を使用してデータベースに接続します。
2. 左ウィンドウ枠でユーザをクリックします。
3. 右ウィンドウ枠でユーザを右クリックし、[\[削除\]](#) をクリックします。
4. [はい](#) をクリックして、操作を確認します。

## 結果

ユーザが削除されるだけでなく、ユーザが所有するすべてのデータベースオブジェクト (テーブルやプロシージャなど) や、そのユーザの外部ログインも削除されます。ユーザがユーザ拡張ロールだった場合、このロールはすべてのユーザおよびユーザ拡張ロールが付与されたロールから取り消されます。さらに、このユーザが、いずれかのサービスの USER 句で指定されている場合は、それらのサービスも削除されます。

## 関連情報

[ユーザ拡張ロール \[1475 ページ\]](#)

### 1.9.1.4 グループ

グループとは、グループ内の他のすべてのユーザに共通のロールおよび権限のセットを所有するユーザのセットです。

グループは、ユーザセットのロールおよび権限を維持するための効率的な方法です。

また、オブジェクトを所有するグループでのメンバーシップは、グループが所有するプロシージャを実行したり、グループが所有するテーブルにクエリしたりといった操作を実行するときに、ユーザがオブジェクト名を修飾する必要がないことを意味します。たとえば、MyData という名前のテーブルが PersonnelData という名前のグループに所有されていて、ユーザ ID M\_Haneef がこのグループのメンバーであるとき、M\_Haneef はクエリにおいてテーブルを PersonnelData.MyData ではなく MyData として参照することができます。ユーザがグループ所有のテーブルと同じ名前のテーブルを所有している場合、データベースサーバはグループが所有するテーブルではなくユーザが所有するテーブルを使用します。ユーザが同じ名前のテーブルを所有する複数のグループに属している場合、そのユーザはテーブル名を修飾する必要があります。

グループはデータベースオブジェクトではありません。代わりに、グループは次の 2 つのアプローチのいずれかにより、ユーザ定義ロールを使用して実現されます。

#### グループ作成のためのユーザ拡張ロールアプローチ

ユーザ拡張ロールは、既存のユーザ ID がロールになるように拡張することで作成されたユーザ定義ロールです。このグループ作成のためのアプローチでは、ユーザ ID が作成され、その後グループに必要なすべてのロールと権限が付与されます。ユーザ ID はユーザ拡張ロールに変換され、さらに必要とするユーザに付与されます。これらのステップが発生する順序は、当然ながら異なる可能性があります。

ロールに拡張されたユーザもログイン機能を持つため、このアプローチは、レプリケーション目的のグループを作成するときに共通です。

SQL Anywhere の以前のバージョンのユーザの場合、このアプローチは、ユーザに GROUP を付与し、さらに他のユーザに MEMBERSHIP IN GROUP を付与する廃止予定の GRANT 文アプローチと同じです。

このアプローチを使用して作成したグループを削除するには、ユーザ拡張ロールを通常のユーザに戻します (DROP ROLE FROM USER *userid*)。オブジェクトの所有権は、通常のユーザに戻されるユーザに留まります。

ユーザ拡張ロールアプローチを使用するとき、グループおよびそのメンバーの情報は、SYSROLEGRANTS 統合ビューまたは SYSGROUPS 互換ビューからクエリされる可能性があります。

#### グループ作成のためのスタンドアロンロールアプローチ

スタンドアロンのロールは、どのユーザ ID にも関連付けられないユーザ定義のロールです。このグループ作成のためのアプローチでは、ロールが作成され、その後定義されたユーザのセットに必要なすべてのロールと権限が付与されます。さらに、ロールは必要とするユーザに付与されます。

スタンドアロンアプローチを使用して作成されたグループを削除するには、ロールを削除します。オプションで、ロールを削除するときに、同時にロールが所有するオブジェクトも削除できます。

スタンドアロンロールアプローチが使用されているとき、グループとそのメンバー情報に SYSGROUPS 互換ビューからクエリすることはできません。代わりに、SYSROLEGRANTS 統合ビューからクエリできます。

いずれのアプローチの場合も、グループメンバーの権限の管理は、各ユーザ ID のレベルではなく、ユーザ定義ロールレベルで行われます。

#### 例

次の SQL 文では、SELECT ANY TABLE システム権限と SA\_DEBUG システムロールを必要とする 3 人のメンバーで構成されるグループのきわめて単純な例を示しています。最初の文のセットでは、ユーザ拡張アプローチを示します。2 番目の文のセットでは、スタンドアロンロールアプローチを示します。

ユーザ拡張アプローチの場合、1 人のメンバー (member1) はロールとメンバーの両方として動作していますが、いずれのアプローチにも 3 人のメンバーが存在すると言えます。

```
CREATE USER member1 IDENTIFIED BY sql;
CREATE USER member2 IDENTIFIED BY sql;
CREATE USER member3 IDENTIFIED BY sql;
CREATE ROLE FOR USER member1;
GRANT ROLE member1 TO member2, member3;
GRANT SELECT ANY TABLE TO member1;
GRANT ROLE SA_DEBUG TO member1;
```

```
CREATE ROLE standaloneRole;
CREATE USER member1 IDENTIFIED BY sql;
CREATE USER member2 IDENTIFIED BY sql;
CREATE USER member3 IDENTIFIED BY sql;
GRANT ROLE standaloneRole TO member1, member2, member3;
GRANT SELECT ANY TABLE TO standaloneRole;
GRANT ROLE SA_DEBUG TO standaloneRole;
```

## 関連情報

[ユーザ定義ロール \[1470 ページ\]](#)

[グループはロールを使用して実現される \[1580 ページ\]](#)

[ユーザ定義ロールの作成 \(SQL の場合\) \[1474 ページ\]](#)

[ユーザ定義ロールの作成 \(SQL Central の場合\) \[1472 ページ\]](#)

[ユーザをユーザ拡張ロールに変換する \(SQL の場合\) \[1477 ページ\]](#)

[ユーザをユーザ拡張ロールに変換 \(SQL Central の場合\) \[1476 ページ\]](#)

[ロールの付与 \(SQL Central の場合\) \[1489 ページ\]](#)

[ロールの付与 \(SQL の場合\) \[1490 ページ\]](#)

### 1.9.1.5 ロールと権限の継承

ロールまたはユーザが持つ権限は、複数のカテゴリにグループ化できます。

ユーザまたはグループ (ロール) に対して明示的に付与された権限

これらは、ユーザまたはロールが指定のデータベースオブジェクトを作成、実行、削除、変更、使用できるかどうかを制御するために明示的に設定された権限です。

明示的に付与されたものではない権限を取り消すことはできません。たとえば、ユーザ BobW が Sales ロールのメンバーであるとして。ユーザがテーブル T の DELETE 権限を Sales ロールに付与する場合、BobW はテーブル T からローを削除できます。DELETE ON T 権限は BobW に明示的に付与されたものではないため、BobW がテーブル T から削除を行わないようにするために BobW からテーブル T での DELETE 権限を取り消すことはできません。

#### オブジェクトを所有することで取得した権限

データベース中で新しくオブジェクトを作成したユーザは、そのオブジェクトの所有者と呼ばれます。所有者には、そのオブジェクトに対してすべての操作を行うパーミッションが自動的に与えられます。たとえば、テーブルの所有者はそのテーブルの構造を変更したり、他のユーザにテーブルのデータを更新するパーミッションを与えたりすることができます。

ユーザがユーザ拡張ロールである場合、そのロールの被付与者は、オブジェクトの所有権に関連付けられた権限は継承しません。権限は明示的に付与する必要があり、明示的に付与されていない場合、ロールを介して継承されることはありません。

#### ロールを介して継承される権限

各ユーザ ID は、1 つ以上のロールに付与できます。被付与者は、そのロールに関連付けられた権限を継承します。

オブジェクトレベルの権限では、データベースオブジェクトの所有権は個別のユーザ ID に関連付けられます。所有者がユーザ拡張ロールまたはスタンドアロンロールであるとき、データベースオブジェクトの所有権はその被付与者には継承されません。たとえば、ユーザ拡張ロールまたはスタンドアロンロールがオブジェクトを所有している場合、ユーザ拡張ロールまたはスタンドアロンロールの被付与者は、自動的にオブジェクトを照会することはできません。オブジェクトへのクエリには、引き続き SELECT 権限が付与されている必要があります。

#### グループを介して継承される権限

グループはユーザ定義ロールで実現されるので、グループメンバーシップを通じた権限の継承は、ロールを介した権限の継承と同じです。

#### 無効なオブジェクトに付与される権限

無効にされたオブジェクトに対する権限を付与できます。無効化されたオブジェクトに対する権限はデータベースに保存され、オブジェクトが有効になると使用できるようになります。ロールがロールに付与されている場合、継承階層は多くのレベルが存在することがあります。

このセクションの内容:

#### [ロールの継承 \[1539 ページ\]](#)

デフォルトでは、システム権限とロールは、ロールの被付与者によって継承されます。

#### [継承のシナリオ \[1539 ページ\]](#)

特に管理権限のさまざまな付与レベルに関して、権限がどのように継承されるのかを示すために複数のシナリオがあります。

## 関連情報

#### [ユーザ権限の評価方法 \[1520 ページ\]](#)

#### [オブジェクトレベル権限の付与 \(SQL Central の場合\) \[1523 ページ\]](#)

#### [ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

#### [ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

## 1.9.1.5.1 ロールの継承

デフォルトでは、システム権限とロールは、ロールの被付与者によって継承されます。

ただし、16.0 より前のデータベースとの下位互換性のため、以前は継承不可権限であったこれらのロールの継承をブロックできるようにする必要がありました。これらのロールを次に示します。

- SYS\_AUTH\_DBA\_ROLE (以前は、DBA 権限)
- SYS\_AUTH\_RESOURCE\_ROLE (以前は、RESOURCE 権限)
- SYS\_AUTH\_BACKUP\_ROLE (以前は、BACKUP 権限)
- SYS\_AUTH\_VALIDATE\_ROLE (以前は、VALIDATE 権限)
- SYS\_RUN\_REPLICATION\_ROLE (以前は、REMOTE DBA 権限)

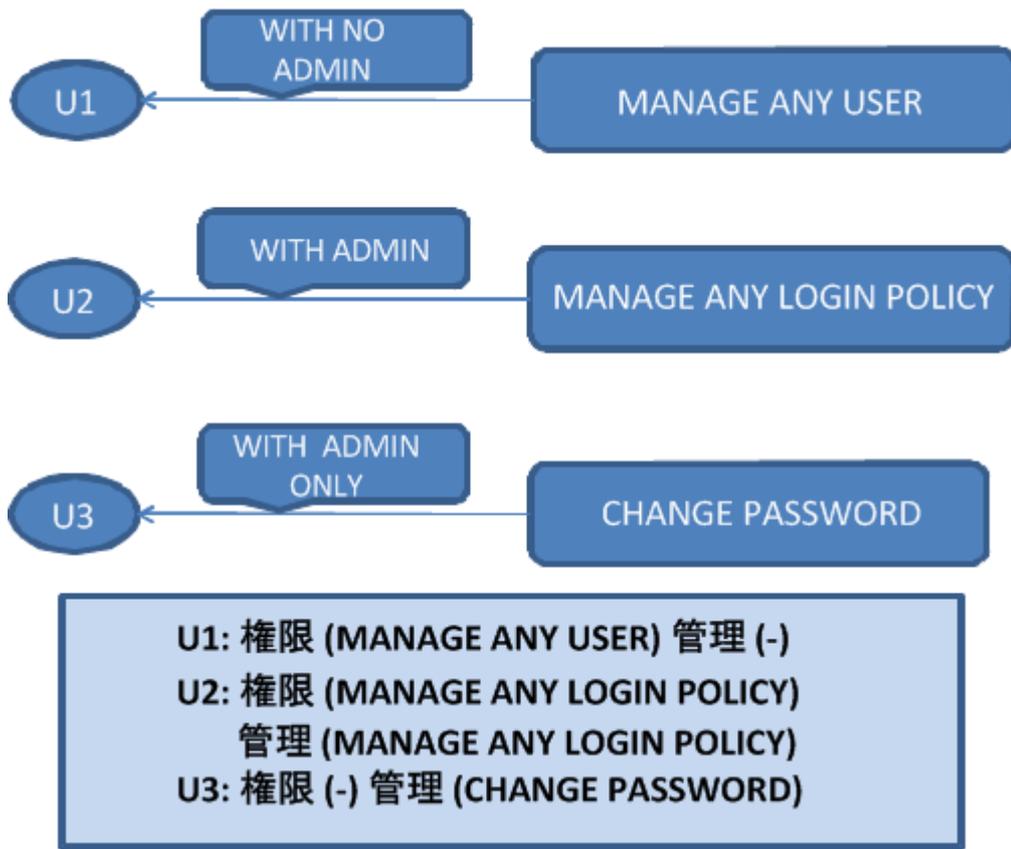
### 関連情報

[ロールとなる一部の権限の継承動作の変更 \[1592 ページ\]](#)

## 1.9.1.5.2 継承のシナリオ

特に管理権限のさまざまな付与レベルに関して、権限がどのように継承されるのかを示すために複数のシナリオがあります。

### シナリオ 1



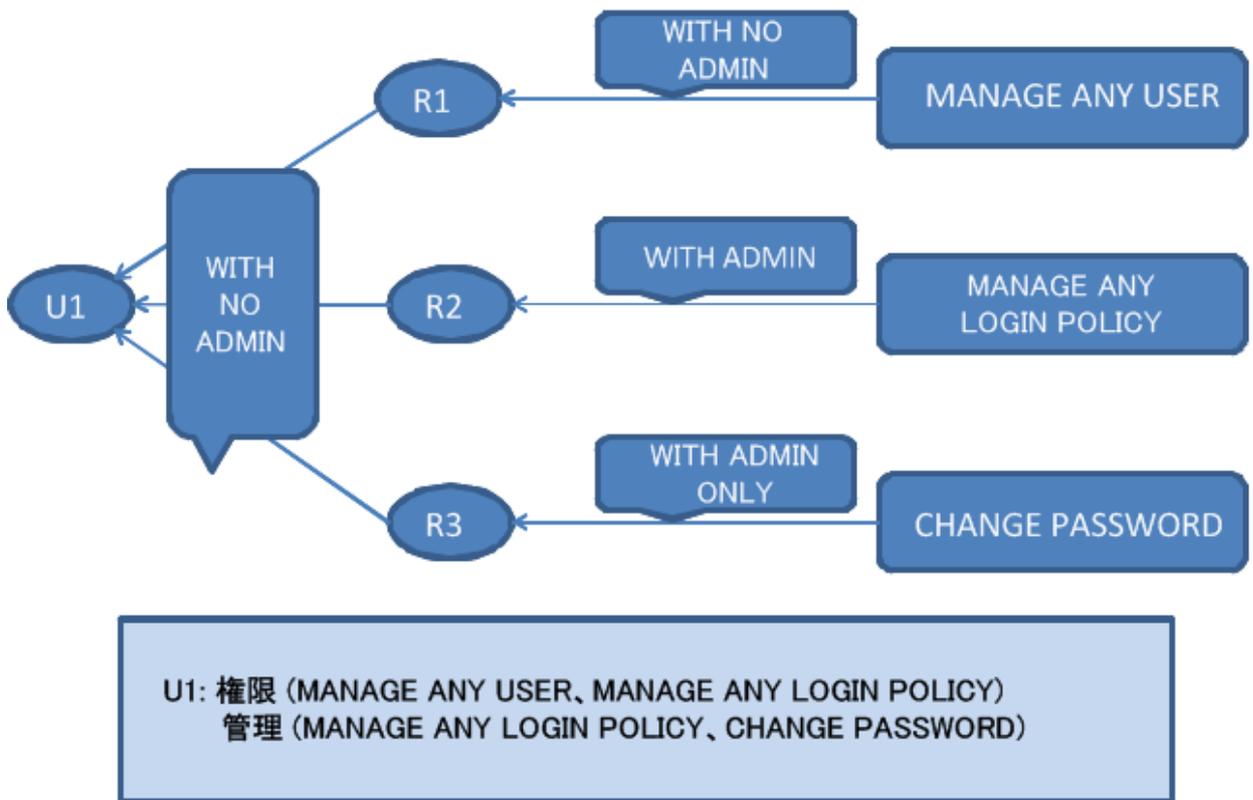
この図は、次のことを示しています。

- ユーザ 1 には、管理権限を含まない MANAGE ANY USER システム権限が付与されました。
- ユーザ 2 には、管理権限を含む MANAGE ANY LOGIN POLICY システム権限が付与されました。
- ユーザ 3 には、管理権限のみを含む CHANGE PASSWORD システム権限が付与されました。

このことから、ユーザ 1、ユーザ 2、およびユーザ 3 には次の権限があります。

	ユーザ 1	ユーザ 2	ユーザ 3
MANAGE ANY USER	行使	-	-
MANAGE ANY LOGIN POLICY	-	行使、管理	-
CHANGE PASSWORD	-	-	管理

## シナリオ 2



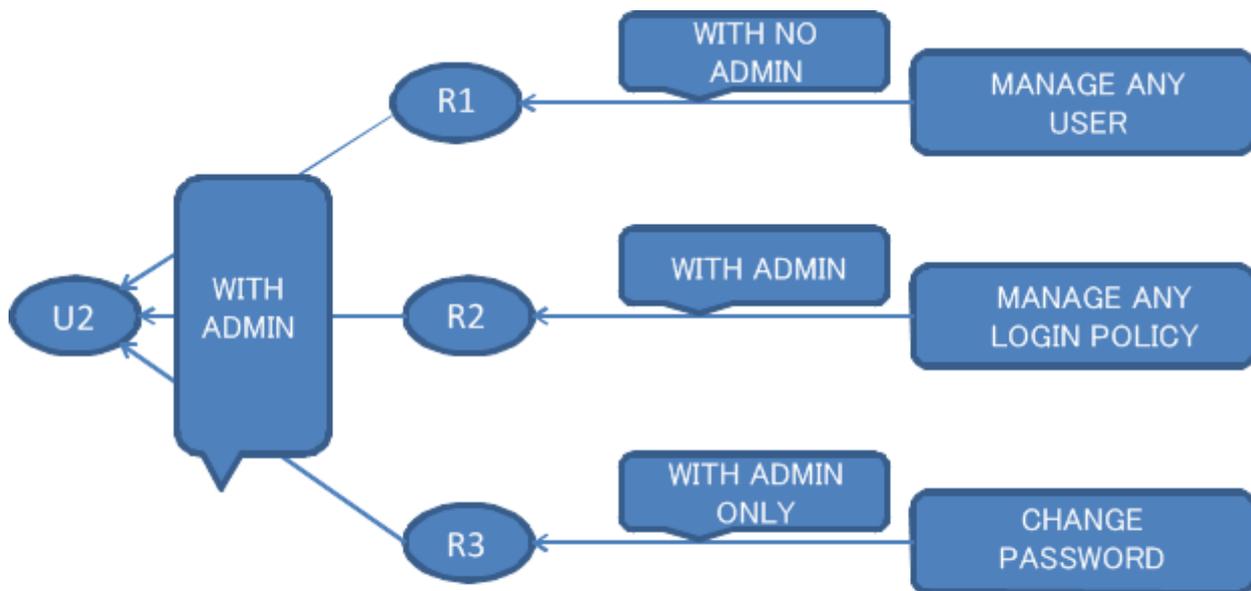
この図は、次のことを示しています。

- ロール 1 には、管理権限を含まない MANAGE ANY USER システム権限が付与されました。
- ロール 2 には、管理権限を含む MANAGE ANY LOGIN POLICY システム権限が付与されました。
- ロール 3 には、管理権限のみを含む CHANGE PASSWORD システム権限が付与されました。
- ユーザ 1 には、管理権限なしのロール 1、ロール 2、ロール 3 が付与されました。

ユーザ 1 には、次の権限があります。

ロール/権限	ユーザ 1
MANAGE ANY USER	行使
MANAGE ANY LOGIN POLICY	行使、管理
CHANGE PASSWORD	管理
R1	-
R2	-
R3	-

### シナリオ 3



**U2: 権限 (MANAGE ANY USER、MANAGE ANY LOGIN POLICY  
管理 (R1、R2、R3、MANAGE ANY LOGIN POLICY、CHANGE PASSWORD)**

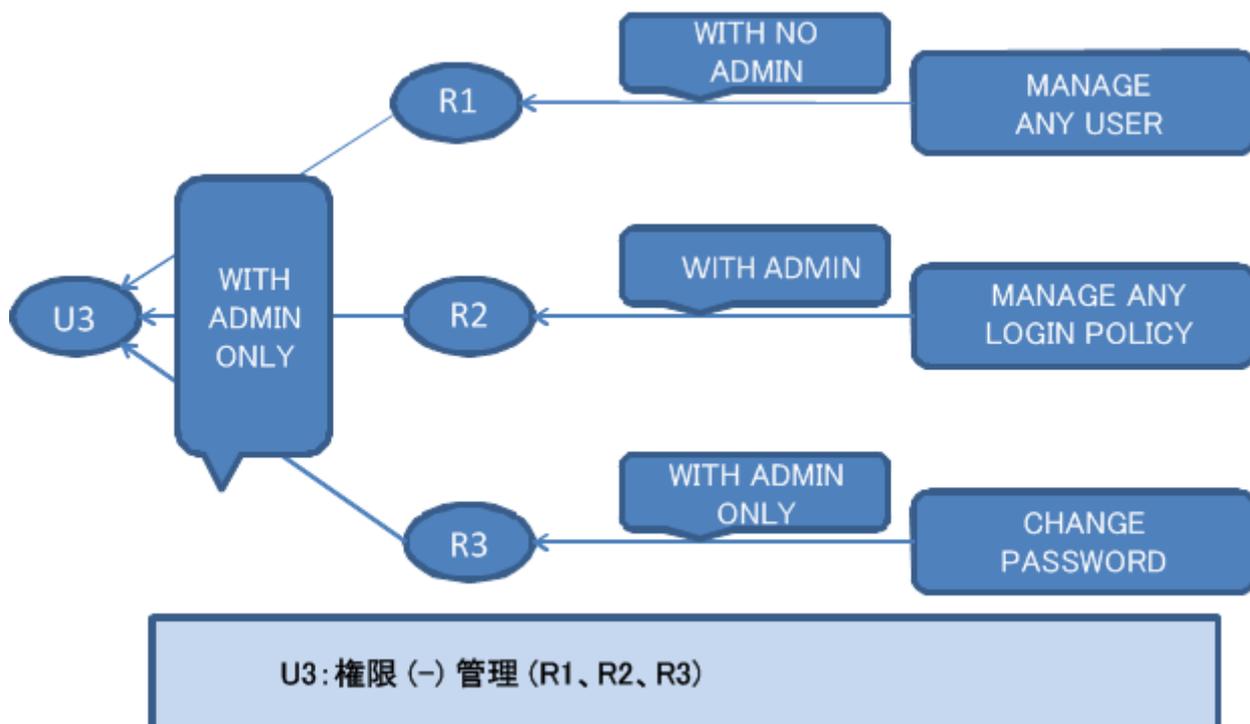
この図は、次のことを示しています。

- ロール 1 には、管理権限を含まない MANAGE ANY USER システム権限が付与されました。
- ロール 2 には、管理権限を含む MANAGE ANY LOGIN POLICY システム権限が付与されました。
- ロール 3 には、管理権限のみを含む CHANGE PASSWORD システム権限が付与されました。
- ユーザ 2 には、管理権限を含むロール 1、ロール 2、ロール 3 が付与されました。

ユーザ 2 には、次の権限があります。

ロール/権限	ユーザ 2
MANAGE ANY USER	行使
MANAGE ANY LOGIN POLICY	行使、管理
CHANGE PASSWORD	管理
R1	管理
R2	管理
R3	管理

## シナリオ 4



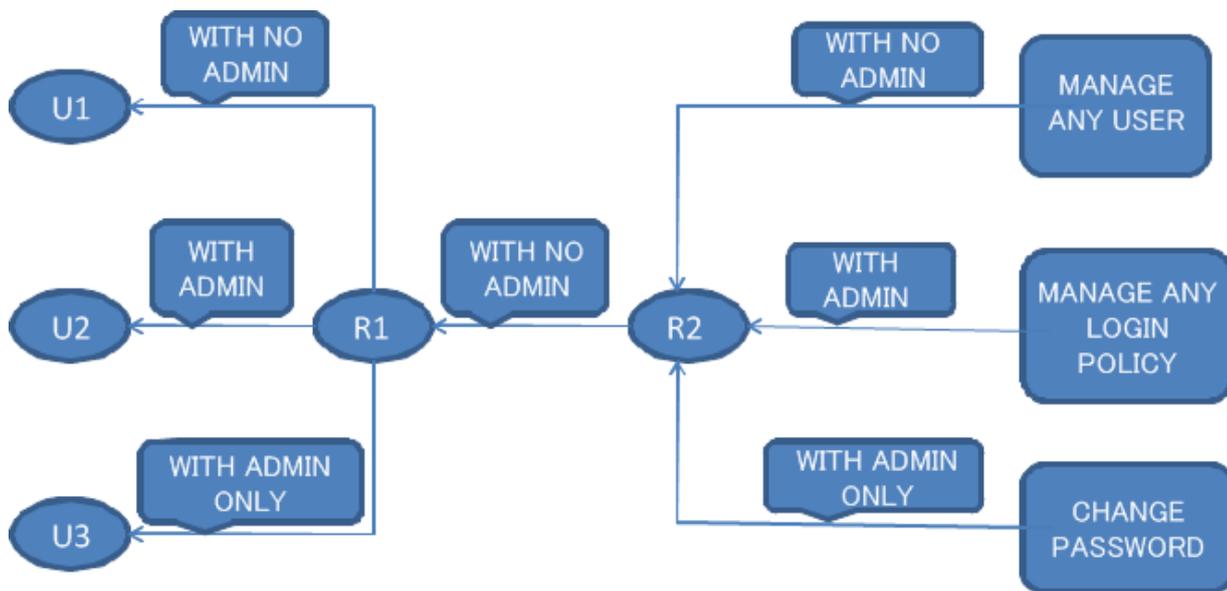
この図は、次のことを示しています。

- ロール 1 には、管理権限を含まない MANAGE ANY USER システム権限が付与されました。
- ロール 2 には、管理権限を含む MANAGE ANY LOGIN POLICY システム権限が付与されました。
- ロール 3 には、管理権限のみを含む CHANGE PASSWORD システム権限が付与されました。
- ユーザ 3 には、管理権限のみを含むロール 1、ロール 2、ロール 3 が付与されました。

ユーザ 3 には、次の権限があります。

ロール/権限	ユーザ 3
MANAGE ANY USER	-
MANAGE ANY LOGIN POLICY	-
CHANGE PASSWORD	-
R1	管理
R2	管理
R3	管理

## シナリオ 5



**U1: 権限 (MANAGE ANY USER、MANAGE ANY LOGIN POLICY) 管理 (MANAGE ANY LOGIN POLICY、CHANGE PASSWORD)**  
**U2: 権限 (MANAGE ANY USER、MANAGE ANY LOGIN POLICY) 管理 (MANAGE ANY LOGIN POLICY、CHANGE PASSWORD)**  
**U3: 権限 (-) 管理 (R1)**

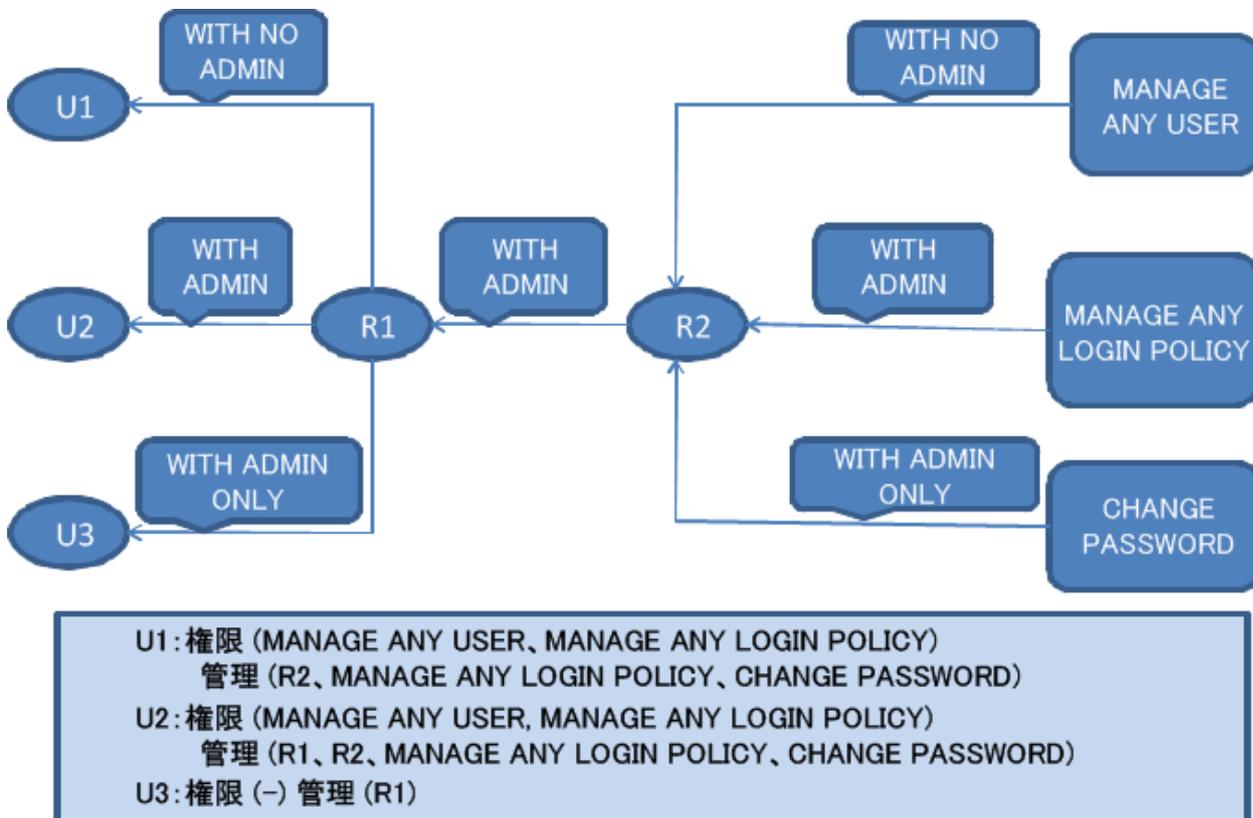
この図は、次のことを示しています。

- ロール 2 には、管理権限を含まない MANAGE ANY USER システム権限が付与されました。
- ロール 2 には、管理権限を含む MANAGE ANY LOGIN POLICY システム権限が付与されました。
- ロール 2 には、管理権限のみを含む CHANGE PASSWORD システム権限が付与されました。
- ロール 1 は、管理権限を含まないロール 2 を付与されました。
- ユーザ 1 は、管理権限を含まないロール 1 を付与されました。
- ユーザ 2 は、管理権限を含むロール 1 を付与されました。
- ユーザ 3 は、管理権限のみを含むロール 1 を付与されました。

ユーザ 1、2、および 3 には、次の権限があります。

ロール/権限	ユーザ 1	ユーザ 2	ユーザ 3
MANAGE ANY USER	行使	行使	-
MANAGE ANY LOGIN POLICY	行使、管理	行使、管理	-
CHANGE PASSWORD	管理	管理	-
R1	-	管理	管理
R2	-	-	-

## シナリオ 6



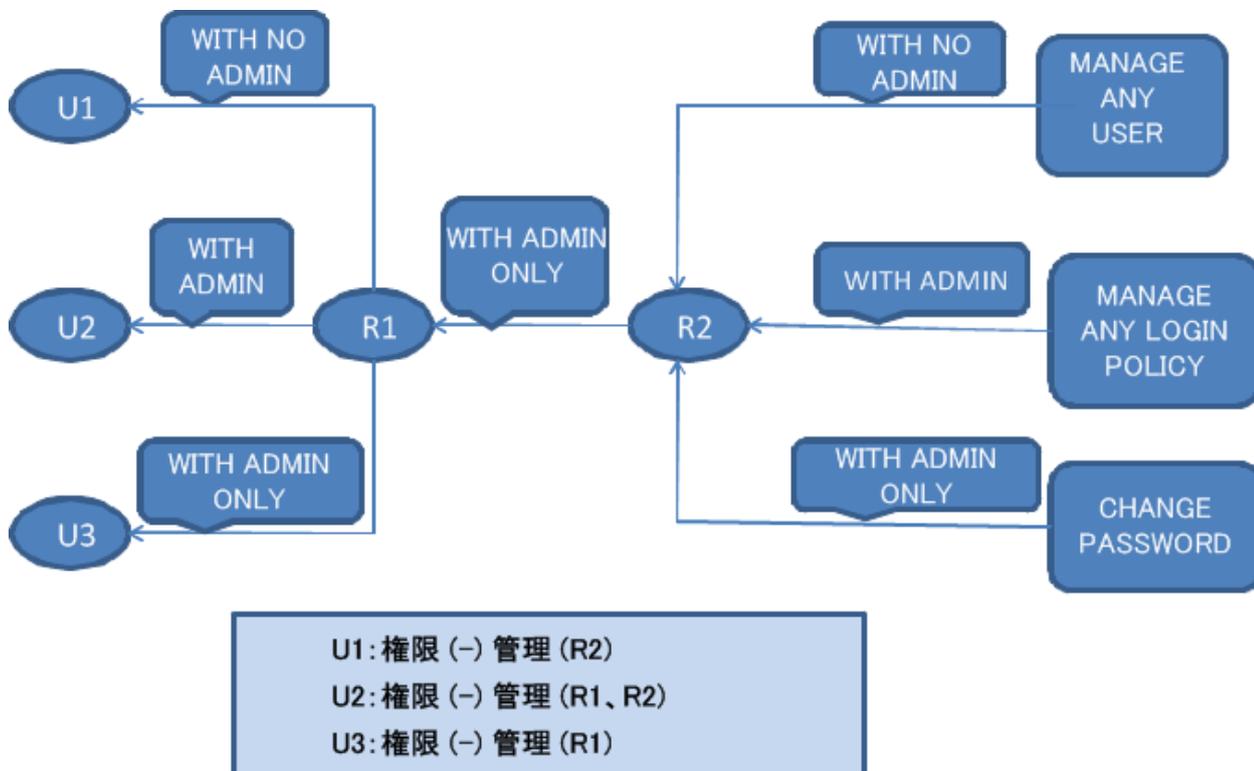
この図は、次のことを示しています。

- ロール 2 には、管理権限を含まない MANAGE ANY USER システム権限が付与されました。
- ロール 2 には、管理権限を含む MANAGE ANY LOGIN POLICY システム権限が付与されました。
- ロール 2 には、管理権限のみを含む CHANGE PASSWORD システム権限が付与されました。
- ロール 1 は、管理権限を含むロール 2 を付与されました。
- ユーザ 1 は、管理権限を含まないロール 1 を付与されました。
- ユーザ 2 は、管理権限を含むロール 1 を付与されました。
- ユーザ 3 は、管理権限のみを含むロール 1 を付与されました。

ユーザ 1、2、および 3 には、次の権限があります。

ロール/権限	ユーザ 1	ユーザ 2	ユーザ 3
MANAGE ANY USER	行使	行使	-
MANAGE ANY LOGIN POLICY	行使、管理	行使、管理	-
CHANGE PASSWORD	管理	管理	-
R1	-	管理	管理
R2	管理	管理	-

## シナリオ 7



この図は、次のことを示しています。

- ロール 2 には、管理権限を含まない MANAGE ANY USER システム権限が付与されました。
- ロール 2 には、管理権限を含む MANAGE ANY LOGIN POLICY システム権限が付与されました。
- ロール 2 には、管理権限のみを含む CHANGE PASSWORD システム権限が付与されました。
- ロール 1 は、管理権限のみを含むロール 2 を付与されました。
- ユーザ 1 は、管理権限を含まないロール 1 を付与されました。
- ユーザ 2 は、管理権限を含むロール 1 を付与されました。
- ユーザ 3 は、管理権限のみを含むロール 1 を付与されました。

ユーザ 1、2、および 3 には、次の権限があります。

ロール/権限	ユーザ 1	ユーザ 2	ユーザ 3
MANAGE ANY USER	-	-	-
MANAGE ANY LOGIN POLICY	-	-	-
CHANGE PASSWORD	-	-	-
R1	-	管理	管理
R2	管理	管理	-

## 1.9.1.6 同一化

ユーザは、一時的にデータベースの別のユーザの ID を仮定します。これは同一化とも呼ばれ、同一化しているユーザの権限のスーパーセットを持つ場合に操作を実行できます。

この制限は、最低の基準と呼ばれ、管理権限やオブジェクトレベル権限にも及びます。ユーザが、最低でも同一化したいユーザと同じ権限および管理権限を持たない場合、同一化はできません。

ユーザが最低の基準を満たすか、それ以上である場合に、同一化している他のユーザが実行する代わりに、なぜそのユーザが操作を実行しないのか不思議に思うかもしれません。その理由は、同一化を行っているユーザがタスクに必要な権限以上を持っている場合、他の権限がタスクの出力に影響を与える可能性があるためです。通常そのタスクを実行するユーザを同一化すれば、この可能性を否定できます。目的は、同一化されているユーザに有効な権限、場合によってはデータベースオプションを再作成することです。

別のユーザを同一化する機能は、SET USER システム権限で制御されます。SET USER システム権限を付与するとき、次のうちのいずれかを同一化できるユーザに設定できます。

- データベース内の任意のユーザ
- 指定したユーザのリストにあるユーザ
- 1つ以上の指定したロールのリストの被付与者であるユーザ

最低の基準は、SET USER システム権限が付与されるときに評価されません。代わりに、ユーザが SETUSER 文を実行して別のユーザを同一化しようとしたときに評価されます。

同一化セッションが進行中に、最低の基準に違反することになる GRANT または REVOKE 操作はデータベースサーバによって許可されず、付与または取り消し操作を続行できないことを示すエラーメッセージが返されます。

### 例

データ入力担当者である JSmithClerk がデータベースでの操作の実行に困っているとします。彼は IT 部門の PJonesIT に相談し、PJonesIT は JSmithClerk を同一化して様子を見て、JSmithClerk が抱えている問題のトラブルシューティングを行うことにしました。PJonesIT が JSmithClerk を同一化するためには、次の基準を満たす必要があります。

1. PJonesIT には、JSmithClerk が持つ権限のスーパーセットが必要です。
2. PJonesIT には、次の 3 つの方法のいずれかで SET USER システム権限が付与されている必要があります。
  - JSmithClerk を含む 1 人以上のユーザに対する直接的な同一化権限 (`GRANT SET USER (JSmithClerk) TO PJonesIT;`)
  - すべてのユーザに対する同一化権限 (`GRANT SET USER TO PJonesIT;`)
  - JSmithClerk が持つロールの 1 つ以上を持つユーザに対する同一化権限 (`GRANT SET USER (WITH ROLES ...) TO PJonesIT;`)
3. PJonesIT は SETUSER 文を発行し、JSmithClerk として同一化セッションを開始する必要があります (`SETUSER JSmithClerk`)。同一化セッションの実行中に行われる操作は、JSmithClerk の権限を使用して実行されます。PJonesIT は、JSmithClerk に対して有効なデータベースオプションを同一化セッションに適用したいと思うかもしれませんが、その場合、PJonesIT は、SETUSER 文に WITH OPTION 句を含めず (`SETUSER WITH OPTION JSmithClerk;`)。

## 同一化についてのその他の注意事項

- 同一化セッションは、同一化を行っているユーザが SETUSER 文を (ユーザを指定せずに) 実行するか、接続が終了するか、ユーザが別のユーザの同一化を開始するまで続行されます。
- SQL Central では、ユーザの **システム権限** タブでは、SET USER ロー (ある場合) にユーザが誰を同一化できるのかに関する詳細が表示されます。詳細を確認するには、マウスを **[オプション]** セルに置きます。詳細には、ユーザに直接付与された SET USER 権限だけが含まれ、継承した SET USER 権限は含まれません。
- ユーザが別のユーザを同一化して操作を実行するとき、操作のトランザクションログに記録されるのは、同一化されているユーザのユーザ ID です。ただし、SETUSER 文もトランザクションログに記録されるので、管理者は同一化セッションの開始 (と終了) を検出できます。
- ユーザは、別のユーザを引き続き同一化しているのかがわからない場合、SELECT CURRENT USER を実行します。返される値は、操作に使用されているユーザ ID です。
- Interactive SQL では、文の実行時に有効なユーザ ID がウィンドウのタイトルバーに表示されます。

このセクションの内容:

[同一化の最低の基準の詳細 \[1548 ページ\]](#)

最低の基準が満たされている場合にだけ、ユーザは他のユーザの同一化を正常に行えます。

### 1.9.1.6.1 同一化の最低の基準の詳細

最低の基準が満たされている場合にだけ、ユーザは他のユーザの同一化を正常に行えます。

基準の検証は、SET USER システム権限が付与される時ではなく、SETUSER 文が実行される時に行われます。SETUSER 文の発行時にユーザが基準を満たしていない場合、同一化の試行は失敗し、エラーが返されます。

正常の同一化のための基準は次の 4 つです。

1. 同一化実行者にターゲットユーザを同一化する権限が付与されていること。
2. 同一化実行者は、最低でもターゲットユーザに付与されているすべてのロールとシステム権限を持っていること。
3. 同一化実行者に、同様またはそれ以上の管理権限を持つロールとシステム権限が付与されていること。  
管理権限の基準を満たすという目的のため、WITH ADMIN OPTION および WITH ADMIN ONLY OPTION 句が同様の管理権限を付与するとみなされること。また、WITH NO ADMIN OPTION 句以上の管理権限を付与するとみなされること。たとえば、User1 は WITH ADMIN OPTION 句で Role1 を付与され、User2 は WITH ADMIN ONLY 句で Role1 を付与され、User3 は WITH NO ADMIN OPTION 句で Role1 を付与されます。User1 と User2 は、同様の管理権限を持つ Role1 を持っているともみなされます。また、User1 と User2 は、User 3 以上の管理権限を持つ Role1 を持っているともみなされます。
4. 拡張とも呼ばれる追加のパラメータをサポートするシステム権限がターゲットユーザに付与されている場合 (たとえば、SET USER システム権限を付与するときにユーザ ID のリストを提供できます)、同一化実行者にシステム権限を付与するために使用される句は、ターゲットユーザに使用されるもののスーパーセットです。  
現在、拡張をサポートするのは SET USER システム権限と CHANGE PASSWORD システム権限のみです。SET USER および CHANGE PASSWORD を同一化実行者に付与するときに指定される拡張は、ターゲットユーザに SET USER または CHANGE PASSWORD が付与されたときに指定された拡張と同等またはそのスーパーセットである必要があります。これは次のように評価されます。

- ANY 拡張は `role-list` 拡張 (ロールのリスト) および `users-list` 拡張 (ユーザのリスト) に対するスーパーセットとみなされます。ターゲットユーザが ANY 拡張を含む SET USER システム権限 (たとえば、GRANT SET USER ANY TO user1) を持つ場合、同一化実行者も ANY 拡張を持っている必要があります。
- ターゲットユーザが `role-list` と `users-list` 両方の拡張を含む SET USER システム権限を持っている場合、同一化実行者も 2 つの拡張を含むシステム権限を持つ必要があり、各拡張はターゲットユーザの対応する拡張と同等またはそのスーパーセットである必要があります。たとえば、同一化実行者とターゲットユーザ両方の拡張リストに User1、User2 および Role1、Role2 がそれぞれ含まれる場合、各句のターゲットリストの付与は同等とみなされます。代わりに、同一化実行者の `users-list` に User1、User2、Role1、Role2 がそれぞれ含まれる場合、ターゲットユーザの `users-list` には User1、Role2 のみが含まれ、同一化実行者の `users-list` はターゲットユーザのリストのスーパーセットとみなされます。
- ターゲットユーザに 1 つのリスト拡張を含む SET USER システム権限がある場合、同一化実行者の拡張リストは、ターゲットユーザのリストと同等またはそのスーパーセットである必要があります。たとえば、同一化実行者とターゲットユーザ両方の `user_list` に User1 と User2 が含まれる場合、両者は同等とみなされます。同一化実行者の `user_list` に User1、User2 が含まれる場合、ターゲットユーザの `user_list` には User2 のみが含まれ、同一化実行者の `user_list` はターゲットユーザの `user_list` のスーパーセットとみなされます。

したがって、同一化実行者は、使用できるユーザを制限する拡張を含む SET USER または CHANGE PASSWORD システム権限を持つが、ターゲットユーザは制限のないシステム権限を持つ場合、拡張に関する最低の基準を満たしていないため、同一化実行者は同一化に失敗します。

## 例

シナリオ 1-2 番目と 3 番目の基準が満たされていると仮定して、次のシナリオを検討します。

- 次の 5 人のユーザが存在するとします。User1、User2、User3、User4、および User5。
- 次の 2 つのロールがあります。Role1 と Role2。
- User1 には ANY 句を含む SET USER システム権限が付与されています。
- User2 には、User1 と User4 に対する `user-list` 句を含む SET USER システム権限が付与されています。
- User3 には、User1、User2、User4、User5 に対する `user-list` 句を含む SET USER システム権限と、Role1 および Role2 に対する `role-list` 句を含む ANY WITH ROLES システム権限が付与されています。
- User4 には、ANY 句および Role1 に対する `role-list` 句を含む SET USER システム権限が付与されています。
- User5 には、User4 に対する `user-list` 句を含む SET USER システム権限と、Role1 に対する `role-list` 句を含む ANY WITH ROLES システム権限が付与されています。
- User1 と User4 は、それぞれ ANY 句を含む SET USER システム権限が付与されているため、User2、User3、User5 を正常に同一化できます(基準 4)。

User2、User3、User5 は、ANY の付与がないため User1 または User4 を同一化できません。(基準 4)

User2 は、次の理由から User3 または User5 を同一化できません。

- User2 にこれらのユーザの同一化を行う権限が付与されていないため。(基準 1)
- SET USER システム権限が、`role-list` 句を指定して User2 に付与されていない(基準 4)

User3 は次の理由により正常に User2 を同一化できます。

- User3 に `user-list` 句経由で User2 を同一化する権限が付与されているため。(基準 1)
- User3 の `user-list` 句は User2 のスーパーセットであるため。(基準 4) User3 には `role-list` 句が付与されていますが、後者は同じ付与を持たないため、User2 の同一化の要件を満たす必要ありません。

User3 は次の理由により正常に User5 を同一化できます。

- User3に `user-list` 句経由で User5を同一化する権限が付与されているため。(基準 1)
- User3の `user-list` 句リストは User5のスーパーセットであるため。(基準 4)
- User3と User5の `role-list` 句リストが同等であるため。(基準 4)

User5は次の理由により他のユーザを同一化できません。

- User1と User4に ANY 付与があるため (基準 4)
- User2と User3に、User5への付与のサブセットではない `user-list` 句を含む付与があるため。(基準 4)

シナリオ 2 - 1番目と4番目の基準が満たされていると仮定して、次を検討します。

- 次の2人のユーザがいます。User6と User7。
- 次の2つのロールがあります。Role4と Role5。
- User6には WITH ADMIN OPTION 句を含む Role4、WITH ADMIN ONLY OPTION 句を含む Role5、WITH ADMIN OPTION 句を含む MANAGE ANY USER システム権限が付与されています。
- User7には、WITH ADMIN OPTION 句を含む Role4と WITH NO ADMIN OPTION 句を含む Role5が付与されています。

User6は次の理由により正常に User7を同一化できます。

- User6と User7はいずれも Role4と Role5が付与されています。User6に追加の権限 (MANAGE ANY USER システム権限) が付与されていても関係ありません。(基準 2)
- User6には、User7と同等の管理権限を含む Role4が付与されています。User6には、User7以上の管理権限を含む Role5が付与されています。(基準 3)

User7は次の理由により User6を同一化できません。

- User7には Role4と Role5が含まれているが、MANAGE ANY USER システム権限が付与されていないため。(基準 2)
- User7には、User6以下の管理権限を含む Role5が付与されているため。(基準 3)

### シナリオ 3

次を検討します。

- 次の3人のユーザがいます。User8、User9、User10。
- 次の2つのロールがあります。Role5と Role6。
- User8には WITH ADMIN OPTION 句を含む Role5、および WITH ADMIN OPTION 句を含む MANAGE ANY USER システム権限が付与されています。
- User9および User10には、WITH NO ADMIN OPTION 句を含む Role5が付与されています。
- User8には、`user-list` 句を含み、User9と User10を同一化するための SET USER システム権限が付与されています。
- User9には、`user-list` 句を含み、User10を同一化するための SET USER システム権限が付与されています。

User8は次の理由により正常に User9を同一化できます。

- User8に `user-list` 句経由で User9を同一化する権限が付与されているため。(基準 1)
- User8の `user-list` 句リストは User9のスーパーセットであるため。(基準 4)
- User8と User9はともに Role5を付与され、User8は User9以上にロールへの管理権限が付与されているため。(基準 2 および 3)

User8は次の理由により正常に User10を同一化できます。

- User8には User10を同一化するための権限が付与されているため。(基準 1)

- User10 には SET USER システム権限が付与されていないため、基準 4 は適用されません。
- User8 と User10 には、ともに Role5 が付与され、ロールには同じ管理権限が付与されているため。(基準 2 および 3)

User9 は次の理由により User8 を同一化できません。

- User9 に User8 の同一化を行う権限が付与されていないため。(基準 1)
- User8 と User9 にはともに Role5 が付与されているが、User9 への付与には、User8 に対するものよりロールへの管理権限の付与が少ないため。(基準 3)

基準の検証は、SET USER システム権限が付与される時ではなく、SETUSER 文が実行されるときに行われます。SETUSER 文の発行時にいずれかの基準が満たされていないと、パーミッションが拒否されたというメッセージが表示され、同一化は開始されません。

## 1.9.1.7 ロールベースのセキュリティ階層のプランと実装

ロールベースのセキュリティ階層を計画し実装する必要があります。

### セキュリティ階層の設計

1. さまざまな承認済みタスクがユーザによって実行されるように特定します。密接に関連するタスクをグループ化します。グループ化は、任意の組織構造 (部門、機能、など) に基づいて行うことができます。各グループに名前を割り当てます。こういったグループ化は、作成するロールです。
2. 特定された各タスクの実行に必要なシステム権限およびオブジェクトレベル権限を特定します。
3. 作成するロールを必要とするユーザなど、さまざまな承認済みタスクの実行に必要なユーザを特定します。
4. (省略可) 作成するロールの管理者を特定します。管理者は、他のユーザにロールを付与したり、ロールを取り消したりできます。
5. (省略可) 作成するロールの一部ではないシステム権限とオブジェクトレベル権限の管理者を特定します。

### セキュリティ階層の構築

1. 必要なロールを作成します。
2. 各ロールに対して、必要なシステム権限およびオブジェクトレベル権限を付与します。
3. ユーザを作成します。
4. 該当する場合の管理権限の付与をはじめ、ユーザに適用可能なロールを付与します。
5. 個々のユーザが必要とする追加のロールおよび権限を付与します。

このセクションの内容:

[セキュリティ: コマンドラインからの権限の制御 \[1552 ページ\]](#)

ある特定のグローバルオペレーションの実行に必要な権限を制御するデータベースサーバオプションがあります。制御される権限には、データベースの起動と停止、データのロードとアンロード、データベースファイルの作成と削除などを行うものがあります。

### 1.9.1.7.1 セキュリティ: コマンドラインからの権限の制御

ある特定のグローバルオペレーションの実行に必要な権限を制御するデータベースサーバオプションがあります。制御される権限には、データベースの起動と停止、データのロードとアンロード、データベースファイルの作成と削除などを行うものがあります。

たとえば、-gd データベースサーバオプションでは、実行中のデータベースサーバ上でデータベースを起動または停止するために必要な権限を設定します。

#### 関連情報

[セキュリティに関する一般的なヒント \[1598 ページ\]](#)

[データベースサーバ起動オプション \[375 ページ\]](#)

### 1.9.1.8 ネストされたオブジェクトの所有権

ビューとプロシージャは、さまざまなユーザが所有する基本のオブジェクトにアクセスできます。

たとえば、usera、userb、userc、userd が別々の 4 名のユーザである場合は、userc.viewc から userd.viewd を作成できます。この userc.viewc は、usera.table から作成された userb.viewb をベースにして作成できます。同じように、プロシージャでも、userd.procd は userc.procc を呼び出すことができ、userc.procc は、usera.tablea に挿入できる userb.procb を呼び出すことができます。

ネストされたビューおよびテーブルには、次の DAC (任意アクセス制御) 規則が適用されます。

- ビューを作成するには、ユーザは、そのビューに含まれるすべてのベースオブジェクト (テーブルやビューなど) に対する SELECT 権限が必要です。
- ビューにアクセスするには、ビューの所有者は、基本のテーブルまたはビューに対する適切な権限を (管理権限で) 付与されている必要があります。また、ユーザは、ビューに対する適切な権限を付与されている必要があります。
- WHERE 句を使用して更新するには、SELECT 権限と UPDATE 権限の両方が必要です。
- ユーザがビュー定義内のテーブルを所有している場合は、そのユーザがビューの所有者ではなく、ビューに対するアクセス権を付与されていなくても、ビューを介してテーブルにアクセスできます。

ネストされたプロシージャには、次の DAC 規則が適用されます。

- プロシージャを作成する場合、ユーザは、基本となるオブジェクト (たとえば、テーブル、ビュー、またはプロシージャ) に対する権限を必要としません。
- プロシージャを実行する場合、プロシージャの所有者は、そのプロシージャが参照するオブジェクトに対する適切な権限を必要とします。

- プロシージャによって参照されるすべてのテーブルをユーザが所有する場合でも、プロシージャに対する EXECUTE 権限を付与されていない限り、プロシージャを実行してテーブルにアクセスすることはできません。

この動作については、次の例で説明します。

このセクションの内容:

例 1: user1 が table1 を作成し、user2 が table1 に基づく view2 を作成する [1553 ページ]

user1 が table1 を作成し、user2 が table1 に基づく view2 を作成します。

例 2: user2 が table1 にアクセスする procedure2 を作成する [1553 ページ]

user2 が table1 にアクセスする procedure2 を作成します。

例 3: user1 が table1 を作成し、user2 が table2 を作成し、user3 が table1 と table2 をジョインする view3 を作成する [1554 ページ]

user1 が table1 を作成し、user2 が table2 を作成し、user3 が table1 と table2 をジョインする view3 を作成する

### 1.9.1.8.1 例 1: user1 が table1 を作成し、user2 が table1 に基づく view2 を作成する

user1 が table1 を作成し、user2 が table1 に基づく view2 を作成します。

- user1 は所有者であるため、常に table1 にアクセスできます。
- user1 は、基本となるテーブルの所有者であるため、常に view2 を介して table1 にアクセスできます。これは、user2 が view2 の権限を user1 に付与しない場合でも該当します。
- user2 が table1 に直接または view2 を介してアクセスできるのは、user1 が table1 の権限を user2 に付与した場合です。
- user3 が table1 にアクセスできるのは、user1 が table1 の権限を user3 に付与した場合です。
- user3 は、user1 が table1 の権限を grant オプションで user2 に付与し、かつ、user2 が view2 の権限を user3 に付与した場合に、view2 を介して table1 にアクセスできます。

### 1.9.1.8.2 例 2: user2 が table1 にアクセスする procedure2 を作成する

user2 が table1 にアクセスする procedure2 を作成します。

- user1 が procedure2 を介して table1 にアクセスできるのは、user2 が procedure2 の EXECUTE 権限を user1 に付与した場合です。view2 では、user1 は権限を必要としませんが、上記の場合とは異なります。

### 1.9.1.8.3 例 3: user1 が table1 を作成し、user2 が table2 を作成し、user3 が table1 と table2 をジョインする view3 を作成する

user1 が table1 を作成し、user2 が table2 を作成し、user3 が table1 と table2 をジョインする view3 を作成する

- user3 が view3 を介して table1 と table2 にアクセスできるのは、user1 が table1 の権限を user3 に付与し、かつ、user2 が table2 の権限を user3 に付与した場合です。
- user3 が table1 の権限を持っているが table2 の権限を持っていない場合、user3 は view3 を使用できません。table1 のカラムからなるサブセットにもアクセスできません。
- user1 または user2 が view3 を使用できるのは、(a) user1 が table1 の権限を grant オプションとともに user3 に付与し、(b) user2 が table2 の権限を grant オプションとともに user3 に付与し、さらに (c) user3 が view3 の権限を user1 または user2 に付与した場合です。

### 1.9.1.9 パスワードおよびユーザ ID の制限事項と考慮事項

ユーザは、データベースに接続可能なパスワードを持っている必要があります。

ユーザ ID には、次に該当する値を指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロンを含む値

パスワードは大文字と小文字が区別され、次に該当する値は指定できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロンを含む値
- 255 バイト長を超える値

デフォルトでは、パスワードが 6 バイトの長さである必要があります。この要件を変更するには、min\_password\_length データベースオプションを設定します。

パスワードに関する追加のルールは、ユーザのログインポリシーで設定できます。

作成または変更したパスワードは、UTF-8 に変換され、SHA256 ハッシュを使用して暗号化された後、データベースに保存されます。データベースをアンロードし、別の文字セットを使用するデータベースに再ロードした場合、既存のパスワードが引き続き機能します。データベースサーバがクライアントの文字セットを UTF-8 に変換できない場合、パスワードには 7 ビット ASCII 文字を使用してください。それ以外の文字を使用すると、パスワードが機能しないことがあります。

パスワードは暗号化された後、クライアントからデータベースサーバに送信されます。

このセクションの内容:

[パスワードの変更 \(SQL Central の場合\) \[1555 ページ\]](#)

ユーザのプロパティを編集して、そのユーザのパスワードを変更します。

[パスワードの変更 \(SQL の場合\) \[1556 ページ\]](#)

IDENTIFIED BY 句を含む ALTER USER 文を使用して、ユーザのパスワードを変更します。

[二重制御パスワード \[1557 ページ\]](#)

二重制御パスワード機能では、パスワードを変更する 2 人の管理者が必要です。

## 関連情報

[ログインポリシー \[613 ページ\]](#)

[min\\_password\\_length オプション \[736 ページ\]](#)

### 1.9.1.9.1 パスワードの変更 (SQL Central の場合)

ユーザのプロパティを編集して、そのユーザのパスワードを変更します。

#### 前提条件

どのユーザも自分のパスワードを変更できます。

別のユーザのパスワードを変更するには、CHANGE PASSWORD システム権限が必要です。

二重制御パスワード変更が有効になっている (つまり、change\_password\_dual\_control ログインポリシーオプションが ON に設定されている) 別のユーザのパスワードを変更するために SQL Central を使用できません。自分のパスワードを変更するときに二重制御パスワード変更が ON に設定されている場合は、通常通りパスワードを変更できます。二重制御パスワード変更は適用されません。

#### コンテキスト

ユーザにはパスワードが設定されており、それを変更しようとしています。

#### 手順

1. SQL Central で、[SQL Anywhere17](#) プラグインを使用してデータベースに接続します。
2. [ユーザ](#) をダブルクリックし、ユーザを右クリックして [プロパティ](#) をクリックします。
3. [このユーザにパスワードを設定](#) フィールドが選択されていることを確認します。
4. [パスワード](#) フィールドと [パスワードの確認](#) フィールドに値を入力します。
5. [適用](#) をクリックしてから [OK](#) をクリックします。

## 結果

ユーザパスワードが変更されました。

## 次のステップ

ユーザに新しいパスワードを通知してください。

## 関連情報

[ログインポリシー \[613 ページ\]](#)

[二重制御パスワードの設定 \(SQL の場合\) \[1558 ページ\]](#)

## 1.9.1.9.2 パスワードの変更 (SQL の場合)

IDENTIFIED BY 句を含む ALTER USER 文を使用して、ユーザのパスワードを変更します。

### 前提条件

どのユーザも自分のパスワードを変更できます。別のユーザのパスワードを変更するには、CHANGE PASSWORD システム権限が必要です。

自分のパスワードを変更するときに、ログインポリシーで二重制御パスワード変更が ON に設定されている場合は、通常通りパスワードを変更できます。自分のパスワードを変更するために、二重制御パスワード変更は必要ありません。

### コンテキスト

ユーザにはパスワードが設定されており、それを変更しようとしています。

### 手順

1. データベースに接続します。
2. 新しいパスワードを指定する IDENTIFIED BY 句を含む ALTER USER 文を実行します。

## 結果

ユーザパスワードが変更されました。

### 例

次の文を実行して、DBA ユーザのパスワードを welcome\_DBA に変更します。

```
ALTER USER DBA  
IDENTIFIED BY welcome_DBA;
```

## 次のステップ

ユーザに新しいパスワードを通知してください。

## 関連情報

[二重制御パスワードの設定 \(SQL の場合\) \[1558 ページ\]](#)

### 1.9.19.3 二重制御パスワード

二重制御パスワード機能では、パスワードを変更する 2 人の管理者が必要です。

この設定により、1 人の管理者が別のユーザのパスワードすべてを知ることができないようにします。1 人の管理者がパスワードの最初の部分を設定し、もう 1 人の管理者がパスワードの 2 番目の部分を設定します。ユーザは、データベースに接続するためのパスワードを 2 つの部分に分けて指定します。

ターゲットユーザは、そのログインポリシーで change\_password\_dual\_control オプションが有効になっている必要があります。

このセクションの内容:

[二重制御パスワードの設定 \(SQL の場合\) \[1558 ページ\]](#)

二重制御パスワードを設定するか、二重制御パスワードの最初または最後の部分を変更します。

## 関連情報

[ログインポリシーオプションとデフォルト値 \[614 ページ\]](#)

## 1.9.1.9.3.1 二重制御パスワードの設定 (SQL の場合)

二重制御パスワードを設定するか、二重制御パスワードの最初または最後の部分を変更します。

### 前提条件

CHANGE PASSWORD システム権限が必要です。

ターゲットユーザは、そのログインポリシーで `change_password_dual_control` オプションが有効になっている必要があります。

### コンテキスト

二重制御パスワード機能により、ユーザのパスワードすべてを管理者が知ってしまうことを防ぎます。

別の管理者が、パスワードの残りの部分を変更する必要があります。ユーザは、パスワードの 2 つの部分を組み合わせ、この組み合わせたパスワードを使用してデータベースに接続します。

パスワードの各部分では大文字と小文字が区別され、次に該当する値は使用できません。

- 空白スペース、一重引用符、または二重引用符で始まる値
- 空白スペースで終わる値
- セミコロンを含む値
- 127 バイト長を超える値

### 手順

1. データベースに接続します。
2. 管理者は次の文を実行して、二重制御パスワードの最初の部分を設定し、次に `password-part1` をユーザに伝えます。

```
ALTER USER user-name IDENTIFIED FIRST BY password-part1;
```

3. もう 1 人の管理者は次の文を実行して、二重制御パスワードの 2 番目の部分を設定し、次に `password-part2` をユーザに伝えます。

```
ALTER USER user-name IDENTIFIED LAST BY password-part2;
```

## 結果

ターゲットユーザのパスワードが、パスワードの最初と最後の部分を組み合わせて使用するよう設定されます (password-part1 password-part2)。

## 次のステップ

ターゲットユーザは、それぞれの部分を組み合わせてパスワードとして使用してデータベースに接続し、新しいパスワードの作成が要求されます。

## 関連情報

[ログインポリシー \[613 ページ\]](#)

## 1.9.1.10 チュートリアル: ロールと権限の付与 (SQL Central の場合)

ロールベースのセキュリティモデルでは、ユーザが特定のデータベース操作を実行するには、正しい権限およびロールを持っている必要があります。

## 前提条件

なし。DBA というユーザ ID でデータベースにログインします。このユーザは、チュートリアルに必要なシステム権限およびロールをすべて持っています。

## コンテキスト

このチュートリアルであなたは、すべての管理機能を備えたデータベース管理者です。プロシージャのデバッグと、デバッグプロセス中に見つかったあらゆるエラーの修正を担当する 2 名の同僚が必要です。また、同じ同僚に、会社のデータベース (このシナリオでは、データベースは SQL Anywhere サンプルデータベースです) のバックアップの実行を担当してもらうことも必要です。最後に、この両方の同僚が、データベースの Employees テーブルを更新できる必要があります。

両方の同僚のユーザ ID を作成して、これらのタスクを行うために必要な権限とロールを両者に付与する必要があります。このチュートリアルでは、SQL Central を使用してチュートリアルのタスクを実行します。

## i 注記

SQL Central は、ユーザの持つ権限やロールなど、データベースに関する情報をキャッシュします。別の接続によるデータベースに変更を加えた場合、SQL Central では、SQL Central でデータベースを再表示するまで変更を認識できません。左パネルのツリーでデータベースまたは任意のオブジェクトのいずれかを選択し、[F5] を押せば認識できます。

1. [レッスン 1: 次の 2 名の新規ユーザを作成します。User1 および User2 \(SQL Central の場合\) \[1560 ページ\]](#)  
SQL Central を使用してサンプルデータベースにログインし、2 名のユーザを新規作成します。
2. [レッスン 2: User1 への権限の付与 \(SQL Central の場合\) \[1561 ページ\]](#)  
システム権限とオブジェクトレベル権限を User1 に付与します。
3. [レッスン 3: ロールを作成して User1 に付与する \(SQL Central の場合\) \[1563 ページ\]](#)  
プロシージャをデバッグし、デバッグプロセス中に見つかったエラーを修正するのに必要な権限を含むロールを作成し、そのロールを User1 に付与します。
4. [レッスン 4: User1 をユーザ拡張ロールに変換し、そのロールを User2 に付与する \(SQL Central の場合\) \[1564 ページ\]](#)  
User1 をユーザ拡張ロールに変換し、そのロールを User2 に付与します。
5. [レッスン 5: User2 のロールと権限を表示する \(SQL Central の場合\) \[1566 ページ\]](#)  
他のロールでのメンバーシップを介してユーザが継承したものも含め、ユーザが持つロールと権限を表示します。

## 関連情報

[権限ベースのセキュリティからロールベースのセキュリティへのアップグレード \[1575 ページ\]](#)

### 1.9.1.10.1 レッスン 1: 次の 2 名の新規ユーザを作成します。User1 および User2 (SQL Central の場合)

SQL Central を使用してサンプルデータベースにログインし、2 名のユーザを新規作成します。

## 手順

1. SQL Central を起動し、データベースに接続します。
  - a. **▶ スタート ▶ プログラム ▶ SQL Anywhere17 ▶ 管理ツール ▶ SQL Central ▶** をクリックします。
  - b. **▶ 接続 ▶ SQL Anywhere17 に接続 ▶** をクリックします。
  - c. パスワードフィールドにパスワード **sql** を入力します。
  - d. アクションドロップダウンリストで、**ODBC データソースを使用した接続** を選択します。
  - e. **ODBC データソース名** で、**参照** をクリックして **SQL Anywhere 17 Demo** を選択し、さらに **OK** をクリックします。
  - f. **接続** をクリックします。これにより、パスワード sql、データベースでの完全な管理権限を持つ DBA というユーザとしてサンプルデータベースに接続されます。

2. 2 名の新規ユーザを作成します。

- a. 左ウィンドウ枠で **ユーザ** を右クリックし、**新規 ユーザ** をクリックします。
- b. **新しいユーザの名前を指定してください**。フィールドに **User1** と入力し、**次へ** をクリックします。
- c. パスワード **sql** を User1 に割り当て、**完了** をクリックします。
- d. 上記の手順を繰り返して、2 番目のユーザ、**User2** を作成します。

## 結果

新規ユーザ、User1 と User2 が作成されました。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL Central の場合\) \[1559 ページ\]](#)

次のタスク: [レッスン 2: User1 への権限の付与 \(SQL Central の場合\) \[1561 ページ\]](#)

## 関連情報

[ユーザの作成 \(SQL Central の場合\) \[1531 ページ\]](#)

[ユーザの作成 \(SQL の場合\) \[1532 ページ\]](#)

## 1.9.1.10.2 レッスン 2: User1 への権限の付与 (SQL Central の場合)

システム権限とオブジェクトレベル権限を User1 に付与します。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

## 手順

1. SQL Central の左ウィンドウ枠で **ユーザ** をクリックして展開し、**User1** をクリックします。
2. BACKUP DATABASE と VALIDATE ANY OBJECT システム権限を User1 に付与します。
  - a. 右ウィンドウ枠で **システム権限** タブを選択します。
  - b. タブを右クリックして、**新規** > **付与されたシステム権限** をクリックします。
  - c. **BACKUP DATABASE** システム権限をクリックしてから **OK** をクリックします。
  - d. 上記手順を繰り返し、今度は **[VALIDATE ANY OBJECT]** システム権限をクリックしてから **[OK]** をクリックします。
  - e. 変更をデータベースに保存するには、**ファイル** > **保存** をクリックします。
3. Employees テーブルでの SELECT UPDATE オブジェクトレベル権限を、管理権限付きで User1 に付与します。
  - a. 右ウィンドウ枠で **テーブル権限** タブをクリックします。
  - b. タブを右クリックして、**新規** > **権限** をクリックします。
  - c. **Employees** テーブルをクリックしてから **OK** をクリックします。Employees テーブルが User1 のテーブルリストに追加されます。文字カラムの各チェックマークは、User1 がテーブルに対して持っているオブジェクトレベル権限を示します。たとえば、**S** は選択、**I** は挿入、などです。ただし、このレッスンでは User1 だけが Employees テーブルからの選択、および更新を行えるようになっています。
  - d. カラムの下のセルをクリックし、**S** (選択用) および **U** (更新用) カラムにのみプラス記号 (+) 付きでチェックマークが残るように、内容をクリアします。プラス記号は管理権限を示します。ローの他のすべてのセルは空にする必要があります。
  - e. 変更をデータベースに保存するには、**ファイル** > **保存** をクリックします。

## 結果

User1 が BACKUP DATABASE と VALIDATE ANY OBJECT システム権限を持つようになります。この権限は、SQL Anywhere サンプルデータベースのバックアップに必要です。また、User1 は Employees テーブルでの SELECT および UPDATE オブジェクトレベル権限も持ち、権限を他のユーザに付与できます。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL Central の場合\) \[1559 ページ\]](#)

前のタスク: [レッスン 1: 次の 2 名の新規ユーザを作成します。User1 および User2 \(SQL Central の場合\) \[1560 ページ\]](#)

次のタスク: [レッスン 3: ロールを作成して User1 に付与する \(SQL Central の場合\) \[1563 ページ\]](#)

## 関連情報

[システム権限 \[1502 ページ\]](#)

[システム権限の付与 \(SQL Central の場合\) \[1520 ページ\]](#)

[システム権限の付与 \(SQL の場合\) \[1521 ページ\]](#)

### 1.9.1.10.3 レッスン 3: ロールを作成して User1 に付与する (SQL Central の場合)

プロシージャをデバッグし、デバッグプロセス中に見つかったエラーを修正するのに必要な権限を含むロールを作成し、そのロールを User1 に付与します。

#### 前提条件

このチュートリアルのものでのレッスンを完了している必要があります。

#### 手順

1. ロールを作成します。
  - a. SQL Central の左ウィンドウ枠で **ロール** を右クリックし、**新規** > **ロール** をクリックします。
  - b. **新しいユーザ定義ロールの名前を指定してください**。フィールドに **DebugAndAlter** と入力し、**次へ** をクリックします。
  - c. **管理者の選択** 画面で **完了** をクリックし、MANAGE ROLES システム権限を持つユーザ (別名、グローバル管理者) がロールを管理できるようにするデフォルトを受け入れます。
2. 必要なシステム権限をロールに付与します。
  - a. 左ウィンドウ枠で、**ロール** をダブルクリックし、**DebugAndAlter** を選択します。
  - b. 右ウィンドウ枠で、**システム権限** タブをクリックします。
  - c. タブを右クリックして、**新規** > **付与されたシステム権限** をクリックします。
  - d. **ALTER ANY OBJECT** システム権限をクリックしてから **OK** をクリックします。
  - e. ステップを繰り返して、**DEBUG ANY PROCEDURE** システム権限を追加します。
  - f. 変更をデータベースに保存するには、**ファイル** > **保存** をクリックします。
3. ロールに対する管理権限を持つ User1 にロールを付与します。
  - a. 左ウィンドウ枠で、**ユーザ** をダブルクリックし、**User1** を選択します。
  - b. 右ウィンドウ枠で、**ロール** タブをクリックします。
  - c. タブを右クリックして、**新規** > **付与されたロール** をクリックします。
  - d. **DebugAndAlter** をクリックしてから **OK** をクリックします。DebugAndAlter ロールを行使する機能が、管理権限なしで User1 に付与されます。

- e. 右ウィンドウ枠の DebugAndAlter ロールのローで、[\[管理\]](#) カラムのセルをクリックし、チェックマークを追加します。これにより、ロールに対する管理権限が User1 に付与されます。
- f. 変更をデータベースに保存するには、[▶ ファイル ▶ 保存](#) をクリックします。

## 結果

DEBUG ANY PROCEDURE と ALTER ANY OBJECT システム権限を持つ DebugAndAlter というロールが作成され、User1 に付与されます。また、User1 はロールに対する管理権限も付与されます。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL Central の場合\) \[1559 ページ\]](#)

前のタスク: [レッスン 2: User1 への権限の付与 \(SQL Central の場合\) \[1561 ページ\]](#)

次のタスク: [レッスン 4: User1 をユーザ拡張ロールに変換し、そのロールを User2 に付与する \(SQL Central の場合\) \[1564 ページ\]](#)

## 関連情報

[ユーザ定義ロールの作成 \(SQL Central の場合\) \[1472 ページ\]](#)

[ロールにロールと権限を設定する \(SQL Central の場合\) \[1486 ページ\]](#)

[ロールの付与 \(SQL Central の場合\) \[1489 ページ\]](#)

### 1.9.1.10.4 レッスン 4: User1 をユーザ拡張ロールに変換し、そのロールを User2 に付与する (SQL Central の場合)

User1 をユーザ拡張ロールに変換し、そのロールを User2 に付与します。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

## コンテキスト

User2に User1と同じ権限とロールを付与するとします。User1に行ったロールと権限の明示的な付与と同じ長いプロセスをたどる代わりに、User1をユーザ拡張ロールに変換し、そのロールを User2に付与できます。User1はロールであると同時に、引き続きデータベースにログインできるユーザでもあります。

## 手順

1. User1をユーザ拡張ロールに変換します。
  - a. 左ウィンドウ枠で、**ユーザ**を展開し、**User1**を右クリックして、**ユーザ拡張ロールに変更**を選択します。
2. ユーザ拡張ロールの User1を User2に付与します。
  - a. 左ウィンドウ枠の**ユーザ**で、**User2**を選択します。
  - b. 右ウィンドウ枠で、**ロールタブ**を選択します。
  - c. タブを右クリックして、**新規** > **付与されたロール** を選択します。
  - d. **User1**をクリックしてから、**OK**をクリックします。
  - e. 変更をデータベースに保存するには、**ファイル** > **保存** をクリックします。

## 結果

User1がユーザ拡張ロールに変換され、User2に付与されました。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL Central の場合\) \[1559 ページ\]](#)

前のタスク: [レッスン 3: ロールを作成して User1に付与する \(SQL Central の場合\) \[1563 ページ\]](#)

次のタスク: [レッスン 5: User2のロールと権限を表示する \(SQL Central の場合\) \[1566 ページ\]](#)

## 関連情報

[ユーザ拡張ロール \[1475 ページ\]](#)

## 1.9.1.10.5 レッスン 5: User2 のロールと権限を表示する (SQL Central の場合)

他のロールでのメンバーシップを介してユーザが継承したのものも含め、ユーザが持つロールと権限を表示します。

### 前提条件

このチュートリアル of これまでのレッスンを完了している必要があります。

### 手順

1. SQL Central で左ウィンドウ枠の **ユーザ** をダブルクリックし、**User2** をクリックします。
2. 画面上部にある **[継承の表示]** チェックボックスがオンになっていることを確認します。
3. 右ウィンドウ枠で、**ロール** タブをクリックし、User2 のロールを表示します。

User2 は次のロールを持つことが、スクリーンショットに 表示されます。

#### dbo システムロール

User2 は、dbo システムロールでの行使権限を持ちますが、管理権限は持ちません。User2 は、PUBLIC システムロールのメンバーになることで、このシステムロールを継承しているため、付与者は存在しません。ユーザを作成すると、デフォルトで PUBLIC システムロールが付与されます。

#### DebugAndAlter ユーザ定義ロール

User2 には DebugAndAlter ロールでの行使および管理権限があります。User2 はこのユーザ定義ロールを User1 から継承しているため、付与者は存在しません。

#### PUBLIC システムロール

User2 には PUBLIC ロールでの行使権限があります。ユーザを作成すると、デフォルトで PUBLIC システムロールが付与されます。User2 の作成時にログインしたユーザであることから、付与者は DBA です。

#### User1 ユーザ拡張ロール

User2 は、User1 ユーザ拡張ロールでの行使権限を持ちますが、管理権限は持ちません。ロールを User2 に付与したときにログインしたユーザであることから、付与者は DBA です。

4. **システム権限** タブをクリックし、User2 が持つシステム権限を表示します。システム権限は明示的に User2 に付与されたのではなく継承されているため、**付与者**カラムは、すべてのカラムに対して空です。システム権限が付与されたときに管理権限は付与されていないため、**管理**カラムは空です。
5. **テーブル権限**タブをクリックして、User2 が User1 から継承した Employees テーブルのオブジェクトレベル権限を表示します。ユーザはオブジェクトレベル権限を継承できますが、オブジェクトレベル権限の管理権限は継承できません。User1 は Employees テーブルの SELECT および UPDATE 権限を行使および管理できますが、User2 はテーブルを SELECT および UPDATE する機能のみを継承します。

## 結果

User2 のロール、権限、管理権限を検証しました。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL Central の場合\) \[1559 ページ\]](#)

前のタスク: [レッスン 4: User1 をユーザ拡張ロールに変換し、そのロールを User2 に付与する \(SQL Central の場合\) \[1564 ページ\]](#)

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[ロール管理者 \[1481 ページ\]](#)

[システム権限の付与 \(SQL Central の場合\) \[1520 ページ\]](#)

[ロールの付与 \(SQL Central の場合\) \[1489 ページ\]](#)

### 1.9.1.11 チュートリアル: ロールと権限の付与 (SQL の場合)

ロールベースのセキュリティモデルでは、ユーザが特定のデータベース操作を実行するには、正しい権限およびロールを持っている必要があります。

#### 前提条件

なし。DBA というユーザ ID でデータベースにログインします。このユーザは、チュートリアルに必要なシステム権限およびロールをすべて持っています。

#### コンテキスト

このチュートリアルであなたは、すべての管理機能を備えたデータベース管理者です。プロシージャのデバッグと、デバッグプロセス中に見つかったあらゆるエラーの修正を担当する 2 名の同僚が必要です。また、同じ同僚に、会社のデータベース (このシナリオでは、データベースは SQL Anywhere サンプルデータベースです) のバックアップの実行を担当してもらうことも必要です。最後に、この両方の同僚が、データベースの Employees テーブルを更新できることが必要です。

両方の同僚のユーザ ID を作成して、これらのタスクを行うために必要な権限とロールを両者に付与する必要があります。

1. [レッスン 1: 次の 2 名のユーザを作成します。UserA と UserB \(SQL\) \[1568 ページ\]](#)

ユーザ ID DBA としてログインし、サンプルデータベースのバックアップ、プロシージャのデバッグ、デバッグプロセス中に見つかった誤りの修正を担当するユーザを 2 名作成します。

2. [レッスン 2: ユーザ A への権限の付与 \(SQL の場合\) \[1569 ページ\]](#)  
システム権限とオブジェクトレベル権限をユーザ A に付与します。
3. [レッスン 3: ロールを作成してユーザ A に付与する \(SQL の場合\) \[1570 ページ\]](#)  
プロシージャをデバッグし、デバッグプロセス中に見つかったエラーを修正するのに必要な権限を含むロールを作成し、そのロールをユーザ A に付与します。
4. [レッスン 4: ユーザ A をユーザ拡張ロールに変換し、そのロールをユーザ B に付与する \(SQL の場合\) \[1572 ページ\]](#)  
SQL で、ユーザ A をユーザ拡張ロールに変換し、そのロールをユーザ B に付与します。ユーザ A はロールであると同時に、引き続きデータベースにログインできるユーザでもあります。
5. [レッスン 5: ユーザ B のロールと権限を表示 \(SQL の場合\) \[1573 ページ\]](#)  
他のロールのメンバーシップを介してユーザが継承したものも含め、ユーザのロール、システム権限、オブジェクトレベル権限を表示します。

## 関連情報

[権限ベースのセキュリティからロールベースのセキュリティへのアップグレード \[1575 ページ\]](#)

### 1.9.1.11.1 レッスン 1: 次の 2 名のユーザを作成します。UserA と UserB (SQL)

ユーザ ID DBA としてログインし、サンプルデータベースのバックアップ、プロシージャのデバッグ、デバッグプロセス中に見つかった誤りの修正を担当するユーザを 2 名作成します。

## 手順

1. Interactive SQL で [接続](#) ウィンドウを開き、サンプルデータベースに接続するための情報を以下のフィールドに入力します。
  - a. **ユーザ ID** 項目に、**DBA** と入力します。
  - b. **パスワード** 項目に、**sql** と入力します。
  - c. **アクション** ドロップダウンリストで、**このコンピュータのデータベースを起動して接続** をクリックします。
  - d. **サーバ名** フィールドに **demo17** と入力します。
2. 2 名の新規ユーザを作成します。
  - a. 次の文を実行して UserA を作成します。

```
CREATE USER UserA IDENTIFIED BY sql;
```

- b. 次の文を実行して UserB を作成します。

```
CREATE USER UserB IDENTIFIED BY sql;
```

## 結果

2 名の新規ユーザ、UserA と UserB が作成されました。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL の場合\) \[1567 ページ\]](#)

次のタスク: [レッスン 2: ユーザ A への権限の付与 \(SQL の場合\) \[1569 ページ\]](#)

## 関連情報

[ユーザの作成 \(SQL Central の場合\) \[1531 ページ\]](#)

[ユーザの作成 \(SQL の場合\) \[1532 ページ\]](#)

## 1.9.1.11.2 レッスン 2: ユーザ A への権限の付与 (SQL の場合)

システム権限とオブジェクトレベル権限をユーザ A に付与します。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

## 手順

1. Interactive SQL で次の文を実行し、BACKUP DATABASE と VALIDATE ANY OBJECT システム権限をユーザ A に付与します。

```
GRANT BACKUP DATABASE, VALIDATE ANY OBJECT TO UserA;
```

2. 次の文を実行し、Employees テーブルでの SELECT および UPDATE オブジェクトレベル権限をユーザ A に管理権限付きで付与します。

```
GRANT SELECT, UPDATE ON GROUPO.EMPLOYEES TO UserA WITH GRANT OPTION;
```

3. これにより、ユーザ A は Employees テーブルから選択および更新を行えます。また、ユーザ A は Employees テーブルからの選択および更新を行う機能を他のユーザに付与することもできます。

## 結果

ユーザ A が BACKUP DATABASE と VALIDATE ANY OBJECT システム権限を持つようになります。この権限は、サンプルデータベースのバックアップに必要です。また、ユーザ A は Employees テーブルでの SELECT および UPDATE オブジェクトレベル権限も持ち、同じ権限を他のユーザに付与できます。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL の場合\) \[1567 ページ\]](#)

前のタスク: [レッスン 1: 次の 2 名のユーザを作成します。UserA と UserB \(SQL\) \[1568 ページ\]](#)

次のタスク: [レッスン 3: ロールを作成してユーザ A に付与する \(SQL の場合\) \[1570 ページ\]](#)

## 関連情報

[システム権限 \[1502 ページ\]](#)

[オブジェクトレベル権限 \[1498 ページ\]](#)

[システム権限の付与 \(SQL Central の場合\) \[1520 ページ\]](#)

[システム権限の付与 \(SQL の場合\) \[1521 ページ\]](#)

### 1.9.1.11.3 レッスン 3: ロールを作成してユーザ A に付与する (SQL の場合)

プロシージャをデバッグし、デバッグプロセス中に見つかったエラーを修正するのに必要な権限を含むロールを作成し、そのロールをユーザ A に付与します。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

## コンテキスト

UserA が必要に応じて他のユーザにロールを付与したり、ロールを取り消したりできるように、ロールの管理権限が付与されていることを確認します。

## 手順

1. Interactive SQL で、次の文を実行してロールを作成します。

```
CREATE ROLE DebugAndFix;
```

2. 次の文を作成して、必要なシステム権限をロールに付与します。

```
GRANT ALTER ANY OBJECT, DEBUG ANY PROCEDURE TO DebugAndFix;
```

3. 次の文を作成して、新しいロールを管理権限付きでユーザ A に付与します。

```
GRANT ROLE DebugAndFix TO UserA WITH ADMIN OPTION;
```

## 結果

DEBUG ANY PROCEDURE と ALTER ANY OBJECT システム権限を持つ DebugAndAlter というロールが作成され、管理権限とともにユーザ A に付与されます。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL の場合\) \[1567 ページ\]](#)

前のタスク: [レッスン 2: ユーザ A への権限の付与 \(SQL の場合\) \[1569 ページ\]](#)

次のタスク: [レッスン 4: ユーザ A をユーザ拡張ロールに変換し、そのロールをユーザ B に付与する \(SQL の場合\) \[1572 ページ\]](#)

## 関連情報

[ユーザ定義ロールの作成 \(SQL Central の場合\) \[1472 ページ\]](#)

[ロールにロールと権限を設定する \(SQL Central の場合\) \[1486 ページ\]](#)

## 1.9.1.11.4 レッスン 4: ユーザ A をユーザ拡張ロールに変換し、そのロールをユーザ B に付与する (SQL の場合)

SQL で、ユーザ A をユーザ拡張ロールに変換し、そのロールをユーザ B に付与します。ユーザ A はロールであると同時に、引き続きデータベースにログインできるユーザでもあります。

### 前提条件

このチュートリアルの前までのレッスンを完了している必要があります。

### コンテキスト

ユーザ B にユーザ A と同じ権限とロールを付与するとします。ユーザ A からユーザ B に付与したのと同じロールと権限の付与と同様の長いプロセスをたどる代わりに、ユーザ A をユーザ拡張ロールに変換し、そのロールをユーザ B に付与できます。ユーザ A はロールであると同時に、引き続きデータベースにログインできるユーザでもあります。

### 手順

1. Interactive SQL で、次の文を実行してユーザ A をユーザ拡張ロールに変換します。

```
CREATE ROLE FOR USER UserA;
```

2. 次の文を実行して、ユーザ拡張ロールユーザ A をユーザ B に付与します。

```
GRANT ROLE UserA TO UserB;
```

これにより、ユーザ B にユーザ A ロールを行使する機能が与えられます。つまり、ユーザ A が行使できる任意のロールおよび権限は、ユーザ B も行使できるということです。

### 結果

ユーザ A がユーザ拡張ロールに変換され、ユーザ B に付与されました。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL の場合\) \[1567 ページ\]](#)

前のタスク: [レッスン 3: ロールを作成してユーザ A に付与する \(SQL の場合\) \[1570 ページ\]](#)

次のタスク: [レッスン 5: ユーザ B のロールと権限を表示 \(SQL の場合\) \[1573 ページ\]](#)

## 関連情報

[ユーザ拡張ロール \[1475 ページ\]](#)

### 1.9.1.11.5 レッスン 5: ユーザ B のロールと権限を表示 (SQL の場合)

他のロールのメンバーシップを介してユーザが継承したのものも含め、ユーザのロール、システム権限、オブジェクトレベル権限を表示します。

## 前提条件

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

1. Interactive SQL で、次の文を実行して管理権限をはじめとしたユーザ B のロールと権限を表示します。

```
CALL sp_displayroles ( 'UserB', 'expand_down' );
```

role_name	parent_role_name	grant_type	role_level
UserA	(NULL)	NO ADMIN	1
PUBLIC	(NULL)	NO ADMIN	1
dbo	PUBLIC	NO ADMIN	2
PUBLIC	UserA	NO ADMIN	2
DebugAndFix	UserA	ADMIN	2

role_name	parent_role_name	grant_type	role_level
BACKUP DATABASE	UserA	NO ADMIN	2
VALIDATE ANY OBJECT	UserA	NO ADMIN	2
ALTER ANY OBJECT	DebugAndFix	NO ADMIN	2
DEBUG ANY PROCEDURE	DebugAndFix	NO ADMIN	3
dbo	PUBLIC	NO ADMIN	3

role\_level 値を継承の階層にあるレベルと考えます。UserB はレベル 1 です。

- role\_level 1 のローから、UserA ロールが UserB に直接付与されたことがわかります。また、PUBLIC ロールが UserB に直接付与されたこともわかります。PUBLIC ロールは自動的に新規ユーザに付与されます。
- role\_level 2 のローから、UserB は UserA から dbo および PUBLIC ロールを継承していることがわかります。また、UserB は、ロールに対する管理権限を含む DebugAndFix ロールも UserA から継承しています。最終的に、UserB は、UserA に付与された BACKUP DATABASE および VALIDATE ANY OBJECT システム権限を継承しません。
- role\_level 3 のローから、DebugAndFix ロールから継承した UserA から、UserB が ALTER ANY OBJECT および DEBUG ANY PROCEDURE システム権限を継承していることがわかります。また、UserB は、今回は作成時に DebugAndFix ロールに自動的に付与された PUBLIC ロールから dbo を再度継承します。

## 2. 次の文を実行して、UserB が持つオブジェクトレベル権限を表示します。

```
CALL sp_objectpermission ( 'UserB' );
```

grantor	grantee
...	...
DBA	UserA
DBA	UserA
SYS	PUBLIC
...	...

ユーザは PUBLIC ロールから継承しているため、結果セットは非常に長くなります。ただし、興味のあるローは上記テーブルの 2 番目と 3 番目のローです。このローは、UserB が Employees テーブルの SELECT および UPDATE 権限を継承していることを反映しているためです。また、これら 2 つの権限がユーザ DBA により付与されたこともわかります。

オブジェクトレベル権限の管理権限は継承不可であるため、DBA が UserA に Employees テーブルでの SELECT および UPDATE 権限を付与する権限を与えた場合でも、UserB は管理権限を継承しません。これらのローの付与可能カラムの Y は、UserA が権限に対する管理権限を持っていることを示します。

## 結果

UserB に付与、および UserB から継承するロールと権限を検証しました。

タスクの概要: [チュートリアル: ロールと権限の付与 \(SQL の場合\) \[1567 ページ\]](#)

前のタスク: [レッスン 4: ユーザ A をユーザ拡張ロールに変換し、そのロールをユーザ B に付与する \(SQL の場合\) \[1572 ページ\]](#)

## 関連情報

[ロールと権限の継承 \[1537 ページ\]](#)

[ロール管理者 \[1481 ページ\]](#)

[システム権限の付与 \(SQL Central の場合\) \[1520 ページ\]](#)

[ロールの付与 \(SQL Central の場合\) \[1489 ページ\]](#)

### 1.9.1.12 権限ベースのセキュリティからロールベースのセキュリティへのアップグレード

データベースを、権限ベースのセキュリティからロールベースのセキュリティにアップグレードすることができます。

この章は、バージョン 16.0 より前のバージョンの SQL Anywhere を使用しているユーザを対象としています。ここでは、ロールベースのセキュリティを使用するためにデータベースをアップグレードするときの変更と、GRANT 文や REVOKE 文などの SQL 文を実行するアプリケーションがある場合に必要な操作について説明します。

このセクションの内容:

#### [権限、パーミッション、グループの変更 \[1576 ページ\]](#)

以前は、権限、パーミッション、オブジェクトレベルのパーミッション、グループがありました。SQL Anywhere 16.0 には、ロール、システム権限、オブジェクトレベル権限およびユーザ拡張ロールを提供するロールベースのセキュリティモデルが導入されました。

#### [権限 \(authority\) は互換ロールに \[1577 ページ\]](#)

REMOTE DBA 権限を除き、データベースをアップグレードするときは、16.0 より前のバージョンのデータベースの権限 (authority) を付与されていたユーザに、その権限と同等の互換ロールが自動的に付与されます。

#### [パーミッションは権限 \(privilege\) に \[1580 ページ\]](#)

バージョン 16.0 では、パーミッションは権限になります。

#### [グループはロールを使用して実現される \[1580 ページ\]](#)

16.0 データベースのアップグレード中に、各グループは、同じ名前のユーザ拡張ロールまたはスタンドアロンロールに変換されます。

#### [データベーススーパーユーザ \(DBA 権限\) の概念への変更 \[1581 ページ\]](#)

16.0 より前のバージョンのデータベースでは、DBA 権限を付与してデータベーススーパーユーザを作成することができました。

#### [GRANT 文の構文の変更 \[1582 ページ\]](#)

アプリケーションで、16.0 より前のバージョンの GRANT 文の構文を権限 (authority)、パーミッション、グループに使用している場合は、ロールと権限 (privilege) のための更新済み GRANT ROLE および GRANT 構文を使用するように変更する必要があります。

### REVOKE 文の構文の変更 [1587 ページ]

アプリケーションで、16.0 より前のバージョンの REVOKE 文の構文を権限 (authority)、パーミッション、グループに使用している場合は、ロールと権限 (privilege) のための更新済み REVOKE ROLE および REVOKE 構文を使用するように変更する必要があります。

### REMOTE DBA への変更 [1591 ページ]

16.0 より前のバージョンのデータベースでは、REMOTE DBA 権限を持つユーザが、dbremote および dbmsync を使用してレプリケーションおよび同期操作を実行することができました。

### ロールとなる一部の権限の継承動作の変更 [1592 ページ]

16.0 より前のバージョンのデータベースでは、DBA、REMOTE DBA、BACKUP、RESOURCE AND VALIDATE の各権限をグループに付与した場合に、基本となるパーミッションはグループのメンバーに継承されませんでした。

### データベースパブリッシャの管理における変更 [1593 ページ]

16.0 より前のバージョンのデータベースでは、GRANT PUBLISH 文と REVOKE PUBLISH 文で PUBLISH 権限を付与して、データベースパブリッシャを制御していました。

### 権限が必要な操作を実行するシステムプロシージャの変更 [1593 ページ]

16.0 より前のバージョンのデータベースでは、権限が必要な操作を行うシステムプロシージャを実行すると、デフォルトで、所有者の権限 (通常は dbo または SYS) で実行され、その定義の SQL SECURITY 句が DEFINER (所有者) に設定されていました。

## 1.9.1.12.1 権限、パーミッション、グループの変更

以前は、権限、パーミッション、オブジェクトレベルのパーミッション、グループがありました。SQL Anywhere 16.0 には、ロール、システム権限、オブジェクトレベル権限およびユーザ拡張ロールを提供するロールベースのセキュリティモデルが導入されました。

SQL Anywhere 16.0 には、ロールベースのセキュリティモデルが導入されています。このモデルでは、以前の権限 (authority)、パーミッション、オブジェクトレベルのパーミッション、グループに代わって、ロール、システム権限、オブジェクトレベルの権限 (privilege)、ユーザ拡張ロールが使用されます。

### i 注記

SQL Anywhere 16.0 データベースサーバは、16.0 より前のバージョンのデータベースと併用できます。この場合は、そのデータベースに対して完全な下位互換性が提供され、そのセキュリティモデルは変更されません。

16.0 より前のバージョンのデータベースでは、権限 (authority) はデータベースレベルのパーミッションでした。たとえば、バックアップ権限を持つユーザはデータベースをバックアップできます。一部の権限には、オブジェクトレベルのパーミッションも付属しています。たとえば、PROFILE 権限を持つユーザは、データベーストレーニングのタスクを実行できます。これに使用するシステムプロシージャは、この権限のないユーザには使用できません。新しい権限の作成、権限を構成するパーミッションの変更と削除は実行できません。管理権限を付与する (WITH GRANT) ことはできますが、管理者であることのみ限定して付与することはできません。

ロールは、機能面で権限 (authority) に代わるものであり、さらに、新しいロールの作成、ロールを構成する権限 (privilege) の変更、およびロールの削除が可能であるという利点を備えています。ロールと権限 (privilege) を使用すると、ユーザに付与する権限をさらにきめ細かく制御でき、他のユーザに対してより容易に付与できるようになります。また、管理権限のみのロールをユーザに付与することもできます。つまり、ユーザは、ロールの付与と取り消しを行うことはできますが、その基本となる権限を行使することはできません。

16.0 より前のバージョンのデータベースでは、パーミッションがあれば、テーブル、ビュー、ユーザなどのデータベースオブジェクトの作成、変更、クエリ、使用、または削除を行うことができました。たとえば、テーブルに対する SELECT パーミッションなどです。

権限 (privilege) は、機能面でパーミッションに代わるものであり、さらに、パーミッションより数多くの権限があるという利点があります。データベースオブジェクトに実行できる、権限が必要な操作にはそれぞれ、付与可能な権限があります。ユーザには、権限を個別に付与するか、ロールを付与することができます。パーミッションという用語は、なくなったわけではありませんが、意味が少し変わりました。パーミッションという用語の以前の意味は、ユーザに付与できる能力でした。現在、パーミッションとは、操作を実行できるかどうかを評価した結果を意味しています。たとえば、所有者であればテーブルを変更できるパーミッションがある場合や、ALTER ANY TABLE システム権限がある場合があります。

16.0 より前のバージョンのデータベースでは、グループとは、グループレベルの設定で権限 (authority) とパーミッションが決定される 1 人以上のユーザの集合でした。あるユーザにグループステータスが付与され、その後、そのグループのメンバーシップが他のユーザに付与されます。

現在は、グループのパラダイムはユーザ拡張ロールで実現されます。あるユーザが持っている権限 (privilege) のセットを他のユーザに付与したい場合は、ユーザ拡張ロールになるようにユーザを拡張してから、そのロールを他のユーザに付与します。グループのパラダイムは、ユーザに付与したいロールと権限を持ち、ユーザ ID と関連付けられないスタンドアロンロールを作成して実現することもできます。

16.0 のデータベースをアップグレードするときは、アップグレード処理によって既存の権限 (authority)、パーミッション、およびグループの階層構造が自動的に、同等のロール、権限 (privilege)、およびユーザ拡張ロールの階層構造に変換されます。16.0 より前のバージョンの権限 (authority) にはそれぞれ、同等の互換ロールがあります。これらのロールは名前が SYS\_AUTH で始まるため、簡単に識別可能です。互換ロールには、16.0 より前のバージョンのユーザが、権限 (authority) を使用して実行していた操作と同じ操作を実行するために必要なシステム権限が含まれています。

## 関連情報

[ロール \[1455 ページ\]](#)

[権限 \[1496 ページ\]](#)

[ユーザ拡張ロール \[1475 ページ\]](#)

### 1.9.1.12.2 権限 (authority) は互換ロールに

REMOTE DBA 権限を除き、データベースをアップグレードするときは、16.0 より前のバージョンのデータベースの権限 (authority) を付与されていたユーザに、その権限と同等の互換ロールが自動的に付与されます。

ユーザが、以前の権限を管理する能力を持っていた場合、そのユーザは、互換ロールを管理する能力を持ちます。

互換ロールは変更できませんが、ユーザ定義ロールに移行してから変更することはできます。互換ロールの移行は簡単です。また、後で互換ロールをリストアすることも簡単です。互換ロールをユーザ定義ロールに移行すると、その互換ロールを付与されていたすべてのユーザには、新しいユーザ定義ロールが自動的に付与されます。互換ロールは、移行されると、自動的に削除されます。ただし、互換ロールは、CREATE ROLE 文を使用してリストアできます。

SQL 文の下位互換性は、権限の付与や取り消しを行うアプリケーションが機能し続けるように提供されています。ただし、古い構文は使用されなくなるため、新しい SQL 構文をロールに使用するようにアプリケーションを変更することを考慮する必要があります。

次の表は、16.0 より前のバージョンの権限と、それがデータベースのアップグレード後にどのロールになるかを示しています。

16.0 より前のバージョンの権限	同等のロール	説明
ALL	SYS_AUTH_RESOURCE_ROLE 互換ロール	ユーザはテーブル、ビュー、ストアドプロシージャ、トリガなどのデータベースオブジェクトを作成できます。
BACKUP	SYS_AUTH_BACKUP_ROLE 互換ロール	ユーザは、BACKUP DATABASE 文または dbbackup ユーティリティを使用して、データベースとトランザクションログをアーカイブバックアップまたはイメージバックアップでバックアップできます。
DBA	SYS_AUTH_DBA_ROLE 互換ロール SYS_AUTH_SA_ROLE 互換ロール SYS_AUTH_SSO_ROLE 互換ロール	<p>このロールでユーザは、権限 (privilege) が必要で実行可能な操作をすべて実行できます。SYS_AUTH_DBA_ROLE 互換性ロールを付与されたユーザは、データベースオブジェクトの作成とこれらのオブジェクトの所有権の他のユーザ ID への割り当て、テーブル構造の変更、新規ユーザ ID の作成、ユーザからのパーミッションの取り消し、データベースのバックアップなどを実行できます。</p> <p>SYS_AUTH_DBA_ROLE 互換ロールで実行できる、権限の必要な操作のうち、ユーザは SYS_AUTH_SA_ROLE 互換ロールで、テーブルの作成やデータのバックアップなど、データベース管理関連作業をすべて実行できます。</p> <p>SYS_AUTH_DBA_ROLE 互換ロールで実行できる、権限の必要な操作のうち、ユーザは SYS_AUTH_SSO_ROLE 互換ロールで、ユーザの作成やオブジェクトに関する権限の付与など、セキュリティおよびアクセス関連の管理作業を実行できます。</p>
PROFILE	SYS_AUTH_PROFILE_ROLE 互換ロール	このロールでユーザは、プロファイリング、トレーシング、診断の各操作を実行できます。
READCLIENTFILE	SYS_AUTH_READCLIENTFILE_ROLE 互換ロール	このロールでユーザは、クライアントコンピュータ上のファイルを読み込みます。たとえば、クライアントコンピュータ上のファイルからデータをロードできます。
READFILE	SYS_AUTH_READFILE_ROLE 互換ロール	このロールでユーザは、SELECT 文で OPENSTRING 句を使用してファイルを読み込みます。

16.0 より前のバージョンの権限	同等のロール	説明
REMOTE DBA	SYS_RUN_REPLICATION_ROLE システムロール	このロールで、SQL リモートユーザは dbremote ユーティリティを使用してレプリケーション作業を、Mobile Link ユーザは dbmlsync ユーティリティを使用して同期作業を実行できます。ただし、レプリケーションを管理することはできません。  また、レプリケーション管理者に付与できる新しいシステムロール、SYS_REPLICATION_ADMIN_ROLE もあります。16.0 より前のバージョンのデータベースでは、レプリケーション管理が管理タスクを実行するために DBA 権限が必要でした。SYS_REPLICATION_ADMIN_ROLE システムロールは、これらの管理タスクを実行するために必要な権限を網羅しています。
RESOURCE	SYS_AUTH_RESOURCE_ROLE 互換ロール	ユーザはテーブル、ビュー、ストアドプロシージャ、トリガなどのデータベースオブジェクトを作成できます。
VALIDATE	SYS_AUTH_VALIDATE_ROLE 互換ロール	このロールでユーザは、VALIDATE 文または dbvalid ユーティリティを使用して、データベース、テーブル、インデックス、チェックサムの検証を実行できます。
WRITECLIENTFILE	SYS_AUTH_WRITECLIENTFILE_ROLE 互換ロール	このロールでユーザは、クライアントコンピュータ上にファイルを書き込みます。たとえば、UNLOAD TABLE 文を使用してクライアントコンピュータにデータを書き込みます。
WRITEFILE	SYS_AUTH_WRITEFILE_ROLE 互換ロール	このロールでユーザは、xp_write_file システムプロシージャを実行できます。

## 関連情報

[ロール \[1455 ページ\]](#)

[レプリケーションに関連するシステムロール \[1457 ページ\]](#)

[GRANT 文の構文の変更 \[1582 ページ\]](#)

[REVOKE 文の構文の変更 \[1587 ページ\]](#)

[ロールとなる一部の権限の継承動作の変更 \[1592 ページ\]](#)

### 1.9.1.12.3 パーミッションは権限 (privilege) に

バージョン 16.0 では、パーミッションは権限になります。

16.0 より前のバージョンのデータベースでは、テーブルやビューに対する ALTER や SELECT など、オブジェクトレベルのパーミッションがありました。これらのパーミッションの付与または取り消しを行う文は引き続き機能しますが、これらのパーミッションは、同じ名前を保持しながら、権限と呼ばれます。オブジェクトレベルの権限に加え、実行するために認証が必要なすべての操作には、付与可能なシステム権限があります。データベースをアップグレードすると、パーミッションを持っていたユーザーは、自動的にアップグレードされて、以前実行できたタスクを実行するために必要な同等の権限を持つようになります。

#### 関連情報

[権限 \[1496 ページ\]](#)

### 1.9.1.12.4 グループはロールを使用して実現される

16.0 データベースのアップグレード中に、各グループは、同じ名前のユーザー拡張ロールまたはスタンドアロンロールに変換されます。

元のグループのメンバーには自動的に、新しいロールとその基本となる権限 (privilege) が付与されます。元のグループに付与されていた権限 (authority) とオブジェクトレベルのパーミッションは、その同等のロールおよびシステム権限に変換され、ユーザー拡張ロールに付与されます。

権限 (authority) が継承可能であった場合は、互換ロールが、新しいユーザー拡張ロールの被付与者に継承されます。権限 (authority) が継承不可であった場合、ユーザー拡張ロールの被付与者は互換ロールを継承しません。

次の表は、システムユーザーおよびグループと、それらの変換後のロールを示しています。

16.0 より前のバージョンのグループ	ロール	説明
dbo	dbo	このグループは、Ultra Light と Mobile Link に使用される多くのシステムストアブプロシージャ、ビュー、およびテーブルを所有しています。
DIAGNOSTICS	DIAGNOSTICS	このロールは、診断テーブルおよびビューを所有しており、それらに操作を実行できます。
PUBLIC	PUBLIC	このロールには、システムテーブルに対する SELECT パーミッションがあります。新しいユーザー ID にはいずれも、PUBLIC ロールが自動的に付与されます。
SA_DEBUG	SA_DEBUG	このロールでユーザーは、SQL Anywhere デバッガを使用できるようになります。

16.0 より前のバージョンのグループ	ロール	説明
SYS	SYS	このロールは、データベースのシステムテーブルおよびビュー (カタログ) を所有しており、それらに操作を実行できます。
SYS_SPATIAL_ADMIN_ROLE	SYS_SPATIAL_ADMIN_ROLE	このロールでユーザは、空間オブジェクトの作成、変更、削除が可能になります。

グループを作成して管理するための SQL 構文は変更されましたが、以前の GRANT 構文と REVOKE 構文は引き続きサポートされています。

## 関連情報

[グループ \[1536 ページ\]](#)

[ロール \[1455 ページ\]](#)

[REVOKE 文の構文の変更 \[1587 ページ\]](#)

[GRANT 文の構文の変更 \[1582 ページ\]](#)

### 1.9.1.12.5 データベーススーパーユーザ (DBA 権限) の概念への変更

16.0 より前のバージョンのデータベースでは、DBA 権限を付与してデータベーススーパーユーザを作成することができました。

DBA 権限を持つユーザは、データベースで権限の必要なあらゆるタスクを実行できました。データベースをアップグレードすると、DBA 権限を持っていたユーザはすべて、SYS\_AUTH\_DBA\_ROLE 互換ロールを取得し、アップグレード時に存在しているすべてのロールと権限の行使権限と管理権限を自動的に受け取ります。

また、新しいロールを作成して、作成時に管理者を指定しない場合は、MANAGE ROLES システム権限を持つユーザ (グローバル管理者) がロールを管理できます。

ただし、新しいロールを作成して、ロール作成の一部として管理者を割り当てた場合は、管理が、管理権限を与えられた被付与者に限定されます。したがって、新しいロールの管理権限をスーパーユーザに持たせる場合は、明示的に付与する必要があります。

## 関連情報

[スーパーユーザ \[1484 ページ\]](#)

[ロール管理者 \[1481 ページ\]](#)

## 1.9.112.6 GRANT 文の構文の変更

アプリケーションで、16.0 より前のバージョンの GRANT 文の構文を権限 (authority)、パーミッション、グループに使用している場合は、ロールと権限 (privilege) のための更新済み GRANT ROLE および GRANT 構文を使用するように変更する必要があります。

下の表に、文をどのように変更すべきかを示します。権限、パーミッション、グループのための古い GRANT 構文の使用は、サポートされていますが、廃止される予定です。

### i 注記

16.0 以前のデータベースでは、DBA、REMOTE DBA、BACKUP、RESOURCE、VALIDATE 権限は、グループとして活動するユーザに割り当てられている場合、非継承でした。そのグループのメンバーは、権限の機能を継承しませんでした。

16.0 以前のデータベースをアップグレードするとき、同等のロールが自動的に付与され (たとえば、BACKUP 権限の代わりに SYS\_AUTH\_BACKUP\_ROLE が付与されます)、継承動作が以前のリリースと整合性を持つように WITH NO SYSTEM PRIVILEGE INHERITANCE 句が指定されます。グループのメンバーは、そのロールが提供する権限を継承しません。

同様に、廃止予定の付与構文を使用して DBA、REMOTE DBA、BACKUP、RESOURCE、VALIDATE を付与し続けると、非継承の古い動作が維持されます。つまり、同等のロールが付与され、WITH NO SYSTEM PRIVILEGE INHERITANCE 句が指定されず、これは、自動的に行われます。

16.0 より前のバージョンのデータベースでは、DBA 権限と REMOTE DBA 権限を付与されたユーザは自動的に、これらの権限の他のユーザへの付与が可能になっていました。推奨される次の新しい構文の WITH ADMIN OPTION 句を使用すると、管理権限の動作で、以前のリリースとの一貫性が確実に保たれます。

廃止予定の構文	新しい構文
<code>GRANT CONNECT TO username [ IDENTIFIED BY ] pwd</code>	変更ありません。
<code>GRANT GROUP TO user</code>	<code>CREATE OR REPLACE ROLE groupname FOR USER userid</code>
<code>GRANT MEMBERSHIP IN GROUP groupname [, ...] TO grantee [, ...]</code>	<code>GRANT ROLE groupname [, ...] TO grantee [, ...]</code>
<code>GRANT DBA TO grantee [, ...]</code>	<code>GRANT ROLE SYS_AUTH_SA_ROLE TO grantee [, ...] WITH ADMIN OPTION WITH NO SYSTEM PRIVILEGE INHERITANCE</code>  <code>GRANT ROLE SYS_AUTH_SSO_ROLE TO grantee [, ...] WITH ADMIN OPTION WITH NO SYSTEM PRIVILEGE INHERITANCE</code>

廃止予定の構文	新しい構文
<code>GRANT REMOTE DBA TO grantee [,...]</code>	<code>GRANT ROLE SYS_RUN_REPLICATION_ROLE TO grantee [,...] WITH ADMIN OPTION WITH NO SYSTEM PRIVILEGE INHERITANCE</code>
<code>GRANT BACKUP TO grantee [,...]</code>	<code>GRANT ROLE SYS_AUTH_BACKUP_ROLE TO grantee [,...] WITH NO SYSTEM PRIVILEGE INHERITANCE</code>
<code>GRANT RESOURCE TO grantee [,...]</code>	<code>GRANT ROLE SYS_AUTH_RESOURCE_ROLE TO grantee [,...] WITH NO SYSTEM PRIVILEGE INHERITANCE</code>
<code>GRANT VALIDATE TO grantee [,...]</code>	<code>GRANT ROLE SYS_AUTH_VALIDATE_ROLE TO grantee [,...] WITH NO SYSTEM PRIVILEGE INHERITANCE</code>
<code>GRANT PROFILE TO grantee [,...]</code>	<code>GRANT ROLE SYS_AUTH_PROFILE_ROLE TO TO grantee [,...]</code>
<code>GRANT READCLIENTFILE TO grantee [,...]</code>	<code>GRANT ROLE SYS_AUTH_READCLIENTFILE_ROLE TO TO grantee [,...]</code>
<code>GRANT READFILE TO grantee [,...]</code>	<code>GRANT ROLE SYS_AUTH_READFILE_ROLE TO TO grantee [,...]</code>
<code>GRANT WRITECLIENTFILE TO grantee [,...]</code>	<code>GRANT ROLE SYS_AUTH_WRITECLIENTFILE_ROLE TO TO grantee [,...]</code>
<code>GRANT PUBLISH TO grantee</code>	変更ありません。ただし、新しい PUBLIC オプションである db_publisher も設定できます。  <code>SET OPTION PUBLIC.db_publisher=grantee</code>
<code>GRANT permission [,...] ON [ owner.]object-name TO grantee [,...]</code>  permission : ALL [ PRIVILEGES ]   ALTER   DELETE   INSERT   REFERENCES [ ( column-name, ... ) ]   SELECT [ ( column-name, ... ) ]   UPDATE [ ( column-name, ... ) ]	変更ありません。

廃止予定の構文	新しい構文
<code>GRANT EXECUTE ON [ owner.]{ proc   user-def-funct } TO grantee [,...]</code>	変更ありません。
<code>GRANT INTEGRATED LOGIN TO user-profile-name [,...] AS USER user</code>	変更ありません。
<code>GRANT KERBEROS LOGIN TO userid [,...] AS USER user</code>	変更ありません。
<code>GRANT CREATE ON dbspaceName [,...] TO grantee [,...]</code>	変更ありません。

このセクションの内容:

[GRANT 文 \(権限とグループ\) \(廃止予定\) \[1584 ページ\]](#)

新しい GRANT ステートメントを使用して、システム権限 (以前はパーミッション) を付与し、GRANT ROLE ステートメントでロール (以前は権限とグループ) を付与します。

## 関連情報

[互換ロール \[1459 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

### 1.9.1.12.6.1 GRANT 文 (権限とグループ) (廃止予定)

新しい GRANT ステートメントを使用して、システム権限 (以前はパーミッション) を付与し、GRANT ROLE ステートメントでロール (以前は権限とグループ) を付与します。

#### 構文 1 - 権限の付与

```
GRANT authority, ...
TO userid, ...
```

```
authority :
BACKUP
| DBA
| PROFILE
```

```
| READCLIENTFILE  
| READFILE  
| [ RESOURCE | ALL ]  
| VALIDATE  
| WRITECLIENTFILE
```

## 構文 2 - ユーザへの、グループステータスまたはグループのメンバーシップの付与

```
GRANT { GROUP | MEMBERSHIP IN GROUP userid, ... }  
TO userid, ...
```

### パラメータ

#### authority

##### **i** 注記

廃止予定の古い GRANT 構文を使用して権限 (authority) を付与すると、権限が、データベースサーバによって同等の互換ロールに変換されます。同等の互換ロールを付与されたユーザにとって、機能が失われることはありません。

#### **BACKUP authority**

SYS\_AUTH\_BACKUP\_ROLE 互換ロールが付与されるように変換されます。

#### **DBA authority**

SYS\_AUTH\_DBA\_ROLE、SYS\_AUTH\_SA\_ROLE、および SYS\_AUTH\_SSO\_ROLE 互換ロールが付与されるように変換されます。

#### **PROFILE authority**

SYS\_AUTH\_PROFILE\_ROLE 互換ロールが付与されるように変換されます。

#### **READ CLIENT FILE privilege**

SYS\_AUTH\_READCLIENTFILE\_ROLE 互換ロールが付与されるように変換されます。

#### **READ FILE privilege**

SYS\_AUTH\_READFILE\_ROLE 互換ロールが付与されるように変換されます。

#### **RESOURCE or ALL authority**

SYS\_AUTH\_RESOURCE\_ROLE 互換ロールが付与されるように変換されます。

#### **VALIDATE authority**

SYS\_AUTH\_VALIDATE 互換ロールが付与されるように変換されます。

#### **WRITECLIENTFILE authority**

SYS\_AUTH\_WRITECLIENTFILE\_ROLE 互換ロールが付与されるように変換されます。

#### **GROUP clause**

これを付与すると、ユーザが、他のユーザに付与可能なユーザ拡張ロールに変換されます。

#### **MEMBERSHIP IN GROUP clause**

これを付与すると、被付与者に、付与されているユーザ拡張ロール (グループ) のすべてのロールと権限が与えられます。

## 備考

権限 (authority) を付与するための構文は廃止される予定ですが、下位互換性を確保するためにサポートされます。権限 (authority) を付与すると、バージョン 16.0 以降で提供されている同等の互換ロールが付与されます。ただし、互換ロールを移行している場合は、廃止予定の構文を使用して権限 (authority) を付与すると、処理に失敗します。たとえば、SYS\_AUTH\_BACKUP\_ROLE 互換ロールをユーザ定義ロールに移行 (移行が完了すると、互換ロールが削除される) してから GRANT BACKUP TO UserA を実行した場合は、SYS\_AUTH\_BACKUP\_ROLE 互換ロールがデータベースに存在していないため、この文は失敗します。この問題は、互換ロールをリストアする (CREATE ROLE SYS\_AUTH\_BACKUP\_ROLE) と解決できます。または、新しい構文を使用し、移行して作成した新しいユーザ定義ロールを参照するようにアプリケーションを変更します。

### i 注記

16.0 以前のデータベースでは、DBA、REMOTE DBA、BACKUP、RESOURCE、VALIDATE 権限は、グループとして活動するユーザに割り当てられている場合、非継承でした。そのグループのメンバーは、権限の機能を継承しませんでした。

16.0 以前のデータベースをアップグレードするとき、同等のロールが自動的に付与され (たとえば、BACKUP 権限の代わりに SYS\_AUTH\_BACKUP\_ROLE が付与されます)、継承動作が以前のリリースと整合性を持つように WITH NO SYSTEM PRIVILEGE INHERITANCE 句が指定されます。グループのメンバーは、そのロールが提供する権限を継承しません。

同様に、廃止予定の付与構文を使用して DBA、REMOTE DBA、BACKUP、RESOURCE、VALIDATE を付与し続けると、非継承の古い動作が維持されます。つまり、同等のロールが付与され、WITH NO SYSTEM PRIVILEGE INHERITANCE 句が指定されます。これは、自動的に行われます。

## 権限

付与している同等のロールに対する管理権限が必要です。たとえば、データベースをバックアップする機能を付与するために GRANT BACKUP 構文を使用している場合は、SYS\_AUTH\_BACKUP\_ROLE 互換ロールに対する管理権限が必要です。

## 関連情報

[互換ロール \[1459 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

[12.0.1 GRANT statement \(12.0.1 GRANT 文\)](#)

## 1.9.112.7 REVOKE 文の構文の変更

アプリケーションで、16.0 より前のバージョンの REVOKE 文の構文を権限 (authority)、パーミッション、グループに使用している場合は、ロールと権限 (privilege) のための更新済み REVOKE ROLE および REVOKE 構文を使用するように変更する必要があります。

下の表に、文をどのように変更すべきかを示します。権限、パーミッション、グループのための古い REVOKE 構文の使用は、サポートされていますが、廃止される予定です。

16.0 より前のバージョンの構文	新しい構文
<code>REVOKE CONNECT FROM userid</code>	変更ありません。
<code>REVOKE GROUP FROM userid</code>	<code>DROP ROLE rolename FROM USER user WITH REVOKE</code>
<code>REVOKE MEMBERSHIP IN GROUP groupname [,...] FROM grantee [,...]</code>	<code>REVOKE ROLE groupname [,...] FROM grantee [,...]</code>
<code>REVOKE authority FROM grantee [,...]</code>  authority : DBA   REMOTE DBA   BACKUP   RESOURCE   VALIDATE   PROFILE   READCLIENTFILE   READFILE   WRITECLIENTFILE	<code>REVOKE ROLE rolename [,...] FROM userid [,...]</code>  role : SYS_AUTH_DBA_ROLE   SYS_RUN_REPLICATION_ROLE   SYS_AUTH_BACKUP_ROLE   SYS_AUTH_RESOURCE_ROLE   SYS_AUTH_VALIDATE_ROLE   SYS_AUTH_PROFILE_ROLE   SYS_AUTH_READCLIENTFILE_ROLE   SYS_AUTH_READFILE_ROLE   SYS_AUTH_WRITECLIENTFILE_ROLE
<code>REVOKE PUBLISH FROM grantee</code>	変更ありません。ただし、新しい PUBLIC オプションである db_publisher も設定できます。  <code>SET OPTION PUBLIC.db_publisher=grantee</code>

16.0 より前のバージョンの構文	新しい構文
<pre>REVOKE permission [,...] ON [ owner.]object-name FROM grantee [,...]</pre> <pre>permission : ALL [ PRIVILEGES ]   ALTER   DELETE   INSERT   REFERENCES [ ( column-name, ... ) ]   SELECT [ ( column-name, ... ) ]   UPDATE [ ( column-name, ... ) ]</pre>	命名規則を除き、変更はありません。オブジェクトレベルのパーミッションは、オブジェクトレベルの権限 (privilege) となります。
<pre>REVOKE EXECUTE ON [ owner.]{ procedure- name   user-defined-function } FROM grantee [,...]</pre>	変更ありません。
<pre>REVOKE INTEGRATED LOGIN FROM user- profile-name [,...]</pre>	変更ありません。
<pre>REVOKE KERBEROS LOGIN FROM userid [,...] AS USER user</pre>	変更ありません。
<pre>REVOKE CREATE ON dbspacename [,...] FROM grantee [,...]</pre>	変更ありません。

SQL Anywhere 12.0.1 REVOKE 文の構文を確認するには、[REVOKE statement](#) を参照してください。

このセクションの内容:

[REVOKE 文 \(権限とグループ\) \(廃止予定\) \[1589 ページ\]](#)

新しい REVOKE 文を使用して、システム権限 (以前はパーミッション) を取り消し、REVOKE ROLE 文でロール (以前は権限とグループ) を取り消します。

## 関連情報

[互換ロール \[1459 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL の場合\) \[1534 ページ\]](#)

## 1.9.112.7.1 REVOKE 文 (権限とグループ) (廃止予定)

新しい REVOKE 文を使用して、システム権限 (以前はパーミッション) を取り消し、REVOKE ROLE 文でロール (以前は権限とグループ) を取り消します。

### i 注記

廃止予定の REVOKE 構文を使用して権限を取り消すと、データベースサーバによって、廃止内容が同等の互換ロールに変換されます。

### 構文

```
REVOKE privilege, ... FROM userid, ...
```

```
permission :  
BACKUPCONNECT  
|  
| CREATE ON dbspace  
| DBA  
| GROUPINTEGRATED LOGIN  
| KERBEROS LOGIN  
|  
| MEMBERSHIP IN GROUP userid, ...  
| PROFILE  
| READCLIENTFILE  
| READFILE  
| [ RESOURCE | ALL ]  
| VALIDATE  
| WRITECLIENTFILE
```

### パラメータ

#### authority

### i 注記

廃止予定の古い REVOKE 構文を使用して権限 (authority) を取り消すと、権限が、データベースサーバによって同等の互換ロールに変換されます。

#### BACKUP authority

SYS\_AUTH\_BACKUP\_ROLE 互換ロールが取り消されるように変換されます。

#### DBA authority

SYS\_AUTH\_DBA\_ROLE、SYS\_AUTH\_SA\_ROLE、および SYS\_AUTH\_SSO\_ROLE 互換ロールが取り消されるように変換されます。

#### PROFILE authority

SYS\_AUTH\_PROFILE\_ROLE 互換ロールが取り消されるように変換されます。

**READ CLIENT FILE privilege**

SYS\_AUTH\_READCLIENTFILE\_ROLE 互換ロールが取り消されるように変換されます。

**READ FILE privilege**

SYS\_AUTH\_READFILE\_ROLE 互換ロールが取り消されるように変換されます。

**RESOURCE or ALL authority**

SYS\_AUTH\_RESOURCE\_ROLE 互換ロールが取り消されるように変換されます。

**VALIDATE authority**

SYS\_AUTH\_VALIDATE\_ROLE 互換ロールが取り消されるように変換されます。

**WRITECLIENTFILE authority**

SYS\_AUTH\_WRITECLIENTFILE\_ROLE 互換ロールが取り消されるように変換されます。

#### **GROUP clause**

ユーザ拡張ロール (グループ) を変換して通常のユーザに戻します。グループではなくなるため、ユーザを追加することはできなくなります。

**MEMBERSHIP IN GROUP clause**

ユーザ拡張ロール (グループ) のメンバーシップを取り消します。

## 備考

REVOKE GROUP は、グループのすべてのメンバーから MEMBERSHIP IN GROUP を自動的に取り消します。

グループにユーザを追加すると、ユーザはそのグループに割り当てられたすべての権限 (privilege) を継承します。ユーザがグループのメンバーとして継承する権限のサブセットを取り消すことはできません。取り消すことができるのは、GRANT 文によって明示的に付与される権限だけです。ユーザごとに異なる権限を付与する必要がある場合は、適切な権限を持つそれぞれのグループを作成するか、必要な権限を各ユーザに明示的に付与します。

ユーザへの権限を取り消し、その権限に WITH GRANT OPTION があった場合は、そのユーザが権限を付与していたすべてのユーザへの権限も取り消されます。たとえば、UserA SELECT...WITH GRANT OPTION 権限をテーブルと UserA に付与し、UserB にそのテーブルの SELECT 権限を付与します。UserA から SELECT 権限を取り消すと、UserB からも取り消されます。

SQL Anywhere 12.0.1 REVOKE 文の構文を確認するには、[REVOKE statement](#) を参照してください。

## 権限

取り消している同等のロールに対する管理権限が必要です。たとえば、データベースをバックアップする機能を取り消すために REVOKE BACKUP 構文を使用している場合は、SYS\_AUTH\_BACKUP\_ROLE 互換ロールに対する管理権限が必要です。

## 関連情報

[互換ロール \[1459 ページ\]](#)

[ユーザまたはロールのロールと権限を表示する \(SQL Central の場合\) \[1533 ページ\]](#)

### 1.9.1.12.8 REMOTE DBA への変更

16.0 より前のバージョンのデータベースでは、REMOTE DBA 権限を持つユーザが、dbremote および dbmlsync を使用してレプリケーションおよび同期操作を実行することができました。

この REMOTE DBA 権限に代わって、SYS\_RUN\_REPLICATION\_ROLE システムロールが使用されます。REMOTE DBA の代わりにこのロールを付与するようにアプリケーションを変更してください。GRANT REMOTE DBA 文の構文は引き続きサポートされますが、廃止される予定です。

もう1つのレプリケーション関連のロールとして、SYS\_REPLICATION\_ADMIN\_ROLE システムロールが導入されています。このロールを持つユーザは、レプリケーションを管理できます。

このセクションの内容:

[GRANT REMOTE DBA 文 \(廃止予定\) \[1591 ページ\]](#)

GRANT REMOTE DBA は、レプリケーションを行う dbremote ユーティリティと同期を行う dbmlsync ユーティリティの実行に必要な SYS\_RUN\_REPLICATION\_ROLE システムロールを付与します。

## 関連情報

[レプリケーションに関連するシステムロール \[1457 ページ\]](#)

### 1.9.1.12.8.1 GRANT REMOTE DBA 文 (廃止予定)

GRANT REMOTE DBA は、レプリケーションを行う dbremote ユーティリティと同期を行う dbmlsync ユーティリティの実行に必要な SYS\_RUN\_REPLICATION\_ROLE システムロールを付与します。

#### 構文

```
GRANT REMOTE DBA
TO userid, ...
[ IDENTIFIED BY password ]
```

## パラメータ

### IDENTIFIED BY 句

IDENTIFIED BY 句は、この文のオプションです。含めると、ユーザのパスワードが変更されます。

## 備考

この文を実行すると、SYS\_RUN\_REPLICATION\_ROLE システムロールが付与されます。REMOTE DBA 権限は廃止される予定です。

プロシージャの定義は SYSPROCEDURE システムビューに表示されるため、この文をプロシージャ内で使用する場合は、文字列リテラルとしてパスワードを指定しないでください。セキュリティ保護のため、プロシージャ定義の外部で宣言される変数を使用してパスワードを指定してください。

## 関連情報

[レプリケーションに関連するシステムロール \[1457 ページ\]](#)

### 1.9.1.12.9 ロールとなる一部の権限の継承動作の変更

16.0 より前のバージョンのデータベースでは、DBA、REMOTE DBA、BACKUP、RESOURCE AND VALIDATE の各権限をグループに付与した場合に、基本となるパーミッションはグループのメンバーに継承されませんでした。

16.0 以降では、あらゆるロール (SYS\_AUTH\_DBA\_ROLE、SYS\_RUN\_REPLICATION\_ROLE、SYS\_AUTH\_BACKUP\_ROLE、SYS\_AUTH\_RESOURCE\_ROLE、および SYS\_AUTH\_VALIDATE\_ROLE を含む) をユーザ定義ロールに付与したときに、デフォルトの動作として、ユーザ定義ロールを付与されたユーザが、このロールの基本となるシステム権限を継承します。

たとえば、userA というユーザがいるとします。この userA に ALTER ANY OBJECT システム権限を付与します。そして、userA がロールになるように拡張することにし、userA を userB に付与します。そして、SYS\_AUTH\_DBA\_ROLE 互換性ロールを userA に付与したいのですが、userB には、SYS\_AUTH\_DBA\_ROLE 互換性ロールで与えられる権限の一部を継承させたくありません。そこで、次のように SYS\_AUTH\_DBA\_ROLE 互換性ロールを付与します。

```
GRANT ROLE SYS_AUTH_DBA_ROLE TO userA WITH NO SYSTEM PRIVILEGE INHERITANCE;
```

このシナリオでは、userB は ALTER ANY OBJECT システム権限のみを userA から継承します。

アップグレード時に、これらのロールの非継承動作を保持するために、GRANT ROLE 文で WITH NO SYSTEM PRIVILEGE INHERITANCE 句がサポートされています。同様に、新しい GRANT 構文を使用するようにアプリケーションを変更する場合は、この句も指定する必要があります。この句は、これらの特定のロール専用です。

## 関連情報

[互換ロール \[1459 ページ\]](#)

[互換ロールの付与 \(SQL Central の場合\) \[1465 ページ\]](#)

[互換ロールの付与 \(SQL の場合\) \[1466 ページ\]](#)

### 1.9.1.12.10 データベースパブリッシャの管理における変更

16.0 より前のバージョンのデータベースでは、GRANT PUBLISH 文と REVOKE PUBLISH 文で PUBLISH 権限を付与して、データベースパブリッシャを制御していました。

現在のパブリッシャは、CURRENT PUBLISHER 特別値のクエリによって決定できます。

PUBLISH 権限に代わって PUBLIC.db\_publisher データベースオプションが使用されるようになりました。このオプションを設定するには、SET ANY SYSTEM OPTION システム権限が必要です。パブリッシャは、このデータベースオプションを変更して変更できますが、下位互換性のために、GRANT PUBLISH と REVOKE PUBLISH で変更する機能も残されています。また、CURRENT PUBLISHER のクエリを実行して、現在のパブリッシャを見つけることもできます。

## 関連情報

[db\\_publisher オプション \[687 ページ\]](#)

### 1.9.1.12.11 権限が必要な操作を実行するシステムプロシージャの変更

16.0 より前のバージョンのデータベースでは、権限が必要な操作を行うシステムプロシージャを実行すると、デフォルトで、所有者の権限 (通常は dbo または SYS) で実行され、その定義の SQL SECURITY 句が DEFINER (所有者) に設定されていました。

バージョン 16.0 では、このようなシステムプロシージャの SQL SECURITY 句はデフォルトで INVOKER に設定されます。つまり、権限の必要な操作が、定義者ではなく、呼び出し側の権限で実行されます。

現在は、データベースを作成またはアップグレードするときに、権限の必要な操作を行うシステムプロシージャが、所有者と呼び出し側のどちらの権限で実行されるかを指定できます。CREATE DATABASE 文、ALTER DATABASE UPGRADE 文、初期化ユーティリティ (dbinit)、およびアップグレードユーティリティ (dbupgrad) は、この選択が可能であるように強化されました。

新しいデータベースでは、何も指定していない場合は、これらのシステムプロシージャに対して呼び出し側が選択されます。アップグレード時のデフォルトは、何も指定しない場合は、アップグレードしているデータベースですでに選択されているものです。

## 関連情報

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

[アップグレードユーティリティ \(dbupgrad\) \[1197 ページ\]](#)

## 1.9.2 データのセキュリティ

データベースには機密情報や個人情報などが含まれている場合があるので、データベースや格納されるデータのセキュリティを考慮した設計になっていることが重要です。

データの安全な環境の構築に役立ついくつかの機能があります。

### システム権限およびロール

データベースにアクセスするユーザを制御します。

### 任意アクセス制御機能

データベースへの接続中にユーザが実行するアクションを制御します。

### 監査

データベースで行われたアクションの記録を管理するのに役立ちます。

### データベースサーバオプション

データベースのロードなどの管理作業を実行するユーザを指定します。このオプションは、データベースサーバの起動時に設定されます。

### ビューとストアードプロシージャ

ユーザがアクセスするデータとユーザが実行する操作を指定します。

### データベース、テーブル、および任意のデータ暗号化

データベースを安全に管理するために、単純難読化または強力な暗号化のいずれかを選択できます。単純難読化は、巧妙で容赦ないデータへのアクセス試行に対してセキュアではありません。強力な暗号化にすると、暗号化キーなしではデータベースにまったくアクセスできなくなります。

テーブル暗号化機能では、データベース全体ではなく個々のテーブルを暗号化できます。

任意のデータ暗号化では、公開鍵または秘密鍵によってメッセージなどのデータのブロックを暗号化でき、それに対応する秘密鍵または公開鍵で復号化することができます。また、秘密鍵でデータに署名して、対応する公開鍵で検証することもできます。

### トランスポートレイヤセキュリティ (TLS)

トランスポートレイヤセキュリティを使用すると、クライアントアプリケーションとデータベースサーバ間の通信を認証および暗号化することができます。トランスポートレイヤセキュリティでは、RSA 暗号方式を使用します。

データベースサーバが稼働しているコンピュータがほかのプロセスも実行している場合、TLS 暗号化を使用することで、クライアント/サーバ間の通信内容にアクセスできます。

### セキュリティ機能

データベースサーバ上のすべてのデータベースで機能を無効にすることができます。

### SELinux のサポート

SELinux のポリシーを使用して、アプリケーションのシステムリソースへのアクセスを制御します。Red Hat Enterprise Linux 5 でアクセスを保護するポリシーが含まれています。

SQL Anywhere SELinux ポリシーのコンパイルおよびインストールについては、[\\$SQLANY17/selinux/readme](#) を参照してください。

このセクションの内容:

[セキュリティ: ビューとプロシージャを使用して、ユーザがアクセスできるデータを制限 \[1596 ページ\]](#)

高レベルのセキュリティが必要なデータベースでは、テーブルに対して直接アクセスを許可することには限界がありません。

[セキュリティに関する一般的なヒント \[1598 ページ\]](#)

データのセキュリティを強化するために、さまざまなアクションを実行できます。

[セキュリティ: ユーザ ID \[1601 ページ\]](#)

データベース管理者は、ユーザ ID とパスワードを割り当てることによって、どのユーザがデータベースにアクセスするかを制御します。

[セキュリティ: パスワード \[1602 ページ\]](#)

パスワードは、データベースのセキュリティシステムの重要な部分です。

[ディスクサンドボックス \[1604 ページ\]](#)

ディスクサンドボックス機能を有効にすると、データベースファイルの読み込み/書き込み操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されます。

[機能の保護 \[1605 ページ\]](#)

データベースサーバ上で動作しているデータベースに、機能がアクセスできないようにすることができます。これらの機能は、保護され使用できなくなります。

[データベースアクティビティの監査 \[1609 ページ\]](#)

監査によって、データベースで実行されたすべてのアクティビティが追跡されます。

[データベースの暗号化と復号化 \[1614 ページ\]](#)

データベース内のデータを第三者が解読しにくくします。

[公開鍵と秘密鍵を使用したメッセージの暗号化 \[1637 ページ\]](#)

RSA キーペアを使用して、メッセージの暗号化、署名、および検証を行います。

## 関連情報

[管理ツールのセキュリティ \(Interactive SQL\) \[1041 ページ\]](#)

[ユーザ \[1530 ページ\]](#)

[セキュリティ: コマンドラインからの権限の制御 \[1552 ページ\]](#)

[テーブル暗号化 \[1621 ページ\]](#)

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

## 1.9.2.1 セキュリティ: ビューとプロシージャを使用して、ユーザがアクセスできるデータを制限

高レベルのセキュリティが必要なデータベースでは、テーブルに対して直接アクセスを許可することには限界があります。

ユーザに与えたテーブルの権限は、テーブル全体に対して適用されます。ところが、テーブルごとでなくユーザの権限をより厳密に定義する必要がある場合が、数多くあります。次に例を示します。

- employee テーブルにアクセスする必要があるユーザに対して、テーブル内にある個人的な情報にまでアクセスを許可することは望ましくない。
- 営業担当者にセールスコールの詳細を含むテーブルの更新を許可したいが、担当者自身の部分に対するアクセスだけに制限したい。

ビューがデータへのアクセスを制限するのに対し、プロシージャはユーザの行動を制限します。ユーザは、プロシージャの対象になるテーブルに対する権限がなくても、プロシージャの EXECUTE 権限を持つことができます。

セキュリティを完全にするには、基本となるテーブルへのアクセスをすべて禁止し、ユーザまたはユーザのグループには、特定のストアプロシージャを実行する権限だけを付与します。この方法では、データベースのデータの修正方法を厳密に定義できます。

### 例

Sales Manager は、自分の部署の営業部員に関するデータベースの情報にアクセスする必要があります。しかし、他部署の従業員の情報にアクセスしなければならない理由はありません。

例として、Sales Manager のユーザ ID を作成してから必要な情報を得るためのビューを作成し、Sales Manager のユーザ ID に適切な権限を与える手順を次に示します。

1. 接続して新しいユーザ ID を作成します。

```
CONNECT DBA IDENTIFIED by sql;  
CREATE USER SalesManager  
IDENTIFIED BY sales;
```

2. 営業部の従業員だけを見るビューを、次のように定義します。

```
CREATE VIEW EmployeeSales AS  
SELECT EmployeeID, GivenName, Surname  
FROM Employees  
WHERE DepartmentID = 200;
```

テーブル参照を所有者で修飾することによって、同じ名前のテーブルの参照と混同されるおそれなくなります。

3. SalesManager にビューを見る権限を与えます。

```
GRANT SELECT ON EmployeeSales TO SalesManager;
```

まったく同じ文を使用して、ビューおよびテーブルに対する権限を与えます。

次の例では、Sales Manager が注文のまとめを確認できるようにするビューを作成します。このビューは、複数のテーブルからの情報を必要とします。

1. ビューを作成します。

```
CREATE VIEW OrderSummary AS  
SELECT OrderDate, Region, SalesRepresentative, CompanyName  
FROM SalesOrders  
KEY JOIN Customers;
```

2. このビューを見る権限を Sales Manager に与えます。

```
GRANT SELECT
ON OrderSummary
TO SalesManager;
```

3. プロセスが正常に動作したことをチェックするには、ユーザ ID SalesManager に接続し、作成したビューをみます。

```
CONNECT SalesManager
IDENTIFIED BY sales;
SELECT *
FROM DBA.EmployeeSales;
SELECT *
FROM DBA.OrderSummary;
```

Sales Manager には、基本となるテーブルを見る権限は付与されていません。次の文は権限エラーを引き起こします。

```
SELECT * FROM GROUPO.Employees;
SELECT * FROM GROUPO.SalesOrders;
```

このセクションの内容:

[セキュリティ: プロシージャとトリガ \[1597 ページ\]](#)

プロシージャとトリガは、テーブルのデータへのユーザのアクセスを制限し、ユーザが直接検査や修正をできないようにすることによって、セキュリティを提供しています。

## 1.9.2.1.1 セキュリティ: プロシージャとトリガ

プロシージャとトリガは、テーブルのデータへのユーザのアクセスを制限し、ユーザが直接検査や修正をできないようにすることによって、セキュリティを提供しています。

たとえば、トリガは関連するテーブルの所有者の権限により実行されますが、テーブルのローを挿入、更新、または削除する権限を持つユーザであれば、トリガを起動できます。同様に、プロシージャ (ユーザ定義関数を含む) はプロシージャの所有者の権限により実行されますが、権限を与えられたユーザはプロシージャを呼び出すことができます。プロシージャとトリガの権限は、それらを起動するユーザが持つ権限とは異なっている場合がよくあります。

### プロシージャ

高度なセキュリティのシステムでは、すべてのベーステーブルへのアクセスを禁止し、代わりに、データとタスクへのプロシージャ経由のアクセスを許可することを検討してください。プロシージャを使用したアクセスを制限するには、次の手順に従います。

1. 実行する、許可されたタスクの各セットに対してロールを作成し、該当するシステム権限をロールに付与します。
2. 各ロールに1つの共通ロールを付与します。
3. 該当するロールに承認済みタスクを実行するためのプロシージャに対する EXECUTE 権限を付与します。
4. 新しいユーザの作成時には、そのユーザが実行するタスクに必要なロールのみを付与します。

## 関連情報

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

[アップグレードユーティリティ \(dbupgrad\) \[1197 ページ\]](#)

### 1.9.2.2 セキュリティに関する一般的なヒント

データのセキュリティを強化するために、さまざまなアクションを実行できます。

#### パスワードの慎重な選択

データベースには、短いパスワード、または容易に推測できるパスワードを使用しないでください。

#### スーパーユーザの使用の制限

すべての権限、ロール、および管理権限を持つロールの作成と付与は避けてください (つまり、スーパーユーザは避けてください)。代わりに、権限と権利の論理的なグループを持つロールを作成し、そのロールを慎重に付与してください。スーパーユーザを作成する場合は、どうしても必要な場合にのみ使用し、そのパスワードを、必要なときに取り出せるように金庫などの安全な場所に保管してください。

データベース管理者に 2 つの ID を与えることを検討してください。つまり、すべての権限を持つ ID と、権限が限られている ID です。そして、すべての権限を持つ ID は、必要なときにのみ使用するようにします。

#### 保護されたデータベース機能の使用

-sf データベースサーバオプションでは、データベースサーバ上のすべてのデータベースに対して機能を有効または無効にすることができます。無効にできる機能には、外部ストアプロシージャや Java の使用、リモートデータアクセス、要求ログ設定の変更機能などがあります。

#### データベース機能の無効化

-sf データベースサーバオプションを使用すると、データベースサーバ上で実行されているデータベースで無効にする機能のリストを指定できます。このリストに含まれる機能は、クライアントアプリケーションだけでなく、データベース内に定義されているストアプロシージャ、トリガ、イベントでも使用できません。この設定が役に立つのは、使用するデータベースが自分の所有しているものではないため、データベースのクライアントおよびサーバ間のファイル転送など、不必要なオペレーティングシステムの対話を行う可能性がある場合です。

#### 外部システム関数の削除

次の外部関数は、セキュリティ上のリスクを生じさせる危険性があります。

- xp\_cmdshell システムプロシージャ
- xp\_startmail システムプロシージャ
- xp\_startsmtp システムプロシージャ
- xp\_sendmail システムプロシージャ
- xp\_stopmail システムプロシージャ
- xp\_stopsmtmp システムプロシージャ

電子メールシステムプロシージャを使用すると、ユーザはサーバに自分が作成した電子メールを送信させることができます。悪意のあるユーザが、電子メールコマンドやコマンドシェルプロシージャを使用して、付与されていないパーミッションでオペレーティングシステムのタスクを実行することもあります。セキュリティを考慮した環境では、このような関数は削除してください。

#### データベースファイルの保護

不正アクセスからデータベースファイル、ログファイル、DB 領域ファイルを保護してください。これらのファイルは、共有ディレクトリや共有ボリュームには保管しないでください。

#### データベースソフトウェアの保護

不正アクセスから SQL Anywhere ソフトウェアを保護してください。ユーザには、アプリケーション、DLL、その他の必要なリソースへのアクセス権だけを付与してください。

#### サービスまたはデーモンとしてのデータベースサーバの実行

権限のないユーザがサーバを停止したり、データベースやログファイルへのアクセスを取得したりしないように、データベースサーバを Windows サービスとして実行してください。UNIX では、同様の目的でサーバをデーモンとして実行します。

#### SATMP 環境変数をユニークなディレクトリに設定 (UNIX)

UNIX プラットフォームでデータベースサーバを保護するには、SATMP をユニークなディレクトリに設定し、他のすべてのユーザに対して、このディレクトリの読み取り、書き込み、実行を制限します。これによって、すべての接続が強制的に TCP/IP を使用することになり、共有メモリ接続よりも安全になります。

クライアントとサーバ間で使用される共有メモリのバッファは、実際のデータが 2 つのサイト間で送受信される前に、ディレクトリツリーから削除されます。他のプロセスは、共有メモリのバッファやファイルが非表示であるために通信データを参照できません。

#### データベースを強力に暗号化

データベースを強力に暗号化すると、キーがなければまったくアクセスできなくなります。他の方法で、データベースを開くことも、データベースやトランザクションログファイルを表示することもできません。

#### -xs サーバオプション HTTP と MaxRequestSize プロトコルオプションを調整

-xs HTTP サーバオプションを使用する場合、MaxRequestSize オプションを設定して、サイズが非常に大きなファイルがデータベースサーバに送信されないようにします。サイズの大きなファイルはサーバリソースと帯域幅を占有する可能性があります。

#### HTTP タイムアウトオプションの設定

-xs HTTP サーバオプションを使用する場合、Timeout (TO) オプションと KeepaliveTimeout (KTO) オプションを設定します。Timeout は、要求を受信するまで HTTP 接続がアクティブである時間を示し、KeepaliveTimeout は、要求から要求の間でキープアライブ接続がオープンである時間を示します。

#### 1 つの要求が無制限にサーバリソースを消費することがないように request\_timeout を設定

request\_timeout データベースオプションを設定して、1 つの要求が無制限にサーバリソースを消費することがないようにします。

#### ディスクサンドボックス機能をセキュリティで保護

ディスクサンドボックス機能を有効にしている場合は、データベースファイル操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されます。

#### データベースの開始と停止

パーソナルデータベースサーバ (dbeng17) を使用している場合、デフォルトでは、すべてのユーザが実行中のサーバ上で追加のデータベースを起動できます。

デフォルトでは、ネットワークデータベースサーバ (dbsrv17) が、稼働中のデータベースサーバでデータベースを起動するには、SERVER OPERATOR システム権限が必要です。-gd データベースオプションは、この機能を実行できるユーザを、すでに接続しているデータベースで特定のレベルの権限を付与されているユーザに制限できます。

#### データベースの作成と削除

パーソナルデータベースサーバ (dbeng17) を実行している場合、デフォルトでは、すべてのユーザが CREATE DATABASE 文または DROP DATABASE 文を使用してデータベースファイルを作成できます。

デフォルトでは、ネットワークデータベースサーバ (dbsrv17) がデータベースを作成または削除するには、SERVER OPERATOR システム権限が必要です。-gu オプションは、この機能を実行できるユーザを、すでに接続しているデータベースで特定のレベルの権限を付与されているユーザに制限できます。

#### サーバの停止

dbstop ユーティリティは、データベースサーバを停止します。このユーティリティは、バッチファイルや、サーバを対話形式で (データベースサーバメッセージウィンドウの [\[シャットダウン\]](#) をクリックして) 停止できない場合に便利です。

パーソナルデータベースサーバ (dbeng17) の場合、デフォルトでは、どのユーザも、dbstop を実行してサーバを停止できます。

ネットワークデータベースサーバ (dbsrv17) の場合、デフォルトでは、データベースサーバを停止するために SERVER OPERATOR システム権限が必要です。-gk オプションは、この機能を実行できるユーザを、データベースで特定のレベルの権限を付与されているユーザに制限できます。

#### データのロードとアンロード

LOAD TABLE 文、UNLOAD TABLE 文、UNLOAD 文はすべて、データベースサーバコンピュータ上のファイルシステムにアクセスできます。-gl データベースサーバオプションを使用して、データのロードとアンロードを行うために必要なデータベース権限を制御できます。

デフォルトの設定は、ネットワークデータベースサーバの DBA です。ネットワークデータベースサーバを稼働中の場合は、ファイルシステムへの不当なアクセスによってセキュリティ問題の起こる可能性があります。

UNIX 以外のオペレーティングシステムのパーソナルデータベースサーバ (dbeng17) のデフォルト設定は *all* です。UNIX オペレーティングシステムのデフォルト設定は DBA です。パーソナルデータベースサーバが稼働中の場合は、すでにファイルシステムにアクセスできるため、セキュリティ上の問題はありません。

#### トランスポートレイヤセキュリティ (TLS) によるクライアント/サーバ通信の暗号化

トランスポートレイヤセキュリティを使用して、クライアントアプリケーションとデータベースサーバ間の通信を認証および暗号化することで、ネットワークパケットのセキュリティを高めることができます。トランスポートレイヤセキュリティでは、RSA 暗号方式を使用します。

データベース独立性をオンにします。-edi データベースサーバオプションを使用して、データベース独立性を有効にします。データベース独立性が有効になると、データベースサーバで動作中の各データベースは、唯一のデータベースであるかのように動作します。

## 関連情報

[セキュリティ: パスワード \[1602 ページ\]](#)

[サービスまたはデーモンとしてデータベースサーバを実行する方法 \[339 ページ\]](#)

[ディスクサンドボックス \[1604 ページ\]](#)

[スーパーユーザ \[1484 ページ\]](#)

[セキュリティ機能キーの作成 \[1607 ページ\]](#)

[SATMP 環境変数 \[553 ページ\]](#)

[-ep データベースサーバオプション \[406 ページ\]](#)

[-ek データベースオプション \[527 ページ\]](#)

- xs データベースサーバオプション [507 ページ]
- request\_timeout オプション [772 ページ]
- sbx データベースサーバオプション [470 ページ]
- sbx データベースオプション [534 ページ]
- sf データベースサーバオプション [471 ページ]

### 1.9.2.3 セキュリティ: ユーザ ID

データベース管理者は、ユーザ ID とパスワードを割り当てることによって、どのユーザがデータベースにアクセスするかを制御します。

各ユーザ ID に権限を付与することによって、データベース接続しているときに各ユーザが実行できるタスクを制御します。

#### ユーザ ID に基づく権限のスキーム

ユーザは、データベースにログインすると、次の基準のいずれかを満たすすべてのデータベースオブジェクトにアクセスできます。

- ユーザが作成したオブジェクト
- ユーザに対して明示的な権限が付与されているオブジェクト
- ユーザの所属グループに明示的な権限が付与されているオブジェクト

ユーザは、この基準を満たさないデータベースオブジェクトにはアクセスできません。つまり、ユーザは、自分が所有するオブジェクト、またはアクセス権限を明示的に付与されているオブジェクトにのみアクセスできます。

#### ロールと権限

ロールと権限を付与することで、ユーザがデータベースオブジェクトに対して実行できるタスク (作成、変更、実行、更新など) と、ユーザが実行できる管理タスク (バックアップ、プロファイリングなど) を制御できます。

ロールと権限を付与するには、GRANT 文および GRANT ROLE 文を使用します。

REVOKE および REVOKE ROLE 文は、付与と反対の機能を持っています。GRANT によって明示的に付与されたロールや権限はいずれも、REVOKE で取り消すことができます。ユーザから CONNECT を取り消すと、そのユーザは、所有するすべてのオブジェクトとともにデータベースから削除されます。

#### Guest ユーザ ID

パスワード付きのユーザ ID「Guest」を作成すると、統合ログインまたは Kerberos ログインを使用してデータベースへの認証を実行できるユーザであれば、誰でもデータベースにログインアクセスできます。GRANT 文を使用したログインマッピングは

不要です。Guest ユーザは割り当てられたロールまたは権限の制約を受けませんが、Guest ユーザ ID の作成は推奨されません。

## 関連情報

[セキュリティ: 統合化ログインを使用すると、無制限データベースアクセスが行われる可能性があります \[163 ページ\]](#)

[Windows 統合化ログイン \[124 ページ\]](#)

[Kerberos ユーザ認証 \[149 ページ\]](#)

[login\\_mode オプション \[717 ページ\]](#)

### 1.9.2.4 セキュリティ: パスワード

パスワードは、データベースのセキュリティシステムの重要な部分です。

安全のために、パスワードは容易に推測できないものにし、ハードドライブやその他のロケーションから簡単にアクセスできないようにしてください。SQL Anywhere パスワードは、大文字と小文字が区別されます。

接続中にクライアントとデータベースサーバ間で送信されるパスワードは暗号化されます。

#### ログインポリシーの実装

ログインポリシーを使用して、パスワードの変更頻度の設定や、アカウントがロックされるまでに実行可能なログイン試行回数の指定を行います。

#### 最小長のパスワードの実装

デフォルトでは、パスワードが 6 バイトの長さである必要があります。セキュリティを強化するために、すべての新しいパスワードに必要な最小長を変更して、短い (したがって、推測しやすい) パスワードを禁止します。これを実行するには、`min_password_length` データベースオプションを 0 より大きな値に設定します。次の文では、パスワードが 8 バイト以上の長さになるようにします。

```
SET OPTION PUBLIC.min_password_length = 8;
```

#### パスワード有効期限の導入

デフォルトでは、データベースパスワードに有効期限はありません。ログインポリシーを使用して、パスワードの有効期限を導入します。

## プロシージャにパスワードを含めない

CREATE USER などの文には、パスワードを指定するために IDENTIFIED BY 句があります。プロシージャの定義は SYSPROCEDURE システムビューに表示されるため、これらの文を含むプロシージャを作成するときには、文字列リテラルとしてパスワードを指定しないでください。代わりに、プロシージャ本文の外部に変数を作成して、IDENTIFIED BY 句から変数を参照します。

## ODBC データソースにパスワードを含めない

セキュリティを考慮した環境では、権限のないユーザが簡単にパスワードを使用できないようにすることが重要です。

ODBC データソースなどにユーザが接続できるプロファイルを作成する場合、パスワードを含めないでください。この方法により、権限のないユーザによってパスワードを参照されないようにします。

## パスワードを含む設定ファイルの暗号化

設定ファイルを作成するときには、パスワードを含めないでください。パスワード情報を除外できないと判断した場合は、ファイルの内容をファイル非表示ユーティリティ (dbfhide) でエンコードして、適切なオペレーティングシステムのパーミッションを使用した不正アクセスから保護することを検討します。

## パスワード検証の使用

verify\_password\_function オプションを使用して、パスワード規則を実装する関数を指定します。

## 管理者 1 人では別のユーザの完全なパスワードを知ることができないようにする

二重制御パスワード機能を使用して、パスワードを変更するために 2 人の管理者が必要となるようにします。1 人の管理者がパスワードの最初の部分を設定し、もう 1 人の管理者がパスワードの 2 番目の部分を設定します。ユーザは、データベースに接続するためのパスワードを 2 つの部分に分けて指定します。ターゲットユーザは、そのログインポリシーで change\_password\_dual\_control オプションが有効になっている必要があります。

## 関連情報

[ログインポリシー \[613 ページ\]](#)

[ODBC データソース \[111 ページ\]](#)

[二重制御パスワードの設定 \(SQL の場合\) \[1558 ページ\]](#)

[verify\\_password\\_function オプション \[826 ページ\]](#)  
[min\\_password\\_length オプション \[736 ページ\]](#)  
[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

## 1.9.2.5 ディスクサンドボックス

ディスクサンドボックス機能を有効にすると、データベースファイルの読み込み/書き込み操作が、メインデータベースファイルがあるディレクトリとそのサブディレクトリに制限されます。

この制限により、ユーザによるアクセスが禁止されているファイルへのアクセスが防止されると同時に、ユーザがファイル操作を実行できます。すべてのデータベースユーザのサンドボックスの場所は同じです。

ディスクサンドボックス機能が有効な場合、相対パス名は、メインデータベースファイル (システム DB 領域) があるディレクトリに対する相対パスとみなされます。ディスクサンドボックス機能が有効でない場合、相対パス名は、データベースサーバの現在の作業ディレクトリに対する相対パスとみなされます。

ディスクサンドボックス機能の最も安全な設定は、常に優先されます。サンドボックス機能を有効にしてデータベースを起動した (-sbox サーバオプションの 1 つまたは START DATABASE 文で指定したか、disk\_sandbox データベースオプションを On に設定した) 場合、ディスクサンドボックス機能を無効にする他の設定にかかわらず、データベースはディスクサンドボックス内で稼働します。ディスクサンドボックス機能は、次の方法で、有効または無効にすることができます。

設定	スコープ	持続性	注記
-sbox データベースサーバオプション	データベースサーバで実行されているすべてのデータベース	現在のセッションのみ	<ul style="list-style-type: none"> <li>セキュリティ機能の設定による制御</li> <li>データベースサーバの起動時に指定</li> </ul>
sa_server_option システムプロシージャの DiskSandbox オプション	データベースサーバで実行されているすべてのデータベース	現在のセッションのみ	<ul style="list-style-type: none"> <li>セキュリティ機能の設定による制御</li> <li>データベースサーバの起動後に指定</li> </ul>
-sbox データベースオプション	データベースサーバで実行されている個々のデータベース	現在のセッションのみ	<ul style="list-style-type: none"> <li>セキュリティ機能の設定による制御</li> <li>データベースサーバの起動時に指定</li> </ul>
sa_db_option システムプロシージャ	データベースサーバで実行されている個々のデータベース	現在のセッションのみ	<ul style="list-style-type: none"> <li>セキュリティ機能の設定による制御</li> <li>データベースサーバの起動後に指定</li> </ul>
START DATABASE 文の DISKSANDBOX 句	データベースサーバで実行されている個々のデータベース	現在のセッションのみ	<ul style="list-style-type: none"> <li>セキュリティ機能の設定による制御</li> <li>データベースサーバの起動後に指定</li> </ul>
disk_sandbox database option	現在のデータベース	セッションを通して持続する設定	<ul style="list-style-type: none"> <li>特定のシステム権限が必要</li> <li>データベースに対して指定</li> </ul>

## ディスクサンドボックス機能を制御するセキュリティ機能

データベースサーバを起動すると、manage\_disk\_sandbox および disk\_sandbox セキュリティ機能がデフォルトで有効になります。この設定によって、ユーザがディスクサンドボックス機能を変更できるかどうかを制御されます。個々の接続の manage\_disk\_sandbox セキュリティ機能を無効にする必要がある場合は、sp\_use\_secure\_feature\_key システムプロシージャを使用する必要があります。

## データベースのバックアップとディスクサンドボックス機能

データベースがディスクサンドボックス内で稼働している場合は、クライアント側のバックアップを無制限に作成できます。

サーバ側のバックアップを作成して、ディスクサンドボックスディレクトリの外にバックアップを格納するには、バックアップを実行する接続のデータベースサーバセキュリティ機能キーを指定します。

## 関連情報

[-sbx データベースサーバオプション \[470 ページ\]](#)

[-sbx データベースオプション \[534 ページ\]](#)

[disk\\_sandbox オプション \[696 ページ\]](#)

[-sf データベースサーバオプション \[471 ページ\]](#)

## 1.9.2.6 機能の保護

データベースサーバ上で動作しているデータベースに、機能がアクセスできないようにすることができます。これらの機能は、保護され使用できなくなります。

機能が保護されている (アクセス不可にされている) 場合は、クライアントアプリケーション、データベース定義ストアプロシージャ、トリガ、イベントで機能を使用できません。セキュリティ機能の設定は、データベースサーバで稼働中のすべてのデータベースに適用されます。セキュリティ機能は、ホストコンピュータシステムの外部環境 (例: ホストコンピュータのディレクトリやファイル) を参照する組み込み論理が含まれる可能性のあるデータベースを起動する必要がある場合に便利です。この機能は、データベースサーバファームを実行してさまざまな顧客のデータベースをホストしているサードパーティベンダに有益です。

-sf データベースサーバオプションを使用すると、データベースサーバで稼働中のデータベースに対して保護する機能を指定できます。

## デフォルトで保護されている機能

-sf データベースサーバオプションを使用してサーバを起動すると、次の機能セットはデフォルトで保護されます。

- `MANAGE_SECURITY`
- `SERVER_SECURITY` (`TRACE_SYSTEM_EVENT` を除く)

## セキュリティ機能キー

**SYSTEM** セキュリティ機能キーは、データベースサーバの起動時に `-sk` データベースサーバオプションと認証コードを指定して作成します。 `sp_use_secure_feature_key` システムプロシージャにこの認証コードを指定して、**SYSTEM** セキュリティ機能キーへのアクセスを可能にします。次に、`sa_server_option` システムプロシージャを使用して、データベースサーバの稼働中に機能が保護されるかどうかを変更します。

`-sk` オプションを指定しないでデータベースを起動した場合は、データベースサーバやそのサーバで実行中のすべてのデータベースについてセキュリティ機能の設定を変更できなくなります。後になって、`sp_create_secure_feature_key` システムプロシージャを使用して **SYSTEM** セキュリティ機能を作成することはできません。代わりに、データベースサーバをシャットダウンし、再起動時に `-sk` オプションを指定する必要があります。

**SYSTEM** セキュリティ機能キーにアクセスすると、`sp_create_secure_feature_key` システムプロシージャを使用して、カスタマイズセキュリティ機能キーを作成することができます。カスタマイズセキュリティ機能キーには、使用できないように保護されていてほかのユーザが特定の機能セットにアクセスできるようにするために使用する認証コードが付いています。

データベースサーバごとのセキュリティ機能キーは 1000 個に制限されています。

認証コードは、6 文字以上の空でない文字列にする必要があり、二重引用符、制御文字 (0x20 未満のすべての文字)、またはバックスラッシュを含めることはできません。

次のシステムプロシージャを使用して、**SYSTEM** セキュリティ機能キーおよびカスタマイズセキュリティ機能キーにアクセスします。

- `sp_use_secure_feature_key` システムプロシージャ

このシステムプロシージャを使用するには、`EXECUTE` 権限が必要です。

アクセスはセッションベースです。データベースサーバから切断されると、アクセスが失われます。サーバに再接続する場合は、アクセスを再取得する必要があります。

次のシステムプロシージャを使用して、機能の保護および保護の解除を行います。

- `'SecureFeatures'` オプションを指定した `sa_server_option` システムプロシージャ

データベースサーバの稼働中に機能が保護または保護解除されます。サーバがシャットダウンされると、設定は失われます。

カスタマイズセキュリティ機能キーは、次のシステムプロシージャを使用して管理します。

- `sp_create_secure_feature_key` システムプロシージャ
- `sp_alter_secure_feature_key` システムプロシージャ
- `sp_drop_secure_feature_key` システムプロシージャ
- `sp_list_secure_feature_key` システムプロシージャ

これらのシステムプロシージャを使用するには、`MANAGE_KEYS` 機能を有効化 (取得) する必要があります。

セキュリティ機能キーは、データベースサーバの稼働中に存在するものです。サーバがシャットダウンされると、キーは失われます。サーバを再起動するときに、キーを再作成する必要があります。

## **i** 注記

SYSTEM セキュリティ機能キーは、MANAGE\_FEATURES と MANAGE\_KEYS の両方の機能が有効なカスタマイズセキュリティ機能キーが作成されている場合にのみ削除できます。

このセクションの内容:

### [セキュリティ機能キーの作成 \[1607 ページ\]](#)

ユーザが使用できるデータベース機能を制御するには、セキュリティ機能データベースサーバオプション (-sf) を使用して、データベースサーバでユーザがアクセスできない機能を指定します。

## 関連情報

[-sf データベースサーバオプション \[471 ページ\]](#)

[-sk データベースサーバオプション \[479 ページ\]](#)

## 1.9.2.6.1 セキュリティ機能キーの作成

ユーザが使用できるデータベース機能を制御するには、セキュリティ機能データベースサーバオプション (-sf) を使用して、データベースサーバでユーザがアクセスできない機能を指定します。

## 前提条件

SERVER OPERATOR システム権限が必要です。また、MANAGE\_KEYS 機能にアクセスできることが必要です。

## コンテキスト

セキュリティ機能の設定は、データベースサーバで稼働中のすべてのデータベースに適用されます。

セキュリティ機能オプション (-sf) で、次のような機能の使用を制御します。

- サーバ側のバックアップ
- 外部ストアプロシージャ
- リモートデータアクセス
- Web サービス

-sk オプションで、データベースサーバのセキュリティ機能へのアクセスを管理する SYSTEM セキュリティ機能キーを指定します。データベースサーバが稼働してからセキュリティ機能の一覧を変更するには、sa\_server\_option システムプロシージャを使用します。データベースサーバが稼働してからカスタマイズセキュリティ機能キーを変更するには、sp\_alter\_secure\_feature\_key システムプロシージャを使用します。

sp\_create\_secure\_feature\_key システムプロシージャによって、カスタマイズセキュリティ機能キーが作成されます。

## 手順

1. コマンドプロンプトで、-sf および -sk オプションを使用してデータベースサーバを起動します。

たとえば、次のコマンドで、データベースサーバを起動して、すべての機能を保護します。このコマンドには、接続のセキュリティ機能へのアクセスを許可するために後で使用できるキーも含まれます。

```
dbsrv17 -n secure_server -sf all -sk secretAuthCode mydemo.db
```

2. データベースサーバに接続します。

```
dbisql -c "UID=DBA;PWD=passwd;Host=myhost;Server=secure_server;DBN=mydemo"
```

3. sp\_use\_secure\_feature\_key システムプロシージャを呼び出して、接続のための SYSTEM セキュリティ機能キーを指定します。使用する認証コードは、-sk オプションで指定されます。

```
CALL sp_use_secure_feature_key ( 'system' , 'secretAuthCode' );
```

4. sa\_server\_option システムプロシージャを使用してサーバのセキュリティ機能のセットを変更します。

次に例を示します。

```
CALL sa_server_option( 'all', '-remote_data_access' );
```

5. 特定のユーザのカスタマイズセキュリティ機能キーを作成します。

たとえば、Bob のカスタマイズセキュリティ機能キーを作成して、電子メールの送信を許可します。

```
CALL sp_create_secure_feature_key ( 'bobsKey' , 'anotherAuthKey' ,  
'sa_send_email' );
```

データベースにログインした後、Bob が電子メールを送信するには、次のコマンドを実行する必要があります。

```
CALL sp_use_secure_feature_key ( 'bobsKey' , 'anotherAuthKey' );
```

## 結果

現在、データベースサーバには、SYSTEM セキュリティ機能だけでなく、特定のユーザに割り当てられたカスタマイズセキュリティ機能もあります。

データベースサーバ secure\_server で稼働中のデータベースのユーザは、remote\_data\_access 機能を除き、すべてのセキュリティ機能にアクセスできません。ただし、ユーザ Bob は、sa\_send\_email にもアクセスできます。

## 関連情報

[-sf データベースサーバオプション \[471 ページ\]](#)

## 1.9.2.7 データベースアクティビティの監査

監査によって、データベースで実行されたすべてのアクティビティが追跡されます。

各データベースには関連するログファイルがあります。このログファイルはデータベースのリカバリに使用され、データベースに対して実行したトランザクションの記録が含まれています。

トランザクションログには、実行されたすべてのデータ定義文と、それを実行したユーザ ID が格納されます。また、すべての更新、削除、挿入、これらの文を実行したユーザも格納されます。ただし、監査の目的によっては、この情報では不十分です。デフォルトでは、トランザクションログにはイベントの時間が含まれません。イベントが発生した順序のみが含まれます。失敗したイベントや、SELECT 文も含まれません。

監査を使用するときには、追加のデータは監査ログに保存されます。このログはトランザクションログ、ユーザ定義 ETD ログ、またはシステムイベントログにあります。このデータは次を含む可能性があります。

- コンピュータ名を含む、すべてのログイン試行 (成否とも)。
- すべてのイベントの正確なタイムスタンプ (ミリ秒まで解析)。
- すべてのパーミッションの検査 (成否とも)。パーミッションが検査されたオブジェクトがあれば、それも含まれます。
- システム権限を必要とするすべてのアクション。
- すべての xp\_cmdshell システムプロシージャの実行。

データベースの監査が有効で、監査ログがトランザクションログの場合、トランザクションログを使用して停止することはできません。その場合、トランザクションログを停止するには、最初に監査を停止するか audit\_log オプションを変更して FILE または SYSLOG を指定します。

### 個々の接続の監査

データベースの監査を有効にした後で、データベースログインプロシージャに conn\_auditing database テンポラリオプションを指定し、接続ごとに監査を有効にすることができます。クライアントコンピュータの IP アドレスや接続のタイプなどの情報に基づいて監査を有効にできます。

ログインプロシージャに conn\_auditing オプションを設定しなかった場合、このオプションはデフォルトで On に設定されません。

次の例は、DBA というユーザによる接続を除くすべての接続で監査を有効にするログインプロシージャの一部を示しています。

```
DECLARE usr VARCHAR(128)
SELECT CONNECTION_PROPERTY( 'Userid' ) INTO usr;
IF usr != 'DBA' THEN
    SET TEMPORARY OPTION conn_auditing='On'
ELSE
    SET TEMPORARY OPTION conn_auditing='Off'
END IF;
```

## 監査コメント

sa\_audit\_string システムストアードプロシージャを使用して、監査証跡にコメントを追加します。これは引数を 1 つとります。引数は 200 バイト以内の文字列です。次に例を示します。

```
CALL sa_audit_string( 'Started audit testing here.' );
```

このコメントは監査文として監査ログに格納されます。

このセクションの内容:

### [監査の設定 \(SQL の場合\) \[1610 ページ\]](#)

トランザクションログにおけるセキュリティ関連情報の監査を設定します。監査はデフォルトでは無効になっています。

### [監査情報の取得 \(dbtran ユーティリティの場合\) \[1611 ページ\]](#)

トランザクションログから監査情報を取り出します。

### [チュートリアル: 監査 \[1613 ページ\]](#)

権限のない情報にアクセスを試み、その後監査機能によってどのように試みが記録されるかを表示します。

## 関連情報

[トランザクションログ \[279 ページ\]](#)

### 1.9.2.7.1 監査の設定 (SQL の場合)

トランザクションログにおけるセキュリティ関連情報の監査を設定します。監査はデフォルトでは無効になっています。

## 前提条件

SET ANY SECURITY OPTION システム権限が必要です。

## 手順

1. データベースに接続します。
2. 監査を有効にするには次の文を実行します。

```
SET OPTION PUBLIC.auditing = 'On';
```

3. 次の文を実行して、トランザクションログへ強制的にログを記録します。

```
SET OPTION PUBLIC.audit_log = 'TRANSLOG'
```

4. 有効にする監査情報のタイプを指定するには、次のシステムプロシージャを実行します。

```
CALL sa_enable_auditing_type( 'all' );
```

*all* を、有効にする監査のタイプに置き換えることによって、収集する監査情報のタイプを制御できます。

5. 監査を無効にするには次の文を実行します。

```
SET OPTION PUBLIC.auditing = 'Off';
```

無効にする監査情報のタイプを指定するには、次のシステムプロシージャを使用します。

```
CALL sa_disable_auditing_type( 'all' );
```

*all* を、無効にする監査のタイプに置き換えることによって、特定タイプの監査情報を収集の対象から除外できます。

## 結果

監査の動作が設定されています。

## 関連情報

[login\\_procedure オプション \[720 ページ\]](#)

[conn\\_auditing オプション \[673 ページ\]](#)

[auditing オプション \[656 ページ\]](#)

[auditing オプション \[656 ページ\]](#)

[auditing\\_options オプション \(システムで使用するために予約済み\) \[658 ページ\]](#)

### 1.9.2.7.2 監査情報の取得 (dbtran ユーティリティの場合)

トランザクションログから監査情報を取り出します。

## 前提条件

実行中のデータベースサーバから監査情報を取得するには BACKUP DATABASE システム権限が必要です。データベースのトランザクションログファイルから監査情報を取得する場合は権限は必要ありません。

データベースの auditing データベースオプションが ON になっている必要があります。

## コンテキスト

dbtran ユーティリティは、トランザクションログを SQL スクリプトファイルに変換します。dbtran ユーティリティが -g オプションで実行されているときには、監査情報を (コメントとして) 追加し、SQL 文を時系列に並べます。年代順で実行中の文では、結果となるデータベースの同一性が保証されないため、この出力はデータベースには適用しないでください。

## 手順

1. データベースに接続します。
2. 稼働中のデータベースサーバに対して、またはデータベーストランザクションログファイルに対して、dbtran ユーティリティを -g オプションで実行します。

オプション	アクション
稼働中のデータベースサーバからの監査情報の取り出し	次のコマンドを実行します。 <pre>dbtran -g -c connection-string -n SQL-file</pre>
トランザクションログファイルからの監査情報の取り出し	データベースサーバを停止して、トランザクションログファイルが使用可能であることを確認します。次のコマンドを実行します。 <pre>dbtran -g transaction-log SQL-file</pre>

## 結果

監査情報が取り出され、SQL スクリプトファイルに保存されます。

### 例

次の例では、監査情報が `demo.sql` ファイルに保存され、このファイルにはサンプルデータベースに関する情報が含まれます。

```
dbtran -g -c "UID=DBA;PWD=sql;DBN=demo" -n demo.sql
```

次の例では、トランザクションログファイル `demo.log` から取り出した監査情報が、`demo.sql` に保存されます。

```
dbtran -g demo.log demo.sql
```

## 関連情報

[接続パラメータのアルファベット順リスト \[44 ページ\]](#)

### 1.9.2.7.3 チュートリアル: 監査

権限のない情報にアクセスを試み、その後監査機能によってどのように試みが記録されるかを表示します。

#### 前提条件

SET ANY SECURITY OPTION および MANAGE ANY USER システム権限が必要です。

#### 手順

1. Interactive SQL を起動し、SQL Anywhere 17 Demo データソースを使用してサンプルデータベースに接続します。
2. 次のように SET OPTION 文を指定して、監査を有効にします。

```
SET OPTION PUBLIC.auditing = 'On';
```

3. 次のように CREATE USER 文を指定して、サンプルデータベースにユーザ Test1 を追加します。

```
CREATE USER Test1  
IDENTIFIED BY welcome;
```

4. 新しい Interactive SQL ウィンドウを開き、Test1 としてサンプルデータベースに接続し、次の SELECT 文を使用して Employees テーブルの機密情報へのアクセスを試みます。

```
SELECT Surname, Salary  
FROM GROUPO.Employees;
```

Employees テーブルから選択するためのパーミッションがないことを示すエラーメッセージが表示されます。

5. 次のコマンドを実行して、このアクティビティの demo.sql にある監査情報を表示します。

```
dbtran -g -c "DSN=SQL Anywhere 17 Demo;PWD=sql" -n demo.sql
```

6. サンプルデータベースを元の状態にリストアします。
  - a. DROP USER 文を使用して、データベースから Test1 ユーザを削除します。

```
DROP USER Test1;
```

- b. 次の SET OPTION 文を使用して、監査を無効にします。

```
SET OPTION PUBLIC.auditing = 'Off';
```

## 1.9.2.8 データベースの暗号化と復号化

データベース内のデータを第三者が解読しにくくします。

データベースを安全に管理するために、単純難読化または強力な暗号化のいずれかを選択できます。

データベース管理者が、次の 4 種類の強力な暗号化のテクノロジーを管理します。

- 暗号化されるもの (データベース、DB 領域、個別のテーブル、トランザクションログ、トランザクションログミラー)
- 暗号化キー
- 暗号化キーの保護
- 暗号化アルゴリズム

データベースが難読化または暗号化されている場合、WinZip などのツールでデータベースを圧縮しても、元のデータベースファイルよりも大幅に小さくはなりません。

このセクションの内容:

### [単純難読化と強力な暗号化 \[1615 ページ\]](#)

2 つのデータベースエンコーディングの方法として、単純難読化と強力な暗号化がサポートされています。

### [データベースを暗号化する方法 \[1616 ページ\]](#)

データベースを暗号化する方法は複数あります。

### [テーブル、カラム、およびマテリアライズドビューを暗号化する方法 \[1617 ページ\]](#)

データベースの一部だけを暗号化する場合は、カラムまたはテーブルを暗号化することを選択できます。

### [セキュリティ: 暗号化されたデータベースのキー \[1626 ページ\]](#)

強力に暗号化されたデータベースやテーブルを使用した暗号化キーのセキュリティを向上させる方法を学習します。

### [暗号化されたデータベースの作成 \(SQL の場合\) \[1628 ページ\]](#)

作成中にデータベースを暗号化します。

### [暗号化されたデータベースの作成 \(dbinit ユーティリティの場合\) \[1630 ページ\]](#)

データベースの作成時にデータベースを暗号化します。

### [単純難読化によりエンコードされたデータベースの作成 \[1631 ページ\]](#)

データベースの作成時にエンコードオプションを指定します。

### [既存のデータベースの暗号化 \(CREATE ENCRYPTED DATABASE 文\) \[1632 ページ\]](#)

強力に暗号化されたデータベースのコピーとともに、すべての関連トランザクションログ、トランザクションログミラー、DB 領域のファイルの暗号化されたコピーも作成します。

### [既存のデータベースの暗号化 \(CREATE ENCRYPTED FILE 文\) \[1633 ページ\]](#)

CREATE ENCRYPTED DATABASE 文が使用できない場合に、既存のデータベースとその関連ファイルの暗号化されたコピーを作成します。たとえば、サポートの目的でデータベースの暗号化が必要な場合、CREATE ENCRYPTED DATABASE 文からエラーが返されます。

### [データベースの復号化 \(SQL の場合\) \[1635 ページ\]](#)

データベースとともに、関連するトランザクションログファイル、トランザクションログミラーファイル、DB 領域ファイルを複合化します。

### [暗号化を使用した場合のパフォーマンスの問題 \[1636 ページ\]](#)

データベースが暗号化されている場合、パフォーマンスは低下します。パフォーマンスの影響は、ディスクとのページの読み取りや書き込みの頻度によって異なります。また、サーバが使用するキャッシュサイズを適切に設定することによって影響を最小限にできます。

## 1.9.2.8.1 単純難読化と強力な暗号化

2つのデータベースエンコーディングの方法として、単純難読化と強力な暗号化がサポートされています。

### 単純難読化

単純難読化は、第三者がディスクユーティリティを使用してデータベースの内容を簡単に検査することを、不可能ではないにしろ困難にすることを目的としています。単純難読化では、データベースをエンコードするためのキー（パスワード）は不要です。

### 強力な暗号化 (AES)

強力な暗号化では、キー（パスワード）がなければデータベースを使用することができません。データベースのデータは検査に対して保護されています。アルゴリズムによって、データベースやトランザクションログファイルに含まれる情報を暗号化して解読できないようにしています。

AESは、強力な暗号化を実装するために使用されているアルゴリズムです。これは、米国商務省標準技術局 (NIST: National Institute of Standards and Technology) によってブロック暗号のための新しい次世代標準暗号化方式 (AES: Advanced Encryption Standard) として選択されたブロック暗号化アルゴリズムです。

AES キーワードを使用して、128ビットの暗号化アルゴリズムを使用することを指定できます。AES256 キーワードを使用して、256ビットの暗号化アルゴリズムを使用することを指定できます。暗号化技術は機能に組み込まれているので、別途ライセンスを持つ必要はありません。

また、AES\_FIPS (128ビット) または AES256\_FIPS (256ビット) キーワードの1つを指定することによって、強力な暗号化に別途ライセンスが必要な FIP 認定の AES モジュールの使用を指定することができます。-fips オプションを指定してデータベースサーバを起動した場合、AES、AES256、AES\_FIPS、または AES256\_FIPS のうちのいずれかの強力な暗号化アルゴリズムで暗号化されたデータベースを開始できますが、単純難読化方式で暗号化されたデータベースは開始できません。-fips が指定されているときは、暗号化されていないデータベースをサーバ上で開始することもできます。

AES\_FIPS または AES256\_FIPS で暗号化されたデータベースを開始するには、コンピュータに別途ライセンスが必要な FIPS-certified AES モジュールをインストールする必要があります。

強力な暗号化テクノロジーはすべて、輸出規制対象品目です。

### 関連情報

[-fips データベースサーバオプション \[411 ページ\]](#)

## 1.9.2.8.2 データベースを暗号化する方法

データベースを暗号化する方法は複数あります。

暗号化されたデータベースを作成するには  
次を使用できます。

- 初期化ユーティリティ (dbinit) と、強力な暗号化を有効にするための `-ep`、`-ek`、または `-ea` などのオプション。
- CREATE DATABASE 文。
- SQL Central の [データベース作成ウィザード](#) で強力的に暗号化されたデータベースを作成します。

既存のデータベースを暗号化するには

既存のデータベースでは、強力な暗号化のオンとオフを簡単に切り替えることはできませんが、次のいずれかを使用して、強力的に暗号化されたデータベースのコピーを作成することができます。

- CREATE ENCRYPTED DATABASE 文を使用することができます。CREATE ENCRYPTED DATABASE 文は、データベースのほかに、トランザクションログ、トランザクションログミラー、DB 領域ファイルのような関連ファイルも暗号化するため、おすすめの方法です。
- CREATE ENCRYPTED DATABASE 文を使用できない場合は、CREATE ENCRYPTED FILE 文を使用できます。この文を使用してデータベースを暗号化したら、トランザクションログ、トランザクションログミラー、DB 領域ファイルなどのすべての関連ファイルを暗号化する必要があります。

このセクションの内容:

### [CREATE ENCRYPTED DATABASE 文と CREATE ENCRYPTED FILE 文の比較 \[1617 ページ\]](#)

既存のデータベースを暗号化する場合は CREATE ENCRYPTED DATABASE 文を使用します。サポートの目的で暗号化するデータベースがあり、CREATE ENCRYPTED DATABASE 文を使用できない場合は、CREATE ENCRYPTED FILE 文を使用します。

## 関連情報

[テーブル、カラム、およびマテリアライズドビューを暗号化する方法 \[1617 ページ\]](#)

[CREATE ENCRYPTED DATABASE 文と CREATE ENCRYPTED FILE 文の比較 \[1617 ページ\]](#)

[公開鍵と秘密鍵を使用したメッセージの暗号化 \[1637 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

[アンロードユーティリティ \(dbunload\) \[1176 ページ\]](#)

## 1.9.2.8.2.1 CREATE ENCRYPTED DATABASE 文と CREATE ENCRYPTED FILE 文の比較

既存のデータベースを暗号化する場合は CREATE ENCRYPTED DATABASE 文を使用します。サポートの目的で暗号化するデータベースがあり、CREATE ENCRYPTED DATABASE 文を使用できない場合は、CREATE ENCRYPTED FILE 文を使用します。

この文の実行時には、暗号化しようとするデータベースには接続できません。

CREATE ENCRYPTED FILE 文と CREATE ENCRYPTED DATABASE 文には次の違いがあります。

- CREATE ENCRYPTED FILE 文はデータベース関連ファイル (トランザクションログ、トランザクションログミラー、DB 領域) ごとに実行する必要がありますが、CREATE ENCRYPTED DATABASE 文では、データベース関連ファイルがすべて自動的に暗号化されます。
- CREATE ENCRYPTED DATABASE 文はリカバリが必要なデータベースには使用できませんが、CREATE ENCRYPTED FILE 文は使用できます。
- CREATE ENCRYPTED DATABASE 文は、プロシージャ、トリガ、またはバッチ内では使用できません。CREATE ENCRYPTED FILE 文は使用できます。
- CREATE ENCRYPTED DATABASE 文では単純難読化アルゴリズムがサポートされていますが、CREATE ENCRYPTED FILE 文ではこのアルゴリズムはサポートされていません。

### 関連情報

[既存のデータベースの暗号化 \(CREATE ENCRYPTED DATABASE 文\) \[1632 ページ\]](#)

[DatabaseKey \(DBKEY\) 接続パラメータ \[67 ページ\]](#)

## 1.9.2.8.3 テーブル、カラム、およびマテリアライズドビューを暗号化する方法

データベースの一部だけを暗号化する場合は、カラムまたはテーブルを暗号化することを選択できます。

カラムの暗号化は、任意のテーブル内の任意のカラムに対していつでも実行できます。テーブルの暗号化を行うには、データベースでテーブルの暗号化が有効になっている必要があります。テーブルの暗号化は、データベースの作成 (初期化) 時に有効にします。

テーブルを暗号化するには

次を使用できます。

- 初期化ユーティリティ (dbinit)。
- CREATE DATABASE 文。
- CREATE ENCRYPTED TABLE DATABASE 文。

カラムを暗号化するには

ENCRYPT 関数。

マテリアライズドビューを暗号化するには

ALTER MATERIALIZED VIEW 文。

このセクションの内容:

#### [カラムの暗号化 \[1618 ページ\]](#)

データベースのカラムを暗号化するには、ENCRYPT 関数を使用します。

#### [テーブル暗号化 \[1621 ページ\]](#)

テーブルの暗号化では、データベース全体の暗号化がもたらすようなパフォーマンスの低下を招くことなく、機密データが含まれるテーブルやマテリアライズドビューを暗号化できます。

#### [データベース内のテーブル暗号化の有効化 \(SQL の場合\) \[1622 ページ\]](#)

テーブル暗号化が有効になっているデータベースを作成するか、または既存のデータベースでテーブル暗号化を有効化すると、あとでテーブルを暗号化することができます。

#### [データベース内のテーブル暗号化の有効化 \(dbinit ユーティリティの場合\) \[1624 ページ\]](#)

データベース作成中にテーブル暗号化を有効化します。

#### [テーブルの暗号化 \[1625 ページ\]](#)

暗号化テーブルを作成するか、既存のテーブルを暗号化します。

## 関連情報

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

### 1.9.2.8.3.1 カラムの暗号化

データベースのカラムを暗号化するには、ENCRYPT 関数を使用します。

ENCRYPT 関数は、同じ AES の強力な暗号化アルゴリズムを使用します。このアルゴリズムはデータベースの暗号化用に使用され、その関数に渡される値を暗号化します。

暗号化されたデータは、DECRYPT 関数で復号化できます。このとき、ENCRYPT 関数で指定したキーと同じキーを使用する必要があります。これらの関数はともに LONG BINARY 値を返します。異なるデータ型を使用する必要がある場合は、CAST 関数を使用して、その値を必要なデータ型に変換できます。

ENCRYPT 関数と DECRYPT 関数も、未加工の暗号化をサポートしています。データベースサーバ内のデータを、エクスポートしてサーバ外で復号化できるフォーマットに暗号化できます。

データベースユーザが復号化された形式のデータにアクセスする必要がある場合でも、暗号化キーにはアクセスできないようにする必要があります。この場合は、DECRYPT 関数を使用するビューを作成できます。これにより、ユーザは暗号化キーを知らなくても、復号化されたデータにアクセスできるようになります。テーブルを使用したビューまたはストアプロシージャを作成する場合は、ALTER VIEW 文や ALTER PROCEDURES 文の SET HIDDEN パラメータを使用して、ユーザがビュー定義やプロシージャ定義を参照することによって暗号化キーにアクセスできないようにすることができます。

## カラムの暗号化の例

次の例では、user\_info というテーブルのパスワードを格納するカラムを暗号化するトリガを使用します。user\_info テーブルは、次のように定義されています。

```
CREATE TABLE user_info (
  employee_ID INTEGER NOT NULL PRIMARY KEY,
  user_name CHAR(80),
  user_pwd CHAR(80) );
```

新しいユーザが追加されたとき、または既存のユーザのパスワードが更新されたときに、2つのトリガが user\_pwd カラムの値を暗号化するためにデータベースに追加されます。

- encrypt\_new\_user\_pwd トリガは、新しいローが user\_info テーブルに追加されるたびに実行されます。

```
CREATE TRIGGER encrypt_new_user_pwd
BEFORE INSERT
ON user_info
REFERENCING NEW AS new_pwd
FOR EACH ROW
BEGIN
  SET new_pwd.user_pwd=ENCRYPT(new_pwd.user_pwd, '8U3dkA');
END;
```

- encrypt\_updated\_pwd トリガは、user\_info テーブルの user\_pwd カラムが更新されるたびに実行されます。

```
CREATE TRIGGER encrypt_updated_pwd
BEFORE UPDATE OF user_pwd
ON user_info
REFERENCING NEW AS new_pwd
FOR EACH ROW
BEGIN
  SET new_pwd.user_pwd=ENCRYPT(new_pwd.user_pwd, '8U3dkA');
END;
```

### データベースに新しいユーザを追加する場合

```
INSERT INTO user_info
VALUES ( '1', 'd_williamson', 'abc123');
```

SELECT 文を発行して user\_info テーブルの情報を表示する場合、user\_pwd カラムの値はバイナリデータ (パスワードの暗号化された形式) であり、INSERT 文で指定された値 abc123 ではありません。

このユーザのパスワードを変更した場合は、encrypt\_updated\_pwd トリガが起動され、新しいパスワードが暗号化形式で user\_pwd カラムに表示されます。

```
UPDATE user_info
SET user_pwd='xyz'
WHERE employee_ID='1';
```

元のパスワードは、次の SQL 文を発行して検索できます。この文はデータを復号化するために DECRYPT 関数と暗号化キーを使用し、値を LONG BINARY から CHAR 型に変換するために CAST 関数を使用しています。

```
SELECT CAST (
  DECRYPT( user_pwd, '8U3dkA' )
  AS CHAR(100))
FROM user_info
WHERE employee_ID = '1';
```

このセクションの内容:

#### [未加工の暗号化 \[1620 ページ\]](#)

未加工の暗号化を使用すると、データベースサーバ内のデータを、エクスポートしてデータベースサーバ外で復号化できるフォーマットに暗号化できます。

### 1.9.2.8.3.1.1 未加工の暗号化

未加工の暗号化を使用すると、データベースサーバ内のデータを、エクスポートしてデータベースサーバ外で復号化できるフォーマットに暗号化できます。

このような暗号化は、未加工と呼ばれます。未加工フォーマットでデータを暗号化するには、暗号化キーと初期化ベクトルを指定する必要があります。また、必要に応じて埋め込みフォーマットも指定します。データを復号化するには、同じパラメータ値を指定する必要があります。

また、DECRYPT 関数を使用して、データベースサーバ内のデータを復号化することもできます。

未加工の暗号化は、次の場合に便利です。

#### データベースユーザがデータにアクセスできないようにする場合

未加工の暗号化を使用して、データベース管理者さえもアクセスさせない機密データを暗号化し、データベースサーバを使用しないでクライアントアプリケーションでデータを復号化することができます。未加工の暗号化は、データの暗号化と復号化をデータベースサーバのみが実行する必要がある場合はおすすめてできません。

#### TLS 暗号化を使用できない場合

未加工の暗号化は、TLS 暗号化の代わりに使用できます。TLS 暗号化とは異なり、未加工の暗号化では、リプレイや中間者攻撃を防止できません。また、データベースサーバも認証できません。

#### 例

データベースの SensitiveData テーブルにある binary\_data カラムから、データベースを使用しないクライアントにデータを送信する必要があります。機密データであるため、次の SQL 文を使用して、データを未加工フォーマットに暗号化します。

```
SELECT ENCRYPT( binary_data, 'TheEncryptionKey', 'AES (FORMAT=RAW)',  
'ThisIsTheIV' ) FROM SensitiveData;
```

暗号化データを、内容を復号化できるアプリケーションとともにクライアントにコピーします。また、アプリケーションで使用する暗号化キー (TheEncryptionKey) と初期化ベクトル (ThisIsTheIV) もクライアントに提供します。クライアントはアプリケーションを使用してデータを復号化し、表示します。

## 1.9.2.8.3.2 テーブル暗号化

テーブルの暗号化では、データベース全体の暗号化がもたらすようなパフォーマンスの低下を招くことなく、機密データが含まれるテーブルやマテリアライズドビューを暗号化できます。

テーブルの暗号化が有効な場合、暗号化されたテーブルのテーブルページ、関連するインデックスページ、テンポラリファイルのページが暗号化されます。暗号化されたテーブルのトランザクションを含むトランザクションログのページも暗号化されません。

データベース内のテーブルを暗号化するためには、テーブル暗号化を有効にしておく必要があります。テーブル暗号化の有効化は、データベースを初期化するときに行います。テーブル暗号化が有効になっているかどうかを確認するには、次のように DB\_PROPERTY 関数を使用して EncryptionScope データベースプロパティの値を取得します。

```
SELECT DB_PROPERTY( 'EncryptionScope' );
```

TABLE が返された場合は、テーブル暗号化が有効になっています。

テーブル暗号化で暗号化アルゴリズムが有効であるかどうかを確認するには、次のように DB\_PROPERTY 関数を使用して Encryption データベースプロパティの値を取得します。

```
SELECT DB_PROPERTY( 'Encryption' );
```

### テーブル暗号化がパフォーマンスに及ぼす影響

暗号化されたテーブルでは、各テーブルページがディスクへの書き込みと同時に暗号化され、ディスクから読み取るときに復号化されます。このプロセスはアプリケーションには影響しません。ただし、暗号化されたテーブルの読み込みや書き込みにおいてパフォーマンスが多少低下することがあります。既存のテーブルを暗号化または復号化する場合、テーブルのサイズによっては時間がかかることがあります。

暗号化されたテーブル内のカラムに対するインデックスのインデックスページ、暗号化されたテーブルのトランザクションを含むトランザクションログのページ、データベースのテンポラリファイルのすべてのページも暗号化されます。その他のデータベースとトランザクションログページは暗号化されません。

暗号化されたテーブルに圧縮されたカラムが含まれている場合があります。その場合、データは圧縮されてから暗号化されます。

テーブルの暗号化は必要記憶域には影響しません。

### テーブル暗号化が有効であるデータベースの起動

テーブル暗号化が有効であるデータベースを起動する方法は、暗号化されたデータベースを起動する場合と同じです。たとえば、-ek オプションを指定してデータベースを起動する場合は、キーを指定する必要があります。-ep オプションを指定してデータベースを起動すると、キーの入力を要求されます。

## 関連情報

[データベースの暗号化と復号化 \[1614 ページ\]](#)

[データベースを暗号化する方法 \[1616 ページ\]](#)

[データベースの作成 \(SQL Central の場合\) \[266 ページ\]](#)

[データベースの作成 \(dbinit ユーティリティの場合\) \[267 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

[-ep データベースサーバオプション \[406 ページ\]](#)

### 1.9.2.8.3.3 データベース内のテーブル暗号化の有効化 (SQL の場合)

テーブル暗号化が有効になっているデータベースを作成するか、または既存のデータベースでテーブル暗号化を有効化すると、あとでテーブルを暗号化することができます。

#### 前提条件

デフォルトでは、CREATE DATABASE 文と CREATE ENCRYPTED TABLE DATABASE 文を実行するには SERVER OPERATOR システム権限が必要です。-gu データベースサーバオプションを使用すると、必要な権限を変更できます。

#### コンテキスト

テーブル暗号化は、データベースの作成時に有効にして設定する必要があります。データベースでテーブル暗号化が有効でない場合、または、データベースの暗号化が有効になっている場合は、CREATE ENCRYPTED TABLE DATABASE 文を使用すると、テーブル暗号化が有効なデータベースのコピーが作成されます。また、元のデータベースファイルは上書きされません。

データベースでテーブル暗号化が有効化されると、暗号化されたテーブルを作成するか、既存のテーブルを暗号化することができます。

#### 手順

テーブル暗号化が有効なデータベースを作成するか、既存のデータベースでテーブル暗号化を有効にします。

オプション	アクション
テーブル暗号化が有効なデータベースを作成する	CREATE DATABASE 文を使用してデータベースを作成します。このとき、ENCRYPTED TABLE 句を指定し、キーと暗号化アルゴリズムを指定します。

オプション	アクション
既存のデータベースでテーブル暗号化を有効にする	CREATE ENCRYPTED TABLE DATABASE でデータベースのコピーを作成し、キーを指定します。

## 結果

テーブル暗号化は有効です。

### 例

次のコマンドは、キー abc と暗号化アルゴリズム AES256\_FIPS を使用する強力なテーブル暗号化を有効にしてデータベース mynewdemo.db を作成します。また、passwd というパスワードで、DBA というユーザ ID も作成します。

```
CREATE DATABASE 'C:¥temp¥mynewdemo.db'
  DBA USER 'DBA' DBA PASSWORD 'passwd'
  ENCRYPTED TABLE
  KEY 'abc'
  ALGORITHM 'AES256_FIPS';
```

次の例は、contacts1 という既存のデータベースから contacts2 というデータベースを作成します。新しいデータベースでは、テーブル暗号化がサポートされています。

```
CREATE ENCRYPTED TABLE DATABASE 'C:¥temp¥contacts2.db'
  FROM 'C:¥temp¥contacts1.db'
  KEY 'Sd8f6654'
  OLD KEY 'Sc8e5543';
```

## 次のステップ

CREATE TABLE 文を使用するか、暗号化する既存のテーブルを ALTER TABLE 文で変更して、データベース内のテーブルを暗号化します。テーブルを暗号化するときは、テーブル暗号化を有効にするときに指定したキー、アルゴリズム、またはその両方を使用します。

## 関連情報

[テーブルの暗号化 \[1625 ページ\]](#)

## 1.9.2.8.3.4 データベース内のテーブル暗号化の有効化 (dbinit ユーティリティの場合)

データベース作成中にテーブル暗号化を有効化します。

### 前提条件

テーブル暗号化は、データベースの作成時に有効にして設定する必要があります。データベースでテーブル暗号化が有効になっていない場合、またはデータベース暗号化が有効な場合は、テーブル暗号化を有効にしてデータベースを再作成する必要があります。

### 手順

dbinit で `-et` オプションと `-ek` オプションを指定し、キーと暗号化アルゴリズムも指定して、データベースを作成します。

### 結果

テーブル暗号化は有効です。

#### 例

次のコマンドでは、キー `abc` と `AES256_FIPS` 暗号化アルゴリズムを使用するテーブルで強力な暗号化が有効なデータベース `new.db` を作成し、`DBA` ユーザ ID を、`passwd` というパスワードで `DBA` に設定します。

```
dbinit -dba DBA,passwd new.db -et -ek abc -ea AES256_FIPS
```

このデータベース内のテーブルを暗号化するときには、`AES256_FIPS` アルゴリズムと `abc` キーが使用されます。

### 関連情報

[テーブルの暗号化 \[1625 ページ\]](#)

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

## 1.9.2.8.3.5 テーブルの暗号化

暗号化テーブルを作成するか、既存のテーブルを暗号化します。

### 前提条件

CREATE TABLE 文を使用するには、次のシステム権限のいずれかが必要です。

- CREATE TABLE
- CREATE ANY TABLE
- CREATE ANY OBJECT

ALTER TABLE 文を作成するには、変更するテーブルの所有者であるか、次のいずれかの権限を持っている必要があります。

- そのテーブルに対する ALTER 権限
- ALTER ANY TABLE
- ALTER ANY OBJECT

データベース内のテーブルを暗号化するためには、そのデータベースでテーブル暗号化が有効になっている必要があります。

### コンテキスト

テーブルを暗号化するときは、データベースの作成時に指定した暗号化のアルゴリズムとキーが使用されます。

### 手順

暗号化が有効なテーブルを作成するか、既存のテーブルを暗号化することができます。

オプション	アクション
暗号化が有効なテーブルを作成する	CREATE TABLE 文の ENCRYPTED 句を使用してテーブルを作成します。
既存のテーブルを暗号化する	ALTER TABLE 文の ENCRYPTED 句を使用してテーブルを暗号化します。

### 結果

テーブルが暗号化されます。

## 例

次のコマンドは、暗号化されたテーブル MyEmployees を作成します。

```
CREATE TABLE MyEmployees (  
  MemberID CHAR(40),  
  CardNumber INTEGER )  
ENCRYPTED;
```

次の文は、MyEmployees2 というテーブルを作成してから暗号化します。

```
CREATE TABLE MyEmployees2 (  
  MemberID CHAR(40),  
  CardNumber INTEGER );  
ALTER TABLE MyEmployees2  
  ENCRYPTED;
```

## 関連情報

[データベース内のテーブル暗号化の有効化 \(SQL の場合\) \[1622 ページ\]](#)

[データベース内のテーブル暗号化の有効化 \(dbinit ユーティリティの場合\) \[1624 ページ\]](#)

## 1.9.2.8.4 セキュリティ: 暗号化されたデータベースのキー

強かに暗号化されたデータベースやテーブルを使用した暗号化キーのセキュリティを向上させる方法を学習します。

強力な暗号化を使用してデータベースまたはデータベーステーブルを暗号化する場合、暗号化キー (パスワード) を指定する必要があります。

### 暗号化キーの指定

データベースを起動するたびに、暗号化キーに `-ek` オプションを設定する必要があります。たとえば、クライアントがデータベースを起動するたびに、暗号化キーの指定が要求されます。クライアントに暗号化キーを提供することに懸念がある場合は、常にデータベース管理者がデータベースを起動するようにします。

暗号化キーの入力に、コマンドプロンプト (デフォルト) またはプロンプトボックスのいずれかを選択できます。プロンプトボックスでのキー入力を選択すると、キーが表示されないため、さらにセキュリティが強化されます。データベースの起動時に `-ep` データベースオプションを指定すると、暗号化キーの入力を求められます。

### 暗号化キーの選択

暗号化キーを選択する際の最善の方法は、簡単には推測できない値を選択することです。また、数字、文字、特殊文字を組み合わせると、キーは推測されにくくなります。暗号化キーでは常に大文字と小文字が区別されます。また、前後のスペース

や、セミコロンを含めることはできません。暗号化キーの長さは任意です。短いと推測されやすいため、長いキーの方が適しています。

## 暗号化キーの保管

キーを忘れた場合はデータベースにまったくアクセスできなくなります。

### 警告

暗号化キーのコピーを安全な場所に保管してください。暗号化キーがわからなくなった場合は、テクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。

このセクションの内容:

[データベースの暗号化キーの変更 \(SQL\) \[1627 ページ\]](#)

暗号化されたデータベースや、テーブル暗号化が有効になっているデータベースの暗号化キーを変更します。

## 1.9.2.8.4.1 データベースの暗号化キーの変更 (SQL)

暗号化されたデータベースや、テーブル暗号化が有効になっているデータベースの暗号化キーを変更します。

### 前提条件

デフォルトでは、CREATE ENCRYPTED DATABASE 文を実行するには SERVER OPERATOR システム権限が必要です。-gu データベースサーバオプションを使用すると、必要な権限を変更できます。

### コンテキスト

暗号化キーを変更しても、既存のファイルは上書きされませんが、新しいキーで暗号化されたファイルのコピーが作成されません。暗号化キーを変更する際、異なるアルゴリズムや することができます。

### 手順

1. CREATE ENCRYPTED DATABASE 文を使用して、暗号化されたデータベースの暗号化キーを変更します。
2. 次のオプションのうちの 1 つを選択してください。

オプション	アクション
テーブル暗号化で使用する暗号化キーの変更	CREATE ENCRYPTED TABLE DATABASE 文を使用してデータベースのコピーを作成し、新しいキーを指定します。
暗号化されたデータベースで使用する暗号化キーの変更	CREATE ENCRYPTED DATABASE 文を使用してデータベースのコピーを作成し、新しいキーを指定します。

## 結果

暗号化キーが変更されます。

### 例

次の例では、abc というキーで暗号化されたデータベースファイル encryptedtemp.db のコピーを作成し、そのコピーに mynewencryptedtemp.db という名前を付けて、暗号化キー abc123 で暗号化します。他のすべてのデータベース関連ファイル (トランザクションログ、トランザクションログミラー、DB 領域ファイル) も、新しい暗号化キーを使用して作成されます。

```
CREATE ENCRYPTED DATABASE 'C:¥temp¥mynewencryptedtemp.db'
FROM 'C:¥temp¥encryptedtemp.db'
KEY 'abc123'
ALGORITHM 'AES'
OLD KEY 'abc';
```

## 1.9.2.8.5 暗号化されたデータベースの作成 (SQL の場合)

作成中にデータベースを暗号化します。

### 前提条件

デフォルトでは、SERVER OPERATOR システム権限が必要です。-gu データベースサーバオプションを使用すると、必要な権限を変更できます。

### コンテキスト

このタスクは、既存のデータベースの暗号化とは異なります。既存のデータベースを暗号化するには、CREATE ENCRYPTED DATABASE 文を使用してください。

## 警告

データベースに強力な暗号化を適用した場合、暗号化キーのコピーを必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、テクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。

## 手順

1. Interactive SQL で、既存のデータベースに接続します。
2. ENCRYPTED 句、KEY オプション、ALGORITHM オプションを含む CREATE DATABASE 文を実行します。

## 結果

暗号化されているデータベースが作成されます。

## 例

たとえば、次の文は、FIPS 認定の 128 ビット AES 暗号化および指定した暗号化キーを使用して、c:¥temp¥ ディレクトリにデータベースファイル myencrypteddb.db を作成します。DBA ユーザ ID およびパスワードが指定され、トランザクションログが有効化されました。

```
CREATE DATABASE 'c:¥temp¥myencrypteddb.db'  
DBA USER 'DBA'  
DBA PASSWORD 'passwd'  
TRANSACTION LOG ON  
ENCRYPTED ON  
  KEY '0kZ2o52AK#'  
  ALGORITHM 'AES_FIPS';
```

## 関連情報

[-gu データベースサーバオプション \[435 ページ\]](#)

## 1.9.2.8.6 暗号化されたデータベースの作成 (dbinit ユーティリティの場合)

データベースの作成時にデータベースを暗号化します。

### コンテキスト

#### 警告

データベースに強力な暗号化を適用した場合、暗号化キーのコピーを必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、テクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。

### 手順

dbinit ユーティリティを実行してデータベースを作成します。

- 強力な暗号化でデータベースをエンコードするには、-ek または -ep オプションを含めて、暗号化キーを指定します。

### 結果

エンコードされたデータベースが作成されます。

#### 例

- 次のコマンドによって、強力的に暗号化されたデータベースを作成し、暗号化キーとアルゴリズムを指定します。

```
dbinit -dba DBA,passwd -ek "0kZ2o56AK#" -ea AES_FIPS myencrypteddb.db
```

このデータベースサーバを起動するには、次のコマンドを実行します。

```
dbsrv17 myencrypteddb.db -ek "0kZ2o56AK#"
```

### 次のステップ

強力的に暗号化されたデータベースの起動時に、暗号化キーを指定する必要があります。

## 関連情報

[初期化ユーティリティ \(dbinit\) \[1113 ページ\]](#)

### 1.9.2.8.7 単純難読化によりエンコードされたデータベースの作成

データベースの作成時にエンコードオプションを指定します。

## コンテキスト

## 手順

dbinit ユーティリティを実行してデータベースを作成します。

- 単純難読化でデータベースをエンコードするには、`-ea simple` オプションを含めます。

## 結果

エンコードされたデータベースが作成されます。

### 例

- 次の例では、単純難読化を使用するデータベース `test.db` を作成し、DBA ユーザ ID とパスワードを設定します。

```
dbinit -dba DBA,passwd -ea simple test.db
```

## 1.9.2.8.8 既存のデータベースの暗号化 (CREATE ENCRYPTED DATABASE 文)

強かに暗号化されたデータベースのコピーとともに、すべての関連トランザクションログ、トランザクションログミラー、DB 領域のファイルの暗号化されたコピーも作成します。

### 前提条件

デフォルトでは、CREATE ENCRYPTED DATABASE 文を実行するには SERVER OPERATOR システム権限が必要です。-gu データベースサーバオプションを使用すると、必要な権限を変更できます。

暗号化しようとしているデータベースは実行中でないことが必要です。

### コンテキスト

CREATE ENCRYPTED DATABASE 文では、ファイルのコピーを作成 (この場合、暗号化形式で) します。元のデータベースファイルは上書きしません。

#### 警告

データベースに強力な暗号化を適用した場合、暗号化キーのコピーを必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、テクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。

### 手順

1. 暗号化するデータベースを起動したあとシャットダウンし、データベースの自動リカバリが必要ない状態にします。  
データベースを正常にシャットダウンできない場合、CREATE ENCRYPTED DATABASE 文ではなく CREATE ENCRYPTED FILE 文を使用する必要があります。
2. Interactive SQL で、ユーティリティデータベース、または暗号化するデータベースではない別のデータベースに接続します。

たとえば、ユーティリティデータベースに接続して、次の操作を行います。

1. データベースサーバを起動します。

```
dbsrv17 -n myserver -su myuser -ID,mypassword
```

2. ユーティリティデータベースに接続します。

Interactive SQL で、[接続ウィンドウを開き](#)、[アクション](#)として[接続文字列](#)を使用して[接続](#)を選択し、[接続文字列](#)に次の内容を指定します。

```
"UID=DBA;PWD=mypassword;Server=myserver;DBN=utility_db;ASTART=No"
```

3. 暗号化されたデータベースのコピーを、CREATE ENCRYPTED DATABASE 文を実行して作成します。

リカバリが必要なため読み取り専用モードでデータベースを起動できないというエラーが表示される場合、代わりに CREATE ENCRYPTED FILE 文を使用する必要があります。

## 結果

データベースファイルのコピーは、暗号化して作成されます。データベースに関連付けられたトランザクションログ、トランザクションログミラー、または DB 領域がある場合は、これらのファイルの暗号化されたコピーも作成されます。

データベースでトランザクションログまたはトランザクションログミラーが使用されている場合、new という単語がファイル名の最初に追加されます。たとえば、トランザクションログ filename.log は、newfilename.log に変更されます。データベースに DB 領域ファイルが含まれている場合は、暗号化を示す E がファイルの拡張子に付加されます。たとえば、ファイル mydbspace.dbs は、mydbspace.dbsE に変更されます。

### 例

次の文では、データベースファイル temp.db の AES で暗号化されたコピーを作成し、そのコピーに encryptedtemp.db という名前を付けます。

```
CREATE ENCRYPTED DATABASE 'C:¥¥Users¥¥Public¥¥Documents¥¥SQL Anywhere 17¥¥Samples¥¥temp.db'  
FROM 'C:¥¥Users¥¥Public¥¥Documents¥¥SQL Anywhere 17¥¥Samples¥¥demo.db'  
KEY 'abcpassword'  
KDI '2000'  
ALGORITHM 'AES';
```

## 次のステップ

暗号化キーを使用して暗号化されたデータベースのコピーを起動し、接続できることを確認します。

## 関連情報

[CREATE ENCRYPTED DATABASE 文と CREATE ENCRYPTED FILE 文の比較 \[1617 ページ\]](#)

### 1.9.2.8.9 既存のデータベースの暗号化 (CREATE ENCRYPTED FILE 文)

CREATE ENCRYPTED DATABASE 文が使用できない場合に、既存のデータベースとその関連ファイルの暗号化されたコピーを作成します。たとえば、サポートの目的でデータベースの暗号化が必要な場合、CREATE ENCRYPTED DATABASE 文からエラーが返されます。

## 前提条件

デフォルトでは、SERVER OPERATOR システム権限が必要です。-gu データベースサーバオプションを使用すると、必要な権限を変更できます。

暗号化されるデータベースは実行できません。

## コンテキスト

CREATE ENCRYPTED DATABASE 文は、暗号化されたデータベースのコピーを作成するのにおすすめの方法です。CREATE DECRYPTED DATABASE 文からエラーが返された場合、次は CREATE DECRYPTED FILE 文を使用します。

暗号化されるデータベースには、暗号化が有効化されたテーブルを持つことはできません。

データベースファイルだけでなく、関連するファイル (トランザクションログファイル、トランザクションログミラーファイル、DB 領域ファイル) が存在する場合はそれらのファイルについても個別に CREATE ENCRYPTED FILE 文を実行する必要があります。

## 手順

1. Interactive SQL で、ユーティリティデータベース、または暗号化するデータベースではない別のデータベースに接続します。

たとえば、ユーティリティデータベースに接続して、次の操作を行います。

1. データベースサーバを起動します。

```
dbsrv17 -n server-name -su myuser -ID,mypassword
```

2. ユーティリティデータベースに接続します。

Interactive SQL で、[接続](#)ウィンドウを開き、[アクション](#)として[接続文字列](#)を使用して接続を選択し、接続文字列に次の内容を指定します。

```
"UID=DBA;PWD=mypassword;Server=myserver;DBN=utility_db;ASTART=No"
```

2. 暗号化されたデータベースのコピーを、CREATE ENCRYPTED FILE 文を実行して作成します。

次に例を示します。

```
CREATE ENCRYPTED FILE 'temp2.db'  
FROM 'C:¥¥Users¥¥Public¥¥Documents¥¥SQL Anywhere 17¥¥Samples¥¥demo.dbdemo.db'  
KEY 'Sd8f6654*Mnn'  
ALGORITHM 'AES_FIPS';
```

### 警告

データベースに強力な暗号化を適用した場合、暗号化キーのコピーを必ず安全な場所に保管してください。暗号化キーがわからなくなった場合は、テクニカルサポートに依頼してもデータにはアクセスできません。アクセスできなくなったデータベースは、廃棄して、新しくデータベースを作成する必要があります。

3. トランザクションログファイル、トランザクションログミラーファイル、DB 領域ファイルなどの各関連ファイルに対して CREATE ENCRYPTED FILE 文を実行し、データベースファイルの暗号化に使用したのと同じ暗号化キーを指定します。

次に例を示します。

```
CREATE ENCRYPTED FILE 'temp2.log'  
FROM 'C:\Users\Public\Documents\SQL Anywhere 17\Samples\demo.log'  
KEY 'Sd8f6654*Mnn'  
ALGORITHM 'AES_FIPS';
```

## 1.9.2.8.10 データベースの復号化 (SQL の場合)

データベースとともに、関連するトランザクションログファイル、トランザクションログミラーファイル、DB 領域ファイルを複合化します。

### 前提条件

デフォルトでは、CREATE DECRYPTED TABLE DATABASE 文を実行するには SERVER OPERATOR システム権限が必要です。-gu データベースサーバオプションを使用すると、必要な権限を変更できます。

### コンテキスト

CREATE DECRYPTED DATABASE 文は、データベースの復号化に推奨されるメソッドです。CREATE DECRYPTED 文からエラーが返された場合、代わりに CREATE DECRYPTED FILE 文を使用する必要があります。

暗号化されているデータベースが実行中である必要があります。

### 手順

1. 復号化するデータベースを起動したあとシャットダウンし、データベースの自動リカバリが必要ない状態にします。  
データベースの起動または正常なシャットダウンができない場合、CREATE DECRYPTED DATABASE 文ではなく CREATE DECRYPTED FILE 文を使用する必要があります。
2. Interactive SQL で、ユーティリティデータベース、または復号化するデータベースではない別のデータベースに接続します。
3. CREATE DECRYPTED DATABASE 文を実行します。

リカバリが必要なため読み取り専用モードでデータベースを起動できないというエラーが表示される場合、代わりに CREATE DECRYPTED FILE 文を使用する必要があります。

## 結果

データベースファイルのコピーが復号化された状態で作成されます。元のデータベースファイルは上書きされません。データベースに関連付けられたトランザクションログ、トランザクションログミラー、または DB 領域がある場合は、これらのファイルの復号化されたコピーも作成されます。

### 例

最初の文では、temp.db の AES256 で暗号化されたコピー encryptedtemp.db を作成します。2 番目の文では、encryptedtemp.db の復号化されたコピー decryptedtemp.db を作成します。

```
CREATE ENCRYPTED DATABASE 'C:¥temp¥encryptedtemp.db'  
  FROM 'C:¥temp¥temp.db'  
  KEY 'abc'  
  ALGORITHM 'AES256';  
CREATE DECRYPTED DATABASE 'C:¥temp¥decryptedtemp.db'  
  FROM 'C:¥temp¥encryptedtemp.db'  
  KEY 'abc';
```

## 1.9.2.8.11 暗号化を使用した場合のパフォーマンスの問題

データベースが暗号化されている場合、パフォーマンスは低下します。パフォーマンスの影響は、ディスクとのページの読み取りや書き込みの頻度によって異なります。また、サーバが使用するキャッシュサイズを適切に設定することによって影響を最小限にできます。

キャッシュの初期サイズを増やすには、サーバの起動時に -c オプションで指定します。キャッシュの動的なサイズ変更がサポートされているオペレーティングシステムでは、使用されるキャッシュサイズが、使用可能なメモリの容量によって制限される場合があります。そのため、キャッシュサイズを増加するには、使用可能なメモリを増加します。

## 関連情報

[ヒント: パフォーマンス向上のためのキャッシュの使用 \[1370 ページ\]](#)

[-c データベースサーバオプション \[385 ページ\]](#)

## 1.9.2.9 公開鍵と秘密鍵を使用したメッセージの暗号化

RSA キーペアを使用して、メッセージの暗号化、署名、および検証を行います。

### RSA キーペアを使用したメッセージの暗号化

コンピュータまたはネットワークがセキュアであるか不明で、目的の受信者だけがデータにアクセスできるようにする必要があるときに、小さなブロックのデータをリモートコンピュータに送信するには、メッセージの暗号化に RSA キーペアを使用します。

sp\_generate\_key\_pair システムプロシージャを使用して、秘密鍵と公開鍵からなるキーペアを作成します。RSA 暗号化では、秘密鍵と公開鍵の 2 つの数学的に関連したキーを使用します。どちらのキーも PEM 形式で返されます。大きいキーの作成には数分かかることがあります。生成された公開鍵および秘密鍵の長さは key\_size パラメータによります。公開鍵の長さはおおよそ  $80+s/6$  バイトで、秘密鍵の長さはおおよそ  $110+7*s/9$  バイトです。ここで s はビット単位のキーサイズです。2048 ビット未満のキーは安全ではありません。指定された key\_size が完全な値を保持するには小さすぎる場合は、変更されません。暗号化されたメッセージの最大サイズは、PKCS1 埋め込みの場合 key-size から 11 を引いた長さで、OAEP 埋め込みの場合 key-size から 42 を引いた長さ (OAEP 埋め込みの場合) です。PADDING=NONE の場合、メッセージの最大サイズはキーサイズと等しくなります。公開鍵はデータを暗号化し、それに対応する秘密鍵はデータを復号化します。暗号化と復号化に同じキーを使用することはできません。

### RSA キーペアによるメッセージの署名と検証

メッセージのプライバシーは重要ではないものの、受信者にそのメッセージが認証されたことと、送信されてから変更されていないことを示したいときには、RSA キーペアを使用してデータに署名し検証します。キーペアを作成した後、SECURE\_SIGN\_MESSAGE 機能を使用してメッセージにデジタル署名をします。受信者は SECURE\_VERIFY\_MESSAGE 機能を使用してメッセージを検証します。秘密鍵を使用して署名したメッセージは、公開鍵を使用してのみ検証することができます。データの署名では、メッセージの暗号化ハッシュを作成し、送信者のプライベートキーを使用してハッシュを暗号化します。受信者は、送信者の公開鍵を使用して復号化することにより署名を検証し、メッセージのハッシュを計算して 2 つのハッシュを比較します。両者が一致する場合、受信者は送信者の秘密鍵にアクセスできるユーザだけがメッセージを送信できたことと、メッセージが変更されていないことがわかります。

このセクションの内容:

#### [キーペアの作成 \[1638 ページ\]](#)

RSA キーペアを作成して、メッセージを暗号化するときやデジタルメッセージに署名し検証するときに使用します。

#### [メッセージの暗号化と復号化 \[1638 ページ\]](#)

RSA キーペアを使用してメッセージを暗号化します。

#### [メッセージの署名および検証 \[1639 ページ\]](#)

RSA キーペアを使用して、メッセージに署名および検証をします。

## 1.9.2.9.1 キーペアの作成

RSA キーペアを作成して、メッセージを暗号化するときやデジタルメッセージに署名し検証するときに使用します。

### コンテキスト

RSA アルゴリズムを使用してデータを暗号化する機能を使用するには、最初に RSA 秘密鍵および公開鍵を作成します。

### 手順

1. Interactive SQL では、次の文を実行して秘密鍵および公開鍵を保持する変数を作成します。

```
CREATE VARIABLE @publickey LONG VARCHAR;  
CREATE VARIABLE @privatekey LONG VARCHAR;
```

2. 次の文を実行して、2048ビット RSA キーペアを生成します。

```
CALL sp_generate_key_pair( 2048, @publickey, @privatekey );
```

3. 秘密鍵および公開鍵をテーブルまたはファイルに保存します。公開鍵を発行するか、またはそれを必要なすべての人に送信することはできますが、秘密鍵はセキュアに保つ必要があります。

### 結果

秘密鍵と公開鍵からなる RSA キーペアが作成され、データベーステーブルまたはファイルに保存されました。

## 1.9.2.9.2 メッセージの暗号化と復号化

RSA キーペアを使用してメッセージを暗号化します。

### 前提条件

メッセージの受信者は RSA キーペアを作成し、公開鍵のみをメッセージの送信者に送信します。

## コンテキスト

メッセージの暗号化は、ネットワーク接続やリモートコンピュータがセキュアであるか不明で、メッセージの受信者だけがデータを読めるようにする必要があるときに、少量のデータをリモートコンピュータに送信するのに便利です。

下のシナリオでは、メッセージの送信者とメッセージの受信者が同じデータベースにアクセス可能です。

## 手順

1. 送信者は、目的の受信者の公開鍵でメッセージを暗号化します。たとえば、送信者は次の文を実行して受信者の公開鍵でメッセージを暗号化し、暗号化したメッセージと受信者の名前をデータベースに挿入します。

```
INSERT INTO DataTable ( data, recipient );
      VALUES ( ENCRYPT( 'Message text', @RecipientsPublicKey, 'RSA' ), 'recipient-
name');
```

メッセージを復号化し読むことができるのは、受信者の秘密鍵にアクセスできる人物のみです。

2. メッセージの受信者は、秘密鍵を使用してメッセージを復号化します。例:

```
SELECT DECRYPT ( data, @RecipientsPrivateKey, 'RSA' )
AS decrypted_message
FROM DataTable
WHERE recipient = 'recipient-name';
```

文により復号化されたメッセージが返されます。

## 結果

メッセージは送信者によって暗号化され、データベースに挿入され、メッセージの受信者によって復号化され読まれます。

### 1.9.2.9.3 メッセージの署名および検証

RSA キーペアを使用して、メッセージに署名および検証をします。

## 前提条件

RSA キーペアを作成してパブリックキーを公開し、メッセージの受信者がアクセスできるようにします。

## コンテキスト

メッセージにデジタル署名をして、受信者にメッセージが認証されたことと送信されてから変更されていないことを保証します。

## 手順

1. メッセージの送信者は、次の文を実行して署名を作成し、シグニチャを使用してメッセージを署名します。

```
CREATE VARIABLE @signature LONG BINARY;  
SELECT SECURE_SIGN_MESSAGE( document, private-key ) INTO @signature;
```

2. メッセージの送信者は、メッセージの受信者に、シグニチャとともにドキュメントを送信します。
3. メッセージの受信者は、次の文を実行し、メッセージが認証されたことと、送信されてから変更されていないことを確認します。

```
CREATE VARIABLE @verified INTEGER;  
SELECT SECURE_VERIFY_MESSAGE( document, signature, senders-public-key ) INTO  
@verified;
```

@verified 変数が 1 と等しい場合、メッセージは認証されています。

## 結果

署名されたメッセージが送信され、受信者がその信頼性を検証しました。

### 1.9.3 トランスポートレイヤセキュリティ

トランスポートレイヤセキュリティ (TLS) は IETF 標準プロトコルであり、デジタル証明書とパブリックキー暗号方式を使用して、クライアント/サーバ通信をセキュリティ保護します。

トランスポートレイヤセキュリティにより、暗号化、改ざん検出、証明書ベースの認証が実現します。

トランスポートレイヤセキュリティは、次の場合に使用できます。

- データベースサーバとクライアントアプリケーションとの間の通信のセキュリティ保護。
- Mobile Link サーバと Mobile Link クライアントとの間の通信のセキュリティ保護。
- セキュアな SQL Anywhere Web サーバの設定。

セキュリティ保護された通信は、サーバ認証を含む次のメッセージの交換 (ハンドシェイク) で始まります。

トランスポートレイヤセキュリティでは、セキュア接続の確立と保護にサーバ証明書を使用します。サーバごとにユニークな証明書ファイルを作成します。

SQL Anywhere クライアント/サーバ通信または Mobile Link 同期用にサーバ認証を使用できます。

- クライアント/サーバ通信では、データベースクライアントが SQL Anywhere データベースサーバの ID を検証します。

- Mobile Link 同期では、Mobile Link クライアント (SQL Anywhere または Ultra Light) が Mobile Link サーバの ID を検証します。

FIPS 認定の暗号化には別途ライセンスが必要です。

## 効率化

トランスポートレイヤセキュリティプロトコルは、パブリックキー暗号化と対称キー暗号化の組み合わせを使用します。パブリックキー暗号化は、認証テクニックとして優れていますが、処理量が多くなります。セキュア接続が確立されると、クライアントとサーバはそれ以降の通信に、キーサイズが 128 ビットで高効率の対称暗号を使用します。

## 証明書

証明書作成ユーティリティ (createcert) を使用して、トランスポートレイヤセキュリティのための X.509 証明書ファイルを作成します。一方、サードパーティ証明書の存在を確認する必要がある場合、またはセキュリティのより高い証明書が必要な場合は、証明書を認証局から購入してください。

このセクションの内容:

### [TLS サポート \[1642 ページ\]](#)

バージョン 1.0、1.1、および 1.2 のトランスポートレイヤセキュリティ (TLS) 暗号化がサポートされています。

### [FIPS 認定の暗号化テクノロジー \[1642 ページ\]](#)

FIPS 認定の暗号化アルゴリズムを使用すると、データベースファイルを暗号化したり、データベースクライアント/サーバ通信、Web サービス、クライアント/サーバ通信における通信を暗号化できます。

### [トランスポートレイヤセキュリティの設定方法 \[1643 ページ\]](#)

トランスポートレイヤセキュリティをシステムで設定します。

### [デジタル証明書 \[1644 ページ\]](#)

トランスポートレイヤセキュリティを設定するには、デジタル証明書が必要です。

### [クライアント側 TLS 証明書の使用 \[1652 ページ\]](#)

クライアントが信頼できる発行元の署名のある証明書を使用する場合に限り、クライアント側の証明書を使用してユーザのデータベースサーバに接続することができます。

### [SQL Anywhere クライアント/サーバ通信の暗号化 \[1653 ページ\]](#)

SQL Anywhere クライアント/サーバ通信は、トランスポートレイヤセキュリティを使用して暗号化できます。

### [SQL Anywhere Web サービス暗号化 \[1658 ページ\]](#)

SQL Anywhere Web サーバは、TLS バージョン 1.0 以降を使用する HTTPS 接続をサポートします。

## 関連情報

### [データベースの暗号化と復号化 \[1614 ページ\]](#)

## 1.9.3.1 TLS サポート

バージョン 1.0、1.1、および 1.2 のトランスポートレイヤセキュリティ (TLS) 暗号化がサポートされています。

### RSA 暗号化

未認定 RSA 暗号化はソフトウェアに含まれており、クライアント/サーバ通信、同期、Web サービスに使用できます。このバージョンは FIPS 認定ではありません。

### FIPS 認定 RSA 暗号化

FIPS 認定 RSA 暗号化は、別々のライセンスで使用できます。

## 1.9.3.2 FIPS 認定の暗号化テクノロジー

FIPS 認定の暗号化アルゴリズムを使用すると、データベースファイルを暗号化したり、データベースクライアント/サーバ通信、Web サービス、クライアント/サーバ通信における通信を暗号化できます。

米国連邦情報処理規格 (FIPS) 140-2 では、暗号化アルゴリズムのセキュリティ要件が定められています。FIPS 140-2 認定プログラムは、米国商務省標準技術局 (NIST: National Institute of Standards and Technology) とカナダ Communications Security Establishment (CSE) の間の共同の取り組みです。カナダとアメリカの連邦当局は、FIPS 104-2 認定製品を受け付けます。

SQL Anywhere は、OpenSSL の暗号化用の 32 ビットおよび 64 ビット FIPS 準拠のモジュールを提供します。Windows では、64 ビットシステムで 64 ビットモジュール (ライブラリ) を使用する必要があります。

### FIPS 認定 RSA 暗号化

FIPS 認定の暗号化には別途ライセンスが必要です。

### FIPS 認定の暗号化を有効にする

-fips データベースサーバオプションを使用して、データベースサーバにおける FIPS 認定の暗号化を使用することができます。また、-fips オプションを Mobile Link サーバで使用することもできます。

Encryption 接続パラメータの FIPS ネットワークプロトコルオプションを使用して、クライアント / サーバ通信に FIPS 認定の暗号化を使用することが選択できます。

## 関連情報

[単純難読化と強力な暗号化 \[1615 ページ\]](#)

### 1.9.3.3 トランスポートレイヤセキュリティの設定方法

トランスポートレイヤセキュリティをシステムで設定します。

#### 前提条件

デジタル証明書を取得します。ID ファイルと証明書ファイルが必要です。

サーバ ID ファイルにはプライベートキーが含まれているので、データベースサーバまたは Mobile Link サーバにセキュリティ保護された状態で格納する必要があります。

署名者 (root) の証明書ファイルはクライアントに配布します。署名者の証明書ファイルは、サーバ ID ファイルへの署名に使用されるものです。

証明書は、証明書認証局から購入するか、証明書作成ユーティリティ (createcert) を使用して作成することができます。証明書を作成する機能もあり、これは特に開発やテストのときに便利です。

#### 手順

1. SQL Anywhere クライアント/サーバアプリケーション用にトランスポートレイヤセキュリティを設定する場合:

トランスポートレイヤセキュリティを指定して **SQL Anywhere** データベースサーバを起動します。

-ec データベースサーバオプションを使用して、セキュリティ (TLS) のタイプ、サーバ ID ファイル名、および ID ファイルのプライベートキーの保護に使用されるパスワードを指定します。

共有メモリを経由した暗号化されていない接続も許可する場合は、-es オプションを指定します。

トランスポートレイヤセキュリティを使用するようにクライアントアプリケーションを設定します。

Encryption 接続パラメータ (ENC) を使用して、信頼できる証明書のパスとファイル名を指定します。

2. SQL Anywhere Web サービス用にトランスポートレイヤセキュリティを設定する場合:

トランスポートレイヤセキュリティを指定して **SQL Anywhere** データベースサーバを起動します。

-xs データベースサーバオプションを使用して、セキュリティのタイプ (HTTPS)、サーバ ID ファイル名、および ID ファイルのプライベートキーの保護に使用されるパスワードを指定します。

ブラウザまたは他の Web クライアントが証明書を信頼するように設定します。

SQL Anywhere Web サービスを暗号化します。

3. Mobile Link 同期用のトランスポートレイヤセキュリティを設定する場合は、次の手順に従います。

トランスポートレイヤセキュリティを指定して **Mobile Link** サーバを起動する

mIsrv17 -x オプションを使用して、セキュリティストリームとサーバ ID ファイル名を指定し、ID ファイルのプライベートキーの保護に使用するパスワードも指定します。

トランスポートレイヤセキュリティを使用するように **Mobile Link** クライアントを設定します。

Mobile Link 同期クライアントユーティリティ (dbmlsync) または Ultra Light アプリケーションを使用して、適切なセキュリティまたはネットワークプロトコルオプションを指定します。セキュリティストリームと、信頼されたサーバ証明書ファイル名を指定します。

## 結果

これで、システムに対するトランスポートレイヤセキュリティの設定が完了しました。

## 関連情報

[デジタル証明書 \[1644 ページ\]](#)

[トランスポートレイヤセキュリティを指定したデータベースサーバ \[1654 ページ\]](#)

[トランスポートレイヤセキュリティを使用するようにクライアントアプリケーションを設定する \[1655 ページ\]](#)

[SQL Anywhere Web サービス暗号化 \[1658 ページ\]](#)

[-xs データベースサーバオプション \[507 ページ\]](#)

### 1.9.3.4 デジタル証明書

トランスポートレイヤセキュリティを設定するには、デジタル証明書が必要です。

証明書は、証明書認証局から入手するか、証明書作成ユーティリティ (createcert) を使用して作成します。

#### オペレーティングシステム証明書ストアの証明書

デフォルトでは、セキュア接続はコンピュータのオペレーティングシステム証明書ストアを使用し、セキュア接続用に信頼できる証明書を取得します。

Windows オペレーティングシステムの HTTPS を除くすべてのセキュア接続では、格納された証明書がキャッシュされます。キャッシュは 24 時間ごとに再ロードされます。最初のセキュア接続の後、24 時間以内に必要な証明書がインストールされた場合、キャッシュが再ロードされるまで、その証明書を必要とする接続は失敗します。24 時間が経過する前に証明書にアクセスできるようにするには、サーバを再起動します。

Windows では、コマンドプロンプトで次のコマンドを実行して、証明書ストアにアクセスします。

```
certmgr.msc
```

## 証明書作成ユーティリティ

証明書作成ユーティリティ (createcert) で、RSA を使用して X.509 証明書ファイルを生成します。

## 証明書ビューアユーティリティ

証明書ビューアユーティリティ (viewcert) で、RSA を使用して X.509 証明書を読み込みます。

## サーバ認証に使用する証明書

証明書は、証明書認証局から入手するか、証明書作成ユーティリティ (createcert) を使用して、サーバ認証に使用する証明書ファイルを作成します。どちらの場合も、ID ファイルと証明書ファイルを作成します。

サーバ認証の場合は、サーバ ID ファイルとクライアントに配布する証明書ファイルを作成します。

## 証明書設定

証明書には、自己署名されたものと、民間の認証機関によって署名されたものがあります。

### 自己署名証明書

自己署名されたサーバ証明書は、単純な設定で使用します。

### ルート証明書

ルート証明書を使用すると、サーバ証明書に署名できるため、複数のサーバが配備されている環境でのデータの整合性と拡張性が向上します。

- サーバ証明書の署名に使用するプライベートキーは、安全な中央のロケーションに保存できます。
- サーバ認証では、クライアントを再設定しなくても、Mobile Link サーバまたはデータベースサーバを追加できます。

### 民間認証局

ルート証明書の代わりに、サードパーティの認証機関を使用できます。民間の認証機関は、プライベートキーを保存するための専用の設備を備えており、高品質なサーバ証明書を作成します。

このセクションの内容:

### [自己署名ルート証明書 \[1646 ページ\]](#)

自己署名ルート証明書は、単一の Mobile Link サーバまたはデータベースサーバを使用する単純な設定に使用できます。

### [証明書チェーン \[1647 ページ\]](#)

複数の ID ファイルが必要な場合は、自己署名証明書の代わりに証明書チェーンを使用することで、セキュリティと拡張性を高めることができます。

### [グローバル署名証明書 \[1649 ページ\]](#)

グローバル署名証明書は、証明書要求への署名に使用される高品質の証明書です。

## 関連情報

[ルート証明書 \[1649 ページ\]](#)

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

[証明書ビューアユーティリティ \(viewcert\) \[1087 ページ\]](#)

### 1.9.3.4.1 自己署名ルート証明書

自己署名ルート証明書は、単一の Mobile Link サーバまたはデータベースサーバを使用する単純な設定に使用できます。

#### → ヒント

サーバ ID ファイルが複数必要な場合は、証明書チェーンまたは民間の認証機関を使用します。認証機関は、ルートプライベートキーを格納する専用の設備で、拡張性と証明書の高度な整合性を提供します。

#### 証明書

サーバ認証証明書の場合、自己署名証明書はクライアントに配布されます。自己署名証明書は、識別情報、サーバのパブリックキー、自己署名されたデジタル署名を含む電子文書です。

#### ID ファイル

サーバ認証証明書の場合、ID ファイルはセキュリティ保護された状態で Mobile Link サーバまたはデータベースサーバに格納されます。ID ファイルは、自己署名証明書 (クライアントに配布) と、対応するプライベートキーを組み合わせたものです。プライベートキーがあると、Mobile Link サーバまたはデータベースサーバは、初期ハンドシェイクでクライアントから送信されたメッセージを復号化できます。

## 関連情報

[証明書チェーン \[1647 ページ\]](#)

[トランスポートレイヤセキュリティを指定したデータベースサーバ \[1654 ページ\]](#)

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

## 1.9.3.4.2 証明書チェーン

複数の ID ファイルが必要な場合は、自己署名証明書の代わりに証明書チェーンを使用することで、セキュリティと拡張性を高めることができます。

証明書チェーンでは、ID の署名にルート証明書が必要です。このルート証明書は、自分で作成するかまたは認証機関から入手することができます。

### 証明書チェーンを使用する利点

証明書チェーンには、次の利点があります。

#### 拡張性

サーバ認証では、特定のルート証明書によって署名されたすべての証明書を信頼するようにクライアントを設定できます。

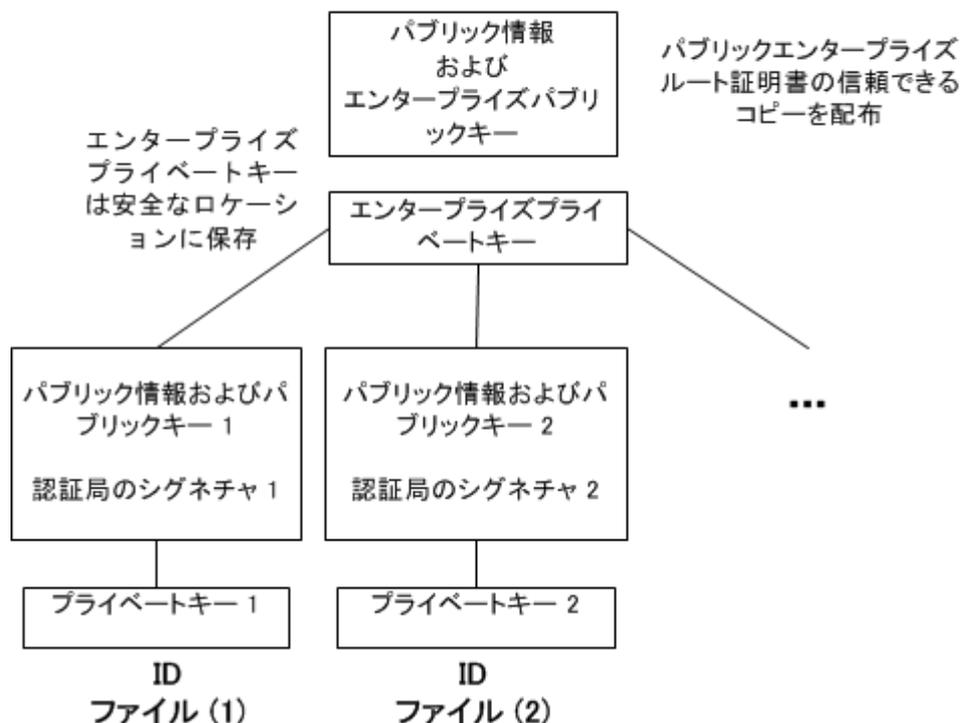
新しい Mobile Link サーバまたはデータベースサーバを追加する場合、クライアントに新しい証明書のコピーは不要です。

#### セキュリティ

ルート証明書のプライベートキーは、ID ファイルにはありません。ルート証明書のプライベートキーを高セキュリティのロケーションに保存するか、専用の設備を備えている認証機関を使用することで、サーバ認証の整合性が保護されます。

次の図は、ルート証明書の基本アーキテクチャを示しています。

## パブリックエンタープライズ ルート証明書およびプライ ベートキー



## マルチサーバ環境での証明書の使用

マルチサーバ環境で使用する証明書を作成するには、次の手順に従います。

- パブリックルート証明書およびプライベートキーを生成します。  
プライベートキーは安全なロケーションに保存します。専用の設備の方が安全です。  
サーバ認証の場合は、パブリックルート証明書をクライアントに配布します。
- ルート証明書を使用して、IDに署名します。  
パブリックルート証明書とプライベートキーを使用して、各IDに署名します。サーバ認証の場合は、IDファイルをサーバ用に使用します。

サードパーティの認証局を使用して、サーバ証明書に署名することもできます。民間認証局は、プライベートキーを保存するための専用の設備を備えており、高品質なサーバ証明書を作成します。

このセクションの内容:

### [ルート証明書 \[1649 ページ\]](#)

ルート証明書を使用すると、複数のサーバが配備されている環境での、データの整合性と拡張性が向上します。

### [署名付き ID ファイル \[1649 ページ\]](#)

ルート証明書を使用して、データベースサーバの ID ファイルに署名できます。

## 関連情報

[自己署名ルート証明書 \[1646 ページ\]](#)

[グローバル署名証明書 \[1649 ページ\]](#)

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

### 1.9.3.4.2.1 ルート証明書

ルート証明書を使用すると、複数のサーバが配備されている環境での、データの整合性と拡張性が向上します。

- 信頼できる証明書を作成するために使用されるプライベートキーは、専用の設備に保存できます。
- データベースサーバ認証では、クライアントを再設定しなくてもデータベースサーバを追加できます。

ルート証明書を設定するには、ルート証明書と、ID の署名に使用するプライベートキーを作成します。

## 関連情報

[署名付き ID ファイル \[1649 ページ\]](#)

[グローバル署名証明書 \[1649 ページ\]](#)

### 1.9.3.4.2.2 署名付き ID ファイル

ルート証明書を使用して、データベースサーバの ID ファイルに署名できます。

データベースサーバ認証の場合、各サーバ用に ID ファイルを生成します。これらの証明書はルート証明書によって署名されるため、createcert -s オプションを使用します。

## 関連情報

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

### 1.9.3.4.3 グローバル署名証明書

グローバル署名証明書は、証明書要求への署名に使用される高品質の証明書です。

これらの高品質の証明書は、民間認証局によって作成および管理されます。

グローバル署名証明書には、次の利点があります。

- 会社内での通信の場合、外部の認可された認証局を共通の信頼機関として使用すると、システムのセキュリティの信頼性が高まります。認証局は、署名を行ったすべての証明書の識別情報が正確であることを保証する必要があります。
- 認証局は、証明書を生成するための管理された環境と高度な方法を提供します。
- ルート証明書のプライベートキーは、秘密にしておきます。企業内ではこの重要情報を格納するのに適した場所がない可能性があります。認証局では専用の設備を設計して管理できます。

## グローバル署名証明書の設定

グローバル署名 ID ファイルを設定するには、次の手順に従います。

- `-r` オプションを指定した `createcert` ユーティリティを使用して証明書を要求します。
- 認証局を使用して各要求に署名します。署名付き要求と対応するプライベートキーを組み合わせ、サーバの ID ファイルを作成できます。

### i 注記

ルート証明書にグローバル署名できる場合があります。これは、認証機関が、他の証明書に署名できる証明書を生成する場合のみ適用されます。

このセクションの内容:

#### [グローバル署名 ID ファイル \[1650 ページ\]](#)

グローバル署名証明書を直接、サーバの ID ファイルとして使用できます。

#### [認証局の証明書に対するクライアントの信頼設定 \[1651 ページ\]](#)

サーバ認証の場合は、サーバにアクセスするクライアントがチェーン内のルート証明書を信頼することを確認する必要があります。

## 関連情報

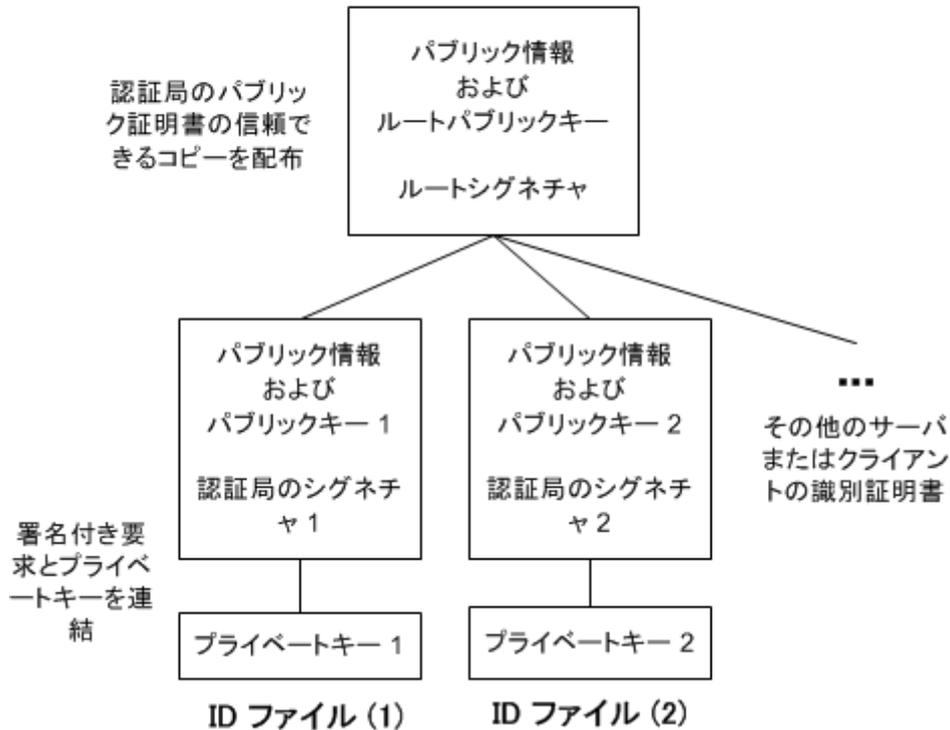
[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

### 1.9.3.4.3.1 グローバル署名 ID ファイル

グローバル署名証明書を直接、サーバの ID ファイルとして使用できます。

次の図は、複数の ID ファイルの設定を示します。

## 認証局のルート証明書



dsrv17 または mlsrv17 コマンドラインで、サーバ ID ファイルと、プライベートキーのパスワードを参照します。

## 関連情報

[トランスポートレイヤセキュリティを指定したデータベースサーバ \[1654 ページ\]](#)

### 1.9.3.4.3.2 認証局の証明書に対するクライアントの信頼設定

サーバ認証の場合は、サーバにアクセスするクライアントがチェーン内のルート証明書を信頼することを確認する必要があります。

グローバル署名証明書の場合、ルート証明書は証明書認証局の証明書です。

グローバル署名証明書を使用する場合、各クライアントは証明書のフィールド値を確認して、同じ認証局が他のクライアント用に署名した証明書を信頼することを避けなければなりません。

## 関連情報

[トランスポートレイヤセキュリティを指定したデータベースサーバ \[1654 ページ\]](#)

### 1.9.3.5 クライアント側 TLS 証明書の使用

クライアントが信頼できる発行元の署名のある証明書を使用する場合に限り、クライアント側の証明書を使用してユーザのデータベースサーバに接続することができます。

#### コンテキスト

この例では、必要な証明書を作成し、単一のルート証明書を使用してクライアント証明書やデータベースサーバ証明書に署名していることを前提としています。運用環境では、2つの異なるルート証明書を使用して、証明機関からデータベースサーバやクライアント証明書を購入する場合があります。

#### 手順

1. 証明書作成ユーティリティ (createcert) を使用してルート証明書を作成します。たとえば、次のコマンドをすべて 1 行に入力して実行します。

```
createcert -b 2048 -x -ca 1 -co myroot.crt -io myroot.id -ko myroot.pk -kp
root_pw -m 123 -sc CA -sst Ontario
-sl Waterloo -so MyCompany -sou MyUnit -scn "Sample root certificate" -u 6,7 -v
10
```

2. createsert を使用して、データベースサーバ証明書を作成します。たとえば、MyServerHost をデータベースサーバが動作しているコンピュータのホスト名に入れ替え、次のコマンドをすべて 1 行に入力して実行します。

```
createcert -b 2048 -c myroot.crt -ck myroot.pk -cp root_pw -ca 0 -co
myserver.crt -io myserver.id -ko myserver.pk -kp server_pw -m 124
-sc CA -sst Ontario -sl Waterloo -so MyCompany -sou MyUnit -scn MyServerHost -u
1,3,4,5 -v 10
```

3. createsert を使用して、クライアント証明書を作成します。たとえば、次のコマンドをすべて 1 行に入力して実行します。

```
createcert -b 2048 -c myroot.crt -ck myroot.pk -cp root_pw -ca 0 -co
myclient.crt -io myclient.id -ko myclient.pk -kp client_pw -m 125
-sc CA -sst Ontario -sl Waterloo -so SAP -sou MyUnit -scn MyClientHost -u
1,3,4,5 -v 10
```

MyClientHost は任意の文字列のため、実際のホスト名と一致していない場合があります。

4. データベースサーバを起動し、ID および `identity_password` オプションを追加してデータベースサーバの ID ファイルを指定、および `trusted_certificate` オプションを追加してルート証明書を指定します。たとえば、次のコマンドをすべて 1 行に入力して実行します。

```
dbsrv17 -n MyDBServer MyDatabase.db -x tcpip -ec
"TLS(identity=myserver.id;identity_password=root_pw;trusted_certificate=myroot.crt)"
```

ルート証明書は、クライアントの証明書に署名する証明書です。

5. クライアントアプリケーションの接続をテストします。たとえば、ping ユーティリティ (`dbping`) を使用して、次のコマンドをすべて 1 行に入力して実行します。

```
dbping -d -c
"UID=DBA;PWD=sql;HOST=MyServerHost;DBN=MyDatabase;ENC=TLS(trusted_certificate=myroot.crt;identity=myclient.id;identity_password=client_pw)"
```

ID および `identity_password` オプションでクライアントの ID ファイルを指定し、`trusted_certificate` オプションでデータベースサーバのルート証明書 (クライアントの証明書に署名する証明書) を指定します。

## 結果

クライアント用の信頼できる証明書が作成され、データベースサーバでクライアント証明書を確認することによって、クライアントアプリケーション接続が許可されました。

## 関連情報

[トランスポートレイヤセキュリティ \[1640 ページ\]](#)

[デジタル証明書 \[1644 ページ\]](#)

[証明書作成ユーティリティ \(createcert\) \[1079 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

[Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

### 1.9.3.6 SQL Anywhere クライアント/サーバ通信の暗号化

SQL Anywhere クライアント/サーバ通信は、トランスポートレイヤセキュリティを使用して暗号化できます。

このセクションの内容:

[トランスポートレイヤセキュリティを指定したデータベースサーバ \[1654 ページ\]](#)

トランスポートレイヤセキュリティ (TLS) を使用してデータベースサーバを起動するには、サーバ ID ファイル名と、プライベートキーを保護するパスワードを指定します。

[トランスポートレイヤセキュリティを使用するようにクライアントアプリケーションを設定する \[1655 ページ\]](#)

トランスポートレイヤセキュリティを使用するように SQL Anywhere クライアントアプリケーションを設定することができます。

## 関連情報

[SQL Anywhere Web サービス暗号化 \[1658 ページ\]](#)

### 1.9.3.6.1 トランスポートレイヤセキュリティを指定したデータベースサーバ

トランスポートレイヤセキュリティ (TLS) を使用してデータベースサーバを起動するには、サーバ ID ファイル名と、プライベートキーを保護するパスワードを指定します。

-ec データベースサーバオプションを使用して、ID ファイルとそのパスワードを指定します。共有メモリを経由した暗号化されていない接続を許可するには、-es オプションも指定する必要があります。

dbsrv17 コマンドラインの一部の構文を次に示します。

```
-ec TLS(  
  IDENTITY=identity-file;  
  IDENTITY_PASSWORD=password )  
-x tcpip
```

#### identity-file

サーバ ID ファイルのパスとファイル名を指定します。FIPS 認定の RSA 暗号化を使用している場合は、RSA アルゴリズムを使用して証明書を生成する必要があります。

ID ファイルには、パブリック証明書とプライベートキーが格納されています。自己署名されていない証明書の場合は、その証明書に署名を行うすべての証明書も ID ファイルに格納されています。

#### password

サーバのプライベートキーのパスワードを指定します。このパスワードは、サーバ証明書を作成するときに指定します。

単純難読化を使用してデータベースサーバを起動することもできます。単純難読化を使用すると、パケットスニッファを使用して、クライアントとサーバの間で送信されるネットワークパケットを読み取るのがより困難になります。ただし、データの整合性は保証されず、サーバ検証を行うこともできません。

TCP/IP プロトコルは、-x データベースサーバオプションを使用して指定します。

#### 例

次の例では、-ec データベースサーバオプションを使用して、トランスポートレイヤセキュリティ、サーバ ID ファイル、サーバのプライベートキーを保護するパスワードを指定します。

```
dbsrv17 -ec tls(identity=rsaserver.id;identity_password=test) -x tcpip secure.db
```

設定ファイルとファイル非表示ユーティリティ (dbfhide) を使用して、コマンドラインオプションを非表示にできます。

## 関連情報

[デジタル証明書 \[1644 ページ\]](#)

[トランスポートレイヤセキュリティの設定方法 \[1643 ページ\]](#)

[-fips データベースサーバオプション \[411 ページ\]](#)

[-ec データベースサーバオプション \[402 ページ\]](#)

[-es データベースサーバオプション \[407 ページ\]](#)

[-x データベースサーバオプション \[500 ページ\]](#)

[ファイル非表示ユーティリティ \(dbfhide\) \[1107 ページ\]](#)

[@data データベースサーバオプション \[382 ページ\]](#)

### 1.9.3.6.2 トランスポートレイヤセキュリティを使用するようにクライアントアプリケーションを設定する

トランスポートレイヤセキュリティを使用するように SQL Anywhere クライアントアプリケーションを設定することができます。暗号化接続パラメータセットを使用して、信頼できる証明書、暗号化のタイプ、ネットワークプロトコルを指定します。

このセクションの内容:

[サーバ認証 \[1655 ページ\]](#)

リモートクライアントは、サーバ認証を使用することで、データベースサーバのアイデンティティを確認できます。

[トランスポートレイヤセキュリティを使用してクライアント接続を確立する方法 \[1657 ページ\]](#)

クライアントアプリケーションがトランスポートレイヤセキュリティを使用するように設定するには、接続文字列の中で Encryption (ENC) 接続パラメータを使用します。

## 関連情報

[トランスポートレイヤセキュリティの設定方法 \[1643 ページ\]](#)

### 1.9.3.6.2.1 サーバ認証

リモートクライアントは、サーバ認証を使用することで、データベースサーバのアイデンティティを確認できます。

デジタル署名と証明書フィールドの確認が一緒に機能して、サーバ認証が実現します。

## デジタル署名

データベースサーバ証明書には、データの整合性を維持し、不正侵入を防ぐための、複数のデジタル署名が含まれています。デジタル署名の作成は、次の手順で行われます。

- 証明書で実行されるアルゴリズムが、ユニークな値またはハッシュを生成します。
- 証明書への署名または認証局のプライベートキーを使用して、ハッシュが暗号化されます。
- デジタル署名と呼ばれる暗号化ハッシュが、証明書に埋め込まれます。

デジタル署名は、自己署名、あるいはルート証明書または認証局の署名を受けています。

クライアントアプリケーションがデータベースサーバにアクセスする場合、各クライアントがトランスポートレイヤセキュリティを使用するように設定されていると、サーバはその証明書のコピーをクライアントに送信します。クライアントは、証明書に含まれているサーバのパブリックキーを使用して証明書のデジタル署名を復号化し、証明書の新しいハッシュを算出して、2つの値を比較します。値が一致する場合は、サーバの証明書の整合性が確認されます。

FIPS 認定の RSA 暗号化を使用している場合は、RSA を使用して証明書を生成する必要があります。

## 証明書の検証

グローバル署名証明書を使用する場合、各クライアントは証明書のフィールド値を確認して、同じ認証局が他のクライアント用に署名した証明書を信頼することを避けなければなりません。この問題を解決するには、証明書の識別情報部分のフィールド値をテストするようにクライアントに要求します。認証局は、署名を行ったすべての証明書の識別情報が正確であることを保証する必要があります。

対応するクライアント接続パラメータを使用して、これらのフィールドを確認します。サードパーティの認証局を使用して証明書にグローバル署名している場合は、証明書フィールドを確認してください。

### 組織

組織フィールドは、certificate\_company 暗号化プロトコルオプションに対応します。

### 組織単位

組織単位フィールドは、certificate\_unit 暗号化プロトコルオプションに対応します。

### 通称

通称フィールドは、certificate\_name 暗号化プロトコルオプションに対応します。

## skip\_certificate\_name\_check プロトコルオプションの使用

このブールプロトコルオプションは、TLS または HTTPS 接続をするときに、データベースサーバのホスト名がデータベースサーバの証明書と照合されるかどうかを制御します。このオプションに ON を設定すると、クライアントがデータベースサーバを十分に認証しなくなるため、おすすめしません。このオプションが OFF に指定され、certificate\_name プロトコル、certificate\_company プロトコル、certificate\_unit プロトコルのいずれも設定されていない場合は、データベースサーバのホスト名がデータベースサーバ証明書のホスト名と一致する必要があります。

データベースサーバの IP アドレスがデータベースサーバの証明書でエンコードされている場合、HOST パラメータでホスト名よりも、IP アドレスを指定する必要があります。

## trusted\_certificates プロトコルオプションの使用

クライアントは、trusted\_certificates 暗号化プロトコルオプションを使用して、信頼できるデータベースサーバ証明書を指定します。信頼できる証明書は、サーバの自己署名証明書、パブリックルート証明書、民間認証局に属する証明書のいずれかです。trusted\_certificates プロトコルオプションが指定されていない場合、データベースサーバ証明書は OS 証明書ストアに対してチェックされます。

### 関連情報

[自己署名ルート証明書 \[1646 ページ\]](#)

[グローバル署名証明書 \[1649 ページ\]](#)

[デジタル証明書 \[1644 ページ\]](#)

[証明書チェーン \[1647 ページ\]](#)

[certificate\\_company プロトコルオプション \(クライアント側のみ\) \[186 ページ\]](#)

[certificate\\_unit プロトコルオプション \(クライアント側のみ\) \[188 ページ\]](#)

[certificate\\_name プロトコルオプション \(クライアント側のみ\) \[187 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

[trusted\\_certificates プロトコルオプション \[234 ページ\]](#)

### 1.9.3.6.2.2 トランスポートレイヤセキュリティを使用してクライアント接続を確立する方法

クライアントアプリケーションがトランスポートレイヤセキュリティを使用するように設定するには、接続文字列の中で Encryption (ENC) 接続パラメータを使用します。

接続文字列の形式は次のとおりです。

```
Encryption=TLS (
  [ FIPS={ y | n }; ]
  [ TRUSTED_CERTIFICATES=public-certificate ]
  [ CERTIFICATE_COMPANY=organization; ]
  [ CERTIFICATE_NAME=common-name; ]
  [ CERTIFICATE_UNIT=organization-unit ] )
```

#### public-certificate

信頼できる証明書を 1 つ以上含むファイルのパスとファイル名を指定します。FIPS 認定の RSA 暗号化を使用している場合は、RSA を使用して証明書を生成する必要があります。

#### organization

証明書に記されている組織フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

#### common-name

証明書に記されている通称フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

## organization-unit

証明書に記されている組織単位フィールドがこの値と一致する場合にだけ、クライアントでサーバ証明書を受け入れるようにします。

キーワード `TRUSTED_CERTIFICATE` も受け入れられます。

### 例

次の例では、`trusted_certificates` 暗号化接続パラメータを使用して、証明書 `public_cert.crt` を指定します。

```
"UID=DBA;PWD=passwd;Host=myhost;Server=myserver;  
ENC=tls(trusted_certificates=public_cert.crt)"
```

次の例では、`trusted_certificates` 暗号化接続パラメータを使用して証明書 `public_cert.crt` を指定し、`certificate_unit` および `certificate_name` 暗号化接続パラメータを使用して証明書フィールドを確認します。

```
"UID=DBA;PWD=passwd;Host=myhost;Server=myserver;  
ENC=tls(trusted_certificates=public_cert.crt;  
certificate_unit=test_unit;certificate_name=my_certificate)"
```

## 関連情報

[デジタル証明書 \[1644 ページ\]](#)

[trusted\\_certificates プロトコルオプション \[234 ページ\]](#)

[certificate\\_company プロトコルオプション \(クライアント側のみ\) \[186 ページ\]](#)

[certificate\\_name プロトコルオプション \(クライアント側のみ\) \[187 ページ\]](#)

[certificate\\_unit プロトコルオプション \(クライアント側のみ\) \[188 ページ\]](#)

[Encryption \(ENC\) 接続パラメータ \[77 ページ\]](#)

## 1.9.3.7 SQL Anywhere Web サービス暗号化

SQL Anywhere Web サーバは、TLS バージョン 1.0 以降を使用する HTTPS 接続をサポートします。

SQL Anywhere Web サービス用にトランスポートレイヤセキュリティを設定するには、次の手順を実行します。

### デジタル証明書を取得する

データベースサーバ証明書ファイルと ID ファイルが必要です。証明書 (認証局の証明書の場合もあります) は、ブラウザまたは Web クライアントに配布します。サーバ ID ファイルは、SQL Anywhere Web サーバに安全に保存されます。

### トランスポートレイヤセキュリティを指定して Web サーバを起動する

`-xs` データベースサーバオプションを使用して、HTTPS、サーバ ID ファイル、プライベートキーを保護するパスワードを指定します。

### Web クライアントを設定する

ブラウザまたは他の Web クライアントが証明書を信頼するように設定します。信頼できる証明書は、自己署名証明書、ルート証明書、または認証機関による証明書です。

## 関連情報

[デジタル証明書 \[1644 ページ\]](#)

[-fips データベースサーバオプション \[411 ページ\]](#)

[-xs データベースサーバオプション \[507 ページ\]](#)

[identity プロトコルオプション \[201 ページ\]](#)

[identity\\_password プロトコルオプション \[202 ページ\]](#)

## 1.10 高可用性と読み込み専用のスケールアウトシステム

データベースミラーリングと Veritas Cluster Server がサポートされており、高可用性が実現しています。

これらの両方の構成によって、現在データベースを実行しているデータベースサーバが使用できなくなった場合に、データベースを実行できる代替データベースサーバが提供されます。

負荷を分散させたい場合は、読み込み専用スケールアウトを使用することが適切な解決法であるかどうか検討してください。

また、Web アプリケーションをホストするために高可用性かつ読み込み専用のスケールアウトを使用することもできます。

このセクションの内容:

[データベースミラーリング \[1660 ページ\]](#)

データベースミラーリングとは、異なるコンピュータ上で実行されている 3 つのデータベースサーバが連携してデータベースやトランザクションログファイルのコピーを保持する構成です。

[SQL Anywhere Veritas Cluster Server エージェント \[1722 ページ\]](#)

クラスタとは、一連のアプリケーションを実行するために連携して動くコンピュータ (ノードと呼ばれる) によるグループです。

[読み込み専用スケールアウト \[1733 ページ\]](#)

読み込み専用スケールアウトによって、データベースへの読み込み専用アクセスを必要とするレポートなどの操作の負荷を軽減できます。

## 関連情報

[読み込み専用スケールアウト \[1733 ページ\]](#)

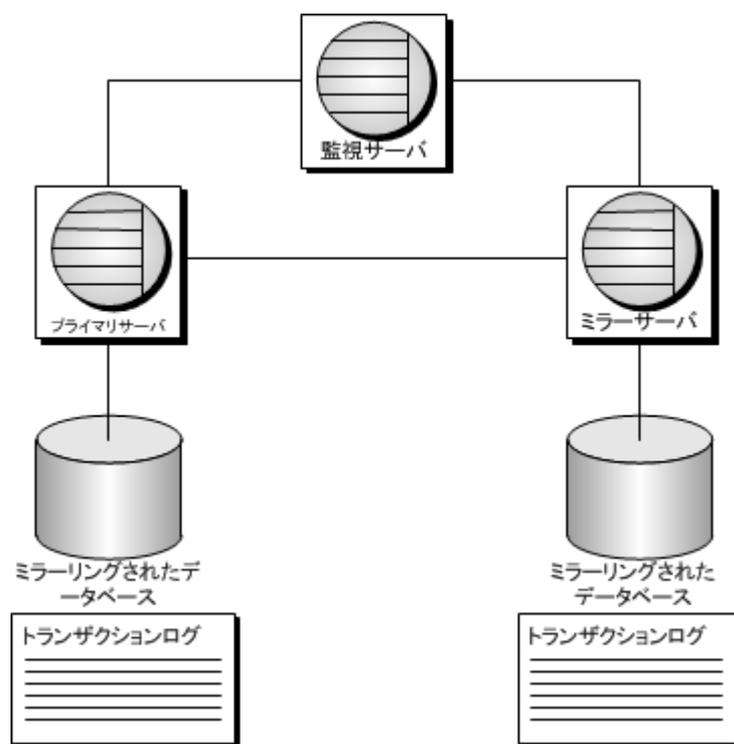
[Relay サーバおよび Outbound Enabler を使用する、Web サービスの高可用性とスケールアウトソリューション \[1736 ページ\]](#)

## 1.10.1 データベースミラーリング

データベースミラーリングとは、異なるコンピュータ上で実行されている3つのデータベースサーバが連携してデータベースやトランザクションログファイルのコピーを保持する構成です。

データベースミラーリングシステムでは、3つの独立した物理コンピュータ上で3つのデータベースサーバが動作する設定が想定されているため、継続的にシステムを動作させることができます。データベースミラーリングでは、コンピュータの1つが一時停止した場合に対応していません。

プライマリサーバとミラーサーバは、それぞれデータベースファイルおよびトランザクションログファイルのコピーを保持します。プライマリサーバにはデータベースの書き込み専用のコピーが保持され、それに対してミラーサーバにはデータベースの読み取り専用のコピーが保持されます。監視サーバと呼ばれる3番目のサーバは、他の2つのサーバのどちらがデータベースの所有権を持つのかを判断する必要があるときに使用されます。データベースの所有権は、データベースの読み書きコピーを保持するサーバを参照します。監視サーバはデータベースのコピーを保持しません。この3種類のデータベースサーバ（プライマリ、ミラー、監視の各サーバ）による構成はミラーリングシステムと呼ばれ、プライマリサーバとミラーサーバは併せてパートナーサーバと呼ばれます。



クライアントがデータベースにアクセスする場合は、プライマリサーバに接続します。データベースに対するあらゆる変更は、プライマリサーバ上のトランザクションログファイルに記録されます。変更がコミットされると、トランザクションログページがミラーサーバに送信され、データベースのミラーコピーに適用されます。サーバがミラーサーバとして動作している間は、そのミラーサーバにあるデータベースのコピーには読み込み専用モードでのみアクセスできます。

データベースミラーリングは別途ライセンスが必要なコンポーネントです。

このセクションの内容:

[監視サーバ \[1661 ページ\]](#)

監視サーバは、プライマリサーバ選定におけるサーバ間の競合を解消します。

#### [データベースミラーリングでのクォーラム \[1662 ページ\]](#)

サーバがプライマリサーバのロールを引き受けるには、クォーラムが必要です。これは、他のサーバの少なくともどちらか 1 つがデータベースの所有に合意していることを意味します。

#### [データベースミラーリングと Mobile Link \[1663 ページ\]](#)

SQL Anywhere 統合データベースを複数の SQL Anywhere リモートデータベースとともに使用して、高可用性環境内で Mobile Link 同期を実行するために設定します。

#### [データベースミラーリングでのアプリケーション開発の考慮事項 \[1664 ページ\]](#)

データベースミラーリングを使用する場合、アプリケーションは、ほとんどのケースでミラーリングされていないデータベースに接続している場合と同じように実行できます。

#### [データベースミラーリングシステムの設定 \[1674 ページ\]](#)

プライマリサーバ、ミラーサーバ、監視サーバでデータベースおよびトランザクションログファイルのコピーを保持するように設定することで、ミラーリングシステムを設定します。

#### [データベースミラーリングシステムの保守 \[1680 ページ\]](#)

データベースモニタリングシステムの保守は重要です。

#### [トラブルシューティング: データベースミラーリングシステム \[1691 ページ\]](#)

ミラーリングシステムのトラブルシューティングに使用できる拡張データベースプロパティは 3 つあります。

#### [チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

データベースミラーリングシステムを作成して、フェイルオーバーに対処します。

#### [チュートリアル: 監視サーバを共有している複数データベースでデータベースミラーリングシステムを作成する \[1711 ページ\]](#)

プライマリサーバとミラーサーバのそれぞれが、ミラーリングシステムに参加している 3 つの個別データベースを実行するミラーリング構成を作成します。

#### [チュートリアル: 監視サーバの移動 \[1718 ページ\]](#)

新しいサーバを作成し、新しい監視サーバを使用するように監視サーバ定義を変更して、ミラーリングシステムを停止することなく監視サーバを移動します。

#### [チュートリアル: パートナサーバの移動 \[1720 ページ\]](#)

システムを停止しないでデータベースミラーリングシステム内のサーバを別のサーバに移動し、現在のミラーサーバのミラー定義を削除し、新しいサーバを作成し、新しいサーバをミラーリングシステムに追加します。

## 関連情報

[ミラーリングシステムでのデータベースへの接続 \[1665 ページ\]](#)

### 1.10.1.1 監視サーバ

監視サーバは、プライマリサーバ選定におけるサーバ間の競合を解消します。

監視サーバがないと、サーバ B が利用できないときにサーバ A がミラーリングシステムで起動された場合、サーバ A はどのデータベースファイルのコピーが最新のものを判断できません。最新ではないファイルを使用してデータベースを起動する

と、もう一方のデータベースのコピーに適用されてコミットされたトランザクションが失われます。また、2つのパートナーサーバが接続を再確立した後では、もう一方のデータベースのコピーをミラーリングに使用できなくなります。

監視サーバは、起動時の競合を解消するほか、2つのサーバが、サーバ間の通信リンクが壊れたままで引き続き実行されている場合についても対応します。監視サーバがなかった場合、どちらのサーバもデータベースの所有権を取得しようとします。この場合も、トランザクションが失われ、それぞれ互換性のないデータベースになってしまいます。監視サーバがあれば、プライマリサーバはデータベースの所有権が存続することを証明して、クライアントにアクセスを提供し続けることができます。プライマリサーバがミラーと監視サーバのどちらとも通信できなくなった場合は、シャットダウンして、どちらかに接続可能になるまで待機する必要があります。

監視サーバとして実行されるサーバは、1つ以上のミラーシステムの監視として機能できます。また、他のデータベースのデータベースサーバとしても機能できます。

## 関連情報

[監視サーバとしてコピーノードを使用する \[1678 ページ\]](#)

### 1.10.1.2 データベースミラーリングでのクォーラム

サーバがプライマリサーバのロールを引き受けるには、クォーラムが必要です。これは、他のサーバの少なくともどちらか1つがデータベースの所有に合意していることを意味します。

ミラーサーバが使用できなくなった場合でも、プライマリサーバと監視サーバが接続されていれば、プライマリサーバは引き続きデータベースへのアクセスを提供します。プライマリサーバがクォーラムを失った場合、プライマリサーバはデータベースへのアクセスを許可できなくなります。その時点で、プライマリサーバはミラーリングされたデータベースを停止し、再起動を試み、データベースが利用可能になる前にクォーラムを再取得するまで待機します。

## データベースミラーリングシステムの起動時に発生すること

データベースミラーリングシステムが起動されると、データベースサーバは起動プロセスを開始してクォーラムを形成し、クライアント接続を受け付けます。次の手順は、このプロセスのイベントの標準的な順序を説明しています。サーバ1とサーバ2はいずれも停止しているパートナー (サーバ2はミラーで先に停止され、その後サーバ1が停止された) で、監視サーバが実行されていると仮定します。

1. 監視サーバはサーバ1とサーバ2からの接続を待機します。
2. サーバ1が起動し、サーバ2および監視サーバへの接続を試行します。
3. サーバ1が監視サーバに接続します。
4. サーバ1がサーバ2への接続に失敗します。これは、サーバ2が実行されていないためです。
5. サーバ1は監視サーバとネゴシエートし、サーバ1がプライマリサーバまたはミラーサーバであるかどうかを判断します。
6. 監視サーバとサーバ1が、サーバ1がプライマリサーバのロールを想定すべきだと同意したときに、クォーラムが形成されます。監視サーバおよびサーバ1の状態ファイルは、サーバ1がパートナーサーバの停止前に最新のプライマリサーバであったことを示しています。
7. サーバ1は接続の受け付けを開始します。

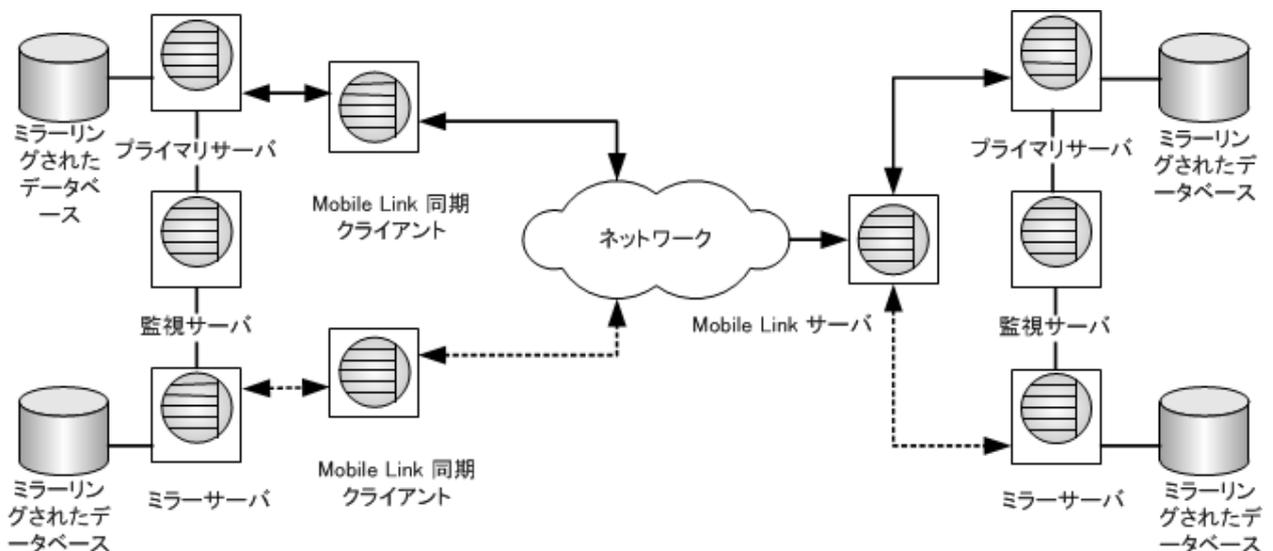
8. サーバ 2 が起動し、サーバ 1 および監視サーバへの接続を試行します。
9. サーバ 2 が監視サーバとサーバ 1 に接続します。
10. サーバ 2 はサーバ 1 および監視サーバとネゴシエートし、サーバ 2 がプライマリサーバまたはミラーサーバであるかどうかを判断します。
11. 3 つすべてのサーバは、サーバ 2 がミラーであることに同意します。
12. サーバ 2 が起動し、読み取り専用接続を受け入れます。
13. サーバ 2 が起動し、サーバ 1 から受け取ったログファイルのページを適用します。

### 1.10.1.3 データベースミラーリングと Mobile Link

SQL Anywhere 統合データベースを複数の SQL Anywhere リモートデータベースとともに使用して、高可用性環境内で Mobile Link 同期を実行するために設定します。

Mobile Link 同期システムは、高可用性環境内で統合データベースを実行するように設定することも、リモートデータベースを高可用性環境内で実行するように設定することも、またはその両方に設定することもできます。

次の図は、統合データベースとリモートデータベースの両方がミラーリングされている環境で実行される Mobile Link 同期を示しています。



### SQL Anywhere 統合データベースの高可用性

プライマリサーバは、データベースファイルとトランザクションログのアクティブなコピーを管理します。このサーバは、アクティブな接続を受け入れて、データベーストランザクションを処理します。すべてのトランザクションはプライマリサーバによってミラーサーバに送信されます。

ミラーサーバは、プライマリサーバに適用されたすべてのトランザクションを、プライマリサーバから直接受信します。フェイルオーバーが発生すると、ミラーサーバがプライマリサーバになります。

Mobile Link サーバは、プライマリサーバとの ODBC 接続を確立し、Mobile Link クライアントからのアップロードストリームを処理します。また、クライアントに送信するダウンロードストリームを生成します。Mobile Link サーバは、プライマリとして動作するサーバに常に接続し、ミラーサーバには接続しません。

## SQL Anywhere リモートデータベースの高可用性

データの同期は、Mobile Link クライアントがリモートデータベースと Mobile Link サーバとの接続を確立したときに開始されます。Mobile Link クライアントは、アクティブなトランザクションログとオフライントランザクションログをスキャンする必要があります。ただし、高可用性環境では、どのデータベースが現在プライマリサーバとして動作しているかや、アクティブとオフラインのトランザクションログがどこにあるかが常にわかっているとかがりません。オフライントランザクションログがデータベースに関連付けられている場合は、Mobile Link クライアントを起動するときに OfflineDirectory 拡張オプション (たとえば、`-e dir=path-name`) を指定する必要があります。

Mobile Link クライアントは、リモートデータベースのトランザクションログに基づいて、Mobile Link サーバに送信するためのアップロードストリームを生成します。その後クライアントはダウンロードストリームを処理して適用します。

### プライマリサーバ障害からのリカバリ

高可用性データ同期システムに単一障害点ができないようにするために、複数の Mobile Link サーバと負荷分散装置を使用します。これにより、データ同期環境の各コンポーネントが冗長化され、環境内でハードウェアやソフトウェアの障害が発生した場合でもシステムが機能します。

同様に、リモートデータベースでの単一障害点を防止するには、xp\_cmdshell システムプロシージャのような外部プロシージャを Mobile Link クライアントの呼び出しに使用する実装を、プライマリデータベースとミラーデータベースの両方に格納されているスケジュールされたイベントによってトリガできます。

### 関連情報

[高可用性と読み込み専用のスケールアウトシステム \[1659 ページ\]](#)

## 1.10.1.4 データベースミラーリングでのアプリケーション開発の考慮事項

データベースミラーリングを使用する場合、アプリケーションは、ほとんどのケースでミラーリングされていないデータベースに接続している場合と同じように実行できます。

ただし、データベースミラーリングと使用するアプリケーションを開発するときに考慮すべき事項がいくつかあります。

このセクションの内容:

[ミラーリングシステムでのデータベースへの接続 \[1665 ページ\]](#)

Host 接続パラメータで両方のパートナーのアドレスを指定し、ServerName 接続パラメータで各サーバの代替名を指定することで、プライマリサーバまたはミラーサーバのいずれかでデータベースに接続します。

#### データベースミラーリングシステムの要件と制限 [1669 ページ]

データベースミラーリングにハードウェアやソフトウェアに関する特別な要件はありません。また、各データベースサーバは、地理的に離れたロケーションで実行されていてかまいません。

#### データベースミラーリングシステムでのパフォーマンスの考慮事項 [1670 ページ]

データベースミラーリングシステムには、パフォーマンスの考慮事項がいくつかあります。

#### データベースミラーリングモード [1671 ページ]

トランザクションをミラーサーバにいつ、どのように記録するのかを制御する実行モードには 3 種類あります。

## 1.10.1.4.1 ミラーリングシステムでのデータベースへの接続

Host 接続パラメータで両方のパートナーのアドレスを指定し、ServerName 接続パラメータで各サーバの代替名を指定することで、プライマリサーバまたはミラーサーバのいずれかでデータベースに接続します。

### 前提条件

ミラーリングサーバパートナーが実行中である必要があります。

### コンテキスト

ミラーリングシステムでは、どのデータベースサーバがプライマリサーバで、どのデータベースサーバがミラーサーバとして動作しているかが不明なこともあります。

データベースに再接続できるクライアントを作成します (たとえば、フェイルオーバーが発生した場合、ユーザがアプリケーションをシャットダウンして再起動する必要が生じることがあります)。

### 手順

ミラーリングされたデータベースを実行しているプライマリサーバまたはミラーサーバに接続するには、接続文字列に次の接続パラメータを含める必要があります。

#### サーバ

プライマリサーバに接続するには

次のオプションのうちの 1 つを選択してください。

オプション	アクション
ServerName 接続パラメータ	データベースミラーリングシステムのプライマリデータベースサーバの代替サーバ名を指定します。これは、CREATE MIRROR SERVER <code>primary_alterate_server_name</code> AS PRIMARY ... 文によって定義された名前です。
NodeType 接続パラメータ	接続文字列で、NodeType (Node) 接続パラメータを PRIMARY に設定します。

ミラーサーバに接続するには  
次のオプションのうちの 1 つを選択してください。

オプション	アクション
ServerName 接続パラメータ	データベースミラーリングシステムのミラーデータベースサーバの代替サーバ名を指定します。これは、CREATE MIRROR SERVER <code>mirror_alterate_server_name</code> AS MIRROR ... 文によって定義された名前です。
NodeType 接続パラメータ	接続文字列で、NodeType (Node) 接続パラメータを MIRROR に設定します。

## HOST

両方のパートナーのアドレスとポートを指定し、どちらのパートナーが現在プライマリであるかということに関係なく、接続が正常に行われるようにします。パートナーそれぞれのホストおよびポート情報は、CREATE MIRROR SERVER `partner_server_name` AS PARTNER `connection_string='SERVER=partner_server_name;host=host_name:port_number` 文の接続文字列で定義されます。

RetryConnectionTimeout 接続パラメータを指定して、クライアントがいずれかのデータベースサーバに接続試行を繰り返す時間を制御できます。

## 結果

クライアントが指定したサーバに接続します。

### 例

たとえば、プライマリデータベースサーバに指定された名前が `myprimary` の場合、クライアントは接続文字列で接続パラメータ `Server=myprimary` を指定します。

```
...UID=user12;PWD=x92H4pY;Server=myprimary;HOST=myhost1:6871,myhost2:6872...
```

このセクションの内容:

[ミラーデータベースで実行されたクエリ \[1667 ページ\]](#)

データベースミラーリングシステムでは、読み込み専用の接続を使用して、ミラーサーバ上で実行されているデータベースにアクセスできます。

[データベースミラーリングまたは読み取り専用スケールアウトシステムでのイベントの実行 \[1668 ページ\]](#)

データベースミラーリングまたは読み取り専用スケールアウトシステムでイベントが実行できます。

## 関連情報

[読み込み専用スケールアウトシステムにおけるデータベースへの接続 \[1740 ページ\]](#)

[RetryConnectionTimeout \(RetryConnTO\) 接続パラメータ \[104 ページ\]](#)

[NodeType \(NODE\) 接続パラメータ \[96 ページ\]](#)

### 1.10.1.4.1.1 ミラーデータベースで実行されたクエリ

データベースミラーリングシステムでは、読み込み専用の接続を使用して、ミラーサーバ上で実行されているデータベースにアクセスできます。

この機能は、このデータベースへの読み込み専用アクセスを必要とするレポートなどの操作の負荷を軽減したい場合に役立ちます。

ミラーデータベースを変更しようとするエラーが発生します。これは、`-r` オプションを使用して読み込み専用でデータベースを起動した場合と同じ動作です。テンポラリテーブルで操作を実行できます。

ミラーデータベースに対して実行されたクエリは、指定された独立性レベルに応じてロックを設定できます。プライマリサーバからの操作の適用がロックによって妨害される場合は、ロックを保持している接続のトランザクションがロールバックされ、その接続用の開いたカーソルがある場合は閉じられます。独立性レベル 0 で実行されているアプリケーションは、ローロックを追加しませんが、スキーマロックは取得します。プライマリサーバからの操作の適用がスキーマロックによって妨害される場合は、ミラーデータベース上のトランザクションのロールバックのみが行われます。

データベースの一貫したビューが必要な (そのため、独立性レベル 0 を使用できない) アプリケーションでは、スナップショットアイソレーションの使用を検討する必要があります。そのためには、`allow_snapshot_isolation` オプションを On に設定します。このオプションはプライマリサーバとミラーサーバの両方に対して有効なので、スナップショットアイソレーションに関連するコストを考慮する必要があります。

ミラーデータベースへの接続は、プライマリサーバに対するトランザクションの影響を受けます。これらの操作は、その後ミラーサーバによって処理され適用されるからです。プライマリサーバ上の更新がコミットされてから、その更新がミラーサーバ上で使用可能になるまでには、わずかな遅延があります。通常この遅延は短時間ですが、ミラーサーバ上で実行されるデータベースにアクセスするときは、このことを念頭においてください。

フェイルオーバーが発生してミラーサーバがプライマリサーバになっても、ミラーデータベースへの接続は維持されます。フェイルオーバー後は、接続によってデータベースに変更を加えることができます。接続先のデータベースが更新可能かどうかを確認するには、`ReadOnly` データベースプロパティの値を問い合わせます。

```
SELECT DB_PROPERTY( 'ReadOnly' );
```

コピーノードまたはミラーデータベースでの接続は、トランザクションログの適用を阻害しているとき、一部の場​​合削除されるかキャンセルされます。たとえば、トランザクションログが変更または削除しようとしているイベントが接続で使用されている場合は、トランザクションログが適用されないようにブロックしている接続が削除され、サーバコンソールにメッセージが出力されま​​す。

## 関連情報

[データベースミラーリングまたは読み取り専用スケールアウトシステムでのイベントの実行 \[1668 ページ\]](#)

[allow\\_snapshot\\_isolation オプション \[643 ページ\]](#)

### 1.10.1.4.1.2 データベースミラーリングまたは読み取り専用スケールアウトシステムでのイベントの実行

データベースミラーリングまたは読み取り専用スケールアウトシステムでイベントが実行できます。

イベントは、プライマリサーバまたはルートサーバのいずれかでのみ実行可能か、プライマリサーバ、ミラーサーバおよびコピーノードで実行可能です。

デフォルトでは、イベントはミラーデータベースで起動されず、プライマリサーバからミラーサーバへのフェイルオーバーが行われた後でのみイベントが起動されます。

イベントを作成または変更するときに、イベントの実行場所を指定します。CREATE EVENT 文または ALTER EVENT 文の FOR 句を使用して、イベントを起動できる場所を指定します。

## DatabaseStart イベントタイプ

デフォルトでは、DatabaseStart イベントタイプを使用するイベントは、サーバがデータベースのプライマリサーバになったときのみ実行されます。ただし、イベントがプライマリサーバ、ミラーサーバ、およびコピーノードで実行できるように指定してある場合、そのイベントはデータベースの起動時に実行されます。ミラーリングシステムでデータベースが起動したときにこのイベントが実行されなかった場合 (たとえば、イベントが作成される前にデータベースが実行されていた場合など)、そのイベントはフェイルオーバーのときに実行できます。

たとえば、データベースミラーリングシステムを起動し、FOR ALL 句を使用して DatabaseStart イベントを作成して、プライマリサーバを停止すると、フェイルオーバーが発生します。この例では、新しいプライマリサーバでイベントが実行されます。DatabaseStart イベントは後続のフェイルオーバー中には実行されません。

## 関連情報

[システムイベント \[954 ページ\]](#)

[Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

## 1.10.1.4.2 データベースミラーリングシステムの要件と制限

データベースミラーリングにハードウェアやソフトウェアに関する特別な要件はありません。また、各データベースサーバは、地理的に離れたロケーションで実行されていてかまいません。

データベースミラーリングシステムを構成するデータベースサーバでは、ミラーリングされたデータベースもされていないデータベースも実行できます。また、監視サーバは、複数のデータベースミラーリングシステムの監視サーバになることができます。

データベースサーバが1つ以上のミラーリングされたデータベースを実行している場合、ミラーリングされるデータベースは、それぞれ異なるディレクトリ内にある必要があります。

データベースミラーリングの使用時には、次の要件と制限が適用されます。

**ネットワークデータベースサーバが必要**

ミラーリングにはデータベースサーバ間のネットワーク通信が必要なため、ネットワークデータベースサーバ (dbsrv17) を使用する必要があります。パーソナルサーバは使用できません。

データベースミラーリングシステム内のすべてのデータベースサーバで、同じマイナリリースを使用する必要がありますが、異なるサポートパッケージを実行できます。

**TCP/IP が必要**

ミラーリングサーバ間で許可されるのは TCP/IP 接続だけです。

**トランザクションログの制限**

トランザクションがトランザクションの損失につながる可能性があるため、データベースミラーリングを使用しているときは、プライマリでトランザクションログをトランケートできません。必要であれば何度でもトランザクションログの名前を変更できます。

**Web サーバ**

SQL Anywhere を Web サーバとしてミラーリングシステムで使用する場合、Web 要求の URL を指定しても、要求が現在のプライマリサーバに送信されることは保証されません。いずれかのサーバコンピュータが URL で指定されている場合、そのサーバが停止すると、要求はタイムアウトします。

定期的なバックアップが引き続き必要です

データベースミラーリングは、バックアップとリカバリのプランの代用ではありません。データベースのバックアップとリカバリの方式を実装します。

**DLL の制限**

オブジェクトがコピーノードまたはミラーサーバで使用可能になっている間は、プライマリデータベースで実行する DDL 文を最小数に留めます。システムオブジェクトを削除または変更すると、接続でそのオブジェクトを使用している場合、ミラーまたはコピーノードへのユーザ接続が削除されることがあります。

**フェイルオーバーとスケジュールされたイベント**

スケジュールされたイベントのあるデータベースでフェイルオーバーが発生した場合、フェイルオーバーはイベントの開始前に完了する必要があります。完了しない場合、イベントは次のスケジュール時刻まで実行されません。ミラーサーバがプライマリサーバのロールに切り替わろうとし、切り替えがまだ完了していない場合、スケジュールされたイベントは実行されず、次のスケジュール時刻に実行されます。イベントの実行中にプライマリサーバがミラーサーバと監視サーバへの接続を失った場合、イベント接続と他のすべての接続は切断されます。ミラーサーバがプライマリサーバのロールを引き受けた後で実行されるようにイベントがスケジュールされている場合、イベントは新しいプライマリサーバで実行されます。

## 関連情報

[データベースミラーリングシステムでのトランザクションログの管理 \[1678 ページ\]](#)

[データベースミラーリングシステムでのバックアップ \[1681 ページ\]](#)

[データベースのバックアップとリカバリ \[894 ページ\]](#)

[データベースミラーリングシステムでのトランザクションログの管理 \[1678 ページ\]](#)

[ミラーデータベースで実行されたクエリ \[1667 ページ\]](#)

[Relay サーバおよび Outbound Enabler を使用する、Web サービスの高可用性とスケールアウトソリューション \[1736 ページ\]](#)

[トラブルシューティング: データベースミラーリングシステムイベントを使用してフェイルオーバー発生時に通知を送信する \[1683 ページ\]](#)

### 1.10.1.4.3 データベースミラーリングシステムでのパフォーマンスの考慮事項

データベースミラーリングシステムには、パフォーマンスの考慮事項がいくつかあります。

- プライマリサーバおよびミラーサーバを実行しているコンピュータは、同様のハードウェア（プロセッサ、ディスク、メモリなど）で設定する必要があります。どちらかのコンピュータで実行されているデータベースサーバは、常にミラーリング対象のデータベースのプライマリサーバとして動作する可能性があります。ミラーサーバの使用率は通常低く、プライマリでの更新アクティビティやミラーサーバへの読み取り専用接続によって発生する負荷により左右されます。
- ミラーサーバ上でのトランザクションは、適用されているプライマリサーバ上のトランザクションより処理が遅れる場合があります。ミラーサーバでのラグタイムが長くなると、ミラーサーバへの接続が読み取り専用となり、より古い情報にアクセスすることになります。また、ミラーサーバが再起動されるか、プライマリサーバを引き継ぐ必要がある場合には、リカバリ時間が長くなる場合があります。ラグタイムミラーオプションを使用して、プライマリサーバとミラーサーバのラグタイムを許容できる範囲に制御します。ラグタイムが設定した値に近づくと、プライマリサーバはトランザクションのレートを減少させます。ラグタイムが設定値を超えると、遅れがなくなるまでトランザクションログに警告が書き込まれます。
- データベースを更新するトランザクションのパフォーマンスは、トランザクションのサイズとコミットの頻度の影響を受けます。トランザクションのサイズまたはコミットの頻度のいずれかを増やすと、プライマリサーバがミラーサーバとやり取りをする回数が多くなり、それによってプライマリサーバが遅延し、ユーザに遅延を通知する可能性が高くなります。
- プライマリサーバに対するクエリのパフォーマンスは、通常はミラーリングの影響を受けません。アプリケーションが更新よりも多くの読み込みを行い、かつ監査が有効になっていない場合は、プライマリサーバのパフォーマンスでの変化を経験することはほとんどありません。しかし、データベース監査が有効になっている場合、監査がトランザクションログに追加の情報を書き込むため、パフォーマンスの変化を経験する可能性があります。
- クライアントアプリケーションのパフォーマンスは、ミラーリングシステムのサーバ間でのネットワーク接続によって、特に、サーバが地理的に異なる場所にあるときに、影響を受ける可能性があります。サーバ間の低速のネットワーク接続は、クライアントアプリケーションのパフォーマンスを低下させる可能性があります。
- 非同期モードまたは非同期フルページモードのいずれかで実行しているミラーリングシステムは、プライマリサーバに対して同期モードで動作しているときよりも待機する機会をより少なくするように要求します。しかし、非同期モードまたは非同期フルページモードであっても、プライマリサーバからミラーサーバに送信されるメッセージのうちのいくつかは、ミラーからの受信確認を待機するので、これが遅延を引き起こすこともあります。さらに、パケットをミラーに送信するときには少量のオーバーヘッドも存在します。非同期モードまたは非同期フルページモードは、トランザクションが失われる可能性があるため、おすすめてできません。
- ミラーサーバがデータベースの所有権を引き継ぐ場合は、不完全なトランザクションをロールバックする必要があるが、トランザクションが長いほど、ロールバックに時間がかかります。フェイルオーバー用のリカバリ速度は、クライアント数とロ

ールバックの必要があるトランザクションの長さに影響されます。リカバリ速度が問題である場合は、アプリケーションを設定して、可能な場合に短いトランザクションを使用できます。

## 関連情報

[データベースミラーリングモード \[1671 ページ\]](#)

### 1.10.1.4.4 データベースミラーリングモード

トランザクションをミラーサーバにいつ、どのように記録するのかを制御する実行モードには 3 種類あります。

- 同期 (デフォルト)
- 非同期
- 非同期フルページ

ほとんどすべての場合、最も信頼性が高いため、デフォルトのモードである同期モードを推奨します。

各モードは、トランザクションがミラーサーバで記録されるタイミングと方法を制御します。設定には、synchronization\_mode オプションの値を設定する SET MIRROR OPTION 文を使用します。

コピーノードへのログページの送信は、選択されたモードに関係なく常に非同期で行われます。

データベースミラーリングシステムに同期モードを選択するとき、フェイルオーバー発生時にプライマリデータベースへの変更のパフォーマンスが重要なのか、トランザクションの損失回避が重要なのかを決定する必要があります。

データベースミラーリングのモードは、MirrorMode データベースプロパティの値を問い合わせることによって確認できます。

```
SELECT DB_PROPERTY( 'MirrorMode' );
```

#### 同期モード (デフォルト)

同期モードでは、コミットされたトランザクションはミラーサーバに記録されることが保証されます。プライマリサーバで障害が発生した場合、ミラーサーバが引き継いだときに、コミットされたトランザクションが失われることはありません。このモードでは、プライマリサーバはトランザクションがコミットされるとトランザクションログページをミラーに送信します。ミラーサーバは、送信されたページをトランザクションログのコピーに書き込むと、転送の受信確認を行います。プライマリサーバは、受信確認を受信してからアプリケーションに応答します。

同期モードを使用することでトランザクションの安全性が保証されます。これは、パートナーサーバが同期された状態になり、プライマリは、ミラーに送信した変更の受信確認を待ってから、処理を続行するからです。

## 非同期モードと非同期フルページモード (非推奨)

非同期および非同期フルページモードは同期モードより高速ですが、信頼性は低くなります。デフォルトでは、プライマリサーバが予期せず停止した場合、ミラーサーバは自動的にプライマリを引き継ぎません。

### 非同期モード

非同期モードでは、コミットされたトランザクションがミラーサーバに記録されることは保証されません。このモードでは、プライマリサーバはトランザクションがコミットされるとトランザクションログページをミラーに送信します。この時点で、プライマリはミラーからの受信確認を待たずに、COMMIT の完了をアプリケーションに応答します。このため、プライマリサーバで障害が発生した場合、ミラーサーバが引き継いだときにコミット済みのトランザクションが一部失われる可能性があります。

### 非同期フルページモード

非同期フルページ (または page) モードでは、ページは COMMIT の実行時にはなく、ページが満杯になったときに送信されます。この設定により、2 つのデータベースサーバ間のトラフィック量を減らし、プライマリサーバのパフォーマンスを向上します。現在のログページは、pagetimeout パラメータに指定した秒数を過ぎてもミラーに送信されなかった場合に、ページが満杯でなくても送信されます。デフォルトの値は 5 秒です。このモードを使用すると、プライマリサーバがダウンし、ミラーサーバがデータベースの所有権を引き継ぐ場合に、コミットされたトランザクションが失われる可能性がある状態に置かれている時間を制限できます。非同期フルページモードは非同期操作の一種であるため、プライマリサーバはミラーからの受信確認を待機しません。

非同期モードおよび非同期フルページモードでは、プライマリサーバに適用されたコミット済みトランザクションがミラーサーバにすべて適用されているとは限らないので、プライマリサーバからミラーサーバへのフェイルオーバーは自動ではありません。このため、非同期のどちらかのモードを使用する場合は、ミラーサーバはデフォルトで、プライマリに障害が発生してもデータベースの所有権を取得できません。この場合に (トランザクションが失われる可能性があったとしても) 自動フェイルオーバーを使用したい場合は、SET MIRROR OPTION 文を使用して auto\_failover オプションを on に設定します。それ以外の場合、障害が発生したサーバは、再起動後にトランザクションが失われたかどうかを確認します。トランザクションが失われていた場合、データベースサーバメッセージログにメッセージを書き込んで、データベースをシャットダウンします。現在のデータベースとトランザクションログは、ミラーリングを続行する前にバックアップを使用して置き換える必要があります。

### i 注記

非同期モードまたは非同期フルページモードを使用している場合は、auto\_failover ミラーリングオプションを ON に設定します。それによって、プライマリサーバで障害が発生した場合、ミラーサーバが自動的にプライマリサーバとなります。auto\_failover オプションを使用すると、トランザクションの損失につながる可能性があります。

synchronize\_mirror\_on\_commit データベースオプションでは、非同期モードまたは非同期フルページモードにおいて、データベースの変更がミラーサーバへ送信されたことがどの時点で保証されるかについて制御します。このオプションを On に設定すると、COMMIT が発生するたびに、トランザクションログに記録されたすべての変更がミラーサーバへ送信され、その変更がミラーサーバによって受信されると、ミラーサーバからプライマリサーバへ受信確認が送信されます。SET TEMPORARY OPTION を使用すれば、特定のトランザクションにオプションを設定できます。また、ログインプロシージャで APPINFO 文字列を調べ、このオプションを特定のアプリケーションだけに設定すると、効果的な場合もあります。

SQL Anywhere では、データベースミラーリングシステムでフェイルオーバーが発生したときに、使用しているモードに関係なく起動するシステムイベントをサポートしています。これらのイベントを使用することで、フェイルオーバーの発生時に管理者に通知することなどができます。

非同期モードまたは非同期フルページモードで実行している場合、フェイルオーバーが発生してトランザクションがデータベースにコミットされない場合の対処法を決定する必要があります。

非同期モードまたは非同期フルページモードのいずれかで実行しているミラーリングシステムは、プライマリサーバに対して同期モードで動作しているときよりも待機する機会をより少なくするように要求します。しかし、非同期モードまたは非同期フルページモードであっても、プライマリサーバからミラーサーバに送信されるメッセージのうちいくつかは、ミラーからの受信確認を待機するので、これが遅延を引き起こすこともあります。さらに、パケットをミラーに送信するときには少量のオーバーヘッドも存在します。非同期モードまたは非同期フルページモードは、トランザクションが失われる可能性があるため、お奨めできません。

このセクションの内容:

#### [同期ステータスのミラーリング \[1673 ページ\]](#)

ミラーリングシステムで同期モードを使用しているとき、ステータスは `synchronizing` または `synchronized` の 2 つのうちいずれかです。

## 関連情報

[トラブルシューティング: プライマリサーバの障害からのリカバリ方法 \[1693 ページ\]](#)

[トラブルシューティング: プライマリサーバの障害からのリカバリ方法 \[1693 ページ\]](#)

[synchronize\\_mirror\\_on\\_commit オプション \[805 ページ\]](#)

[トラブルシューティング: データベースミラーリングシステムイベントを使用してフェイルオーバー発生時に通知を送信する \[1683 ページ\]](#)

### 1.10.1.4.4.1 同期ステータスのミラーリング

ミラーリングシステムで同期モードを使用しているとき、ステータスは `synchronizing` または `synchronized` の 2 つのうちいずれかです。

#### **synchronizing**

ミラーサーバが接続されていないか、プライマリサーバのログページの読み込みが完了していません。同期モードが非同期の場合、この値も返されます。 `is`

#### **synchronized**

ミラーサーバが接続され、プライマリサーバ上でコミットされたすべての変更が反映されています。

パートナーサーバのどれかが起動し、ミラーとして動作することになった場合、まず取得していないログページをプライマリサーバに要求します。この処理には、プライマリサーバ上で現在アクティブなログ以外のログファイルからページをコピーすることも含まれます。ミラーは、ページを受信しては、そこに含まれている変更を自身のデータベースのコピーに適用します。ミラーがプライマリからすべてのページを受信すると、プライマリとミラーは同期した状態になります。この時点以降、プライマリ上でコミットされたあらゆる変更はミラーに送信され、ミラーによって受信確認される必要が生じます。

非同期モードと非同期フルページモードでは、ミラーは上記と同様にログページを要求します。ただし、2 つのサーバが同期した状態になることはありません。ミラーがプライマリから取得できるすべてのログページを要求すると、プライマリは、ページが更新されたらすべてミラーに送信するよう通知されます。

読み取り専用スケールアウトシステムでは、コピーノードの同期ステータスは常に同期されます。

## 1.10.15 データベースミラーリングシステムの設定

プライマリサーバ、ミラーサーバ、監視サーバでデータベースおよびトランザクションログファイルのコピーを保持するように設定することで、ミラーリングシステムを設定します。

### 前提条件

MANAGE ANY MIRROR SERVER システム権限が必要です。

このタスクでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

### コンテキスト

運用ミラーリングシステムでは、パートナーサーバおよび監視サーバを 3 つの別個のコンピュータで実行する必要があります。データベースミラーリングシステムを設定する場合は、次のすべての例で、localhost とポート番号を対応するデータベースサーバが実行されるコンピュータ名とポートに変更する必要があります。

### 手順

1. `-su` オプションと `-xp on` オプションを使用して、データベースサーバでミラーリングするデータベースを起動します。このサーバは、データベースのパートナーの 1 つである初期プライマリサーバになります。データベースにはトランザクションログが必要です。次に例を示します。

```
dbsrv17 -n mirror_server1 -x tcpip(PORT=6871;DOBROAD=no) -su passwd "c:
¥server1¥mirror_demo.db" -xp on
```

2. データベースに接続します。次に例を示します。

```
dbisql -c "UID=DBA;PWD=passwd;SERVER=mirror_server1"
```

3. CREATE MIRROR SERVER 文を使用して、データベースのパートナーサーバと監視サーバを定義します。プライマリサーバとミラーサーバは、データベースシステムのパートナーとしても定義する必要があります。

次に例を示します。

- 次の 2 つの SQL 文は、データベースミラーリングシステムのパートナーサーバとして mirror\_server1 と mirror\_server2 を定義します。

```
CREATE MIRROR SERVER mirror_server1
AS PARTNER
connection_string='SERVER=mirror_server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1.state';
CREATE MIRROR SERVER mirror_server2
AS PARTNER
connection_string='SERVER=mirror_server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2.state';
```

- 次の SQL 文が定義するのは次のとおりです。
  - mirror\_server1 の代替サーバ名としての mirror\_demo\_primary。クライアントはプライマリサーバとして動作するデータベースサーバに接続するために、mirror\_demo\_primary をサーバ名として使用します。
  - mirror\_server2 の代替サーバ名としての mirror\_demo\_mirror。クライアントはミラーサーバとして動作するデータベースサーバに接続するために、mirror\_demo\_mirror をサーバ名として使用します。

```
CREATE MIRROR SERVER mirror_demo_primary
AS PRIMARY
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6872';
CREATE MIRROR SERVER mirror_demo_mirror
AS MIRROR
connection_string='SERVER=mirror_demo_mirror;HOST=localhost:6871,localhost:6872';
```

- 次の SQL 文は、データベースミラーリングシステムの監視サーバを定義します。

```
CREATE MIRROR SERVER demo_arbiter
AS ARBITER
connection_string = 'SERVER=demo_arbiter;HOST=localhost:6870';
```

4. ミラーリングシステムのミラーリングオプションを設定します。認証文字列の指定は必須です。次に例を示します。

```
SET MIRROR OPTION authentication_string='abc';
```

5. プライマリデータベースおよび現在のトランザクションログのコピーを他のトランザクションログとともに 2 番目のパートナーが存在するコンピュータ上に作成します。たとえば、BACKUP DATABASE 文を使用します。

プライマリサーバコンピュータとミラーサーバコンピュータ上のトランザクションログファイルは、現在のトランザクションログファイルの開始オフセットを含め、同一である必要があります。

6. データベースミラーリングシステムの 2 つ目のデータベースサーバを起動します。

```
dbsrv17 -n mirror_server2 -x tcpip(PORT=6872;DOBROAD=no) -su passwd "c:
¥server2¥mirror_demo.db" -xp on
```

7. 監視サーバを起動します。

```
dbsrv17 -n demo_arbiter -su passwd -x "TCPIP(PORT=6870;DOBROAD=no)" -xf "c:
¥arbiter¥arbiter.state" -xa "AUTH=abc;DBN=mirror_demo"
```

## 結果

これで、クライアントはミラーリングされたデータベースに接続できるようになりました。

システムのデータベースサーバを設定するには、プライマリとミラーのロールが必要です。これらのサーバに付けた名前は、クライアントからデータベースサーバに接続するときに代替サーバ名として使用されます。どちらか一方のパートナーサーバは、プライマリサーバまたはミラーサーバとして動作できます。

## 次のステップ

システムをモニタのリソースリストに追加して、データベースミラーリングシステムをモニタします。

SQL Central からプライマリデータベースに接続して、データベースミラーリングシステムのデータベースサーバのステータスを確認します。正常性と統計情報ウィンドウ枠にデータベースミラーリング情報が表示されます。

このセクションの内容:

[データベースミラーリングシステムがプライマリサーバを選択する仕組み \[1676 ページ\]](#)

通常の起動では、次の入力によって、どのサーバがプライマリサーバになるかが決まります。

[データベースミラーリングシステムでのトランザクションログの管理 \[1678 ページ\]](#)

パートナーサーバは、起動されると、現在のトランザクションログファイルが存在するディレクトリ内のすべてのトランザクションログファイルを調べて、適用する必要があるファイルを特定します。

[監視サーバとしてコピーノードを使用する \[1678 ページ\]](#)

読み込み専用のスケールアウトシステムを含むミラーリングシステムの監視サーバを実行するために、新しいデータベースサーバを作成しない場合は、1 つのサーバをコピーノードおよび監視サーバとして動作するように設定します。

## 関連情報

[データベースの正常性と統計情報 \[1058 ページ\]](#)

[レッスン 7: \(省略可\) データベースミラーリングシステムのモニタリング \[1297 ページ\]](#)

[ミラーリングシステムのデータベースサーバの停止 \(dbstop ユーティリティ\) \[1684 ページ\]](#)

[トラブルシューティング: プライマリサーバが起動できない \[1694 ページ\]](#)

[-su データベースサーバオプション \[480 ページ\]](#)

[-xp データベースオプション \[537 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

### 1.10.1.5.1 データベースミラーリングシステムがプライマリサーバを選択する仕組み

通常の起動では、次の入力によって、どのサーバがプライマリサーバになるかが決まります。

- ステータス情報ファイルの内容
- 各データベースサーバのトランザクションログの位置
- 優先プライマリサーバの指定

`-xp on` を指定して起動したデータベースサーバ (たとえば S1) で、ミラーリングが定義されていないデータベースを起動してからミラーリングを定義した場合、S1 がデータベースの初期プライマリサーバになります。

または、データベースミラーリングシステムを初めて設定するときに、ステータス情報ファイルがなく、データベースとトランザクションログのコピーが同じである場合、どちらのサーバでもプライマリとして使用できます。この場合は、サーバ名が比較され、名前の小さい方がプライマリサーバになります。たとえば、名前 server1 は server2 より小さいので、server1 がプライマリサーバになります。

最初に起動するときは、ロールに合意できるように、両方のサーバが稼働し接続されている必要があります。ステータス情報ファイルに記録されている、以前のステータス情報がないため、監視サーバの存在だけでは不十分です。

このセクションの内容:

[データベースミラーリングシステムで使用するデータベースサーバ \[1677 ページ\]](#)

データベースミラーリングシステムでは、使用するサーバとして1つのパートナーサーバを指定できます。

## 関連情報

[トラブルシューティング: パートナおよび監視サーバのステータス情報ファイル \[1692 ページ\]](#)

[データベースミラーリングでのクォーラム \[1662 ページ\]](#)

### 1.10.1.5.1.1 データベースミラーリングシステムで使用するデータベースサーバ

データベースミラーリングシステムでは、使用するサーバとして1つのパートナーサーバを指定できます。

優先サーバはデフォルトでプライマリサーバとして実行されるパートナーで、データベースの所有権を持ちます。

使用するサーバが利用できなくなった場合、ミラーサーバとして動作していたサーバがプライマリサーバになります。使用するサーバの再起動時には、次のことが行われます。

1. 現在のプライマリサーバから、まだ取得していないトランザクションログエントリを取得します。
2. 現在のプライマリサーバに、データベースの所有権を放棄するように依頼します。サーバがロールを切り替えます。使用するサーバがプライマリサーバになり、他のサーバはミラーサーバになります。データベースの所有権が変更されると、優先サーバではないサーバにあるデータベースへの接続は失われます。

パートナーサーバを定義する CREATE MIRROR SERVER 文に PREFERRED='YES' を追加することによって、優先サーバを指定します。

#### 例

次の文では、使用するサーバとして mirror\_server1 サーバが作成されます。

```
CREATE MIRROR SERVER mirror_server1
AS PARTNER
connection_string='SERVER=mirror_server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1.state'
preferred='YES';
```

## 関連情報

[ユーザが開始したロールの切り替え \(フェイルオーバー\) \[1682 ページ\]](#)

[データベースミラーリングでのクォーラム \[1662 ページ\]](#)

## 1.10.1.5.2 データベースミラーリングシステムでのトランザクションログの管理

パートナーサーバは、起動されると、現在のトランザクションログファイルが存在するディレクトリ内のすべてのトランザクションログファイルを調べて、適用する必要があるファイルを特定します。

データベースサーバは、トランザクションログに記録された操作をデータベースに適用してから、プライマリサーバとミラーサーバのどちらとして動作するかを判断します。トランザクションログファイルとデータベースファイルは同一のディレクトリに格納できます。ただし、このディレクトリは、データベースの起動を遅延させる可能性があるため、他のファイルは入れるべきではありません。特にデータベースのトランザクションログファイルが入っているディレクトリには、他のデータベースのトランザクションログファイルを入れるべきではありません。

プライマリサーバロールを引き受けたサーバは、ミラーサーバ上の現在のトランザクションログと同一の開始オフセットのトランザクションログを持ち、同様に、その後のトランザクションログファイルがあればプライマリサーバの現在のトランザクションログファイルまでのものを持つ必要があります。

ミラーサーバロールを引き受けたサーバは、プライマリサーバからトランザクションログページの受信を開始します。プライマリでトランザクションログの名前が変更された場合は、ミラーでも名前が変更されます。ミラーは、新しいトランザクションログページを、トランザクションログ用に指定された名前の新しいファイルに書き出します。

プライマリでは、トランザクションログファイルを定期的に削除できます。トランザクションログファイルの名前が変更されるたび、ミラーには、プライマリで削除されずに残っている最も古いトランザクションログファイルがどれであるかが通知されます。ミラーでも、通知されたものより古いトランザクションログファイルはすべて削除されます。

トランケーションがトランザクションの損失につながる可能性があるため、データベースミラーリングを使用しているときは、プライマリで現在のトランザクションログをトランケートできません。トランザクションログの名前は、必要に応じて何度でも変更できます。古いトランザクションログをプライマリから削除するには、スケジュールされたイベントを使用し、ログが必要なくなったことが確認されたら削除します。たとえば毎日実行するイベントを作成し、トランザクションログの名前を変更できます。このイベントは、1週間以上前のトランザクションログのコピーを削除することもできます。

### 関連情報

[トラブルシューティング: データベースミラーリングシステム \[1691 ページ\]](#)

## 1.10.1.5.3 監視サーバとしてコピーノードを使用する

読み込み専用のスケールアウトシステムを含むミラーリングシステムの監視サーバを実行するために、新しいデータベースサーバを作成しない場合は、1つのサーバをコピーノードおよび監視サーバとして動作するように設定します。

### 前提条件

MANAGE ANY MIRROR SERVER システム権限が必要です。

このタスクでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

## 手順

1. 監視サーバにもしたいコピーノードを停止します。
2. 次のオプションのうちの1つを選択してください。コマンドラインメソッド (dbsrv「XX」または dbeng「XX」) は、次の例では複数行になっていますが、すべて1行に入力してください。

オプション	アクション
監視サーバを既存のコピーノードに追加する	<ol style="list-style-type: none"> <li>1. -xf、-su、および -xa データベースサーバオプションを使用してコピーノードを起動します。次に例を示します。 <pre>dbsrv17 -n scaleout_child_demo -su passwd -x "tcpip(PORT=6873)" -xf "c:¥scaleoutdemo¥copynode¥arbiter.state" -xa "AUTH=abc;DBN=scaleoutdemo" "c:¥scaleoutdemo¥copynode ¥scaleoutdemo.db" -xp on</pre> </li> <li>2. 監視サーバの定義を作成して、その名前がミラーリングシステムのどのデータベースサーバの名前とも一致しないようにします。CREATE MIRROR SERVER 文を実行します。次に例を示します。 <pre>CREATE MIRROR SERVER myarbiter AS ARBITER connection_string = 'SERVER=scaleout_child_demo;HOST=localhost: 6873';</pre> </li> </ol>
コピーノードを既存の監視サーバに追加する	<ol style="list-style-type: none"> <li>1. 既存の監視サーバを停止します。</li> <li>2. プライマリサーバに接続し、既存の監視ミラーサーバ定義を削除します。</li> <li>3. プライマリサーバから異なる名前前の (ただし connection_string にある実際のサーバ名を持つ) 新しい監視ミラーサーバ定義を作成します。次に例を示します。 <pre>CREATE MIRROR SERVER newarbiter AS ARBITER connection_string = 'SERVER=actual_server_name;HOST=arbiter_host: 6870';</pre> </li> <li>4. ミラーされるデータベースのバックアップを取り、コピーノードが実行されるコンピュータ上にコピーします。</li> <li>5. 監視サーバを再起動しますが、mirrored.db がミラーされるデータベースの名前になっている、サーバコマンドに mirrored.db -xp ON を含めます。</li> </ol>

## 結果

監視サーバもスケールアウトコピーノードで実行されます。

## 例

たとえば、次のサーバ定義を使用するミラーリングおよび読み込み専用スケールアウトシステムに、TheArbiter という名前のデータベースサーバはありません。

```
CREATE MIRROR SERVER "scaleout_child"  
AS COPY  
connection_string = 'SERVER=scaleout_child;HOST=winxp-2:6878';  
CREATE MIRROR SERVER "TheArbiter"  
AS ARBITER  
connection_string = 'SERVER=scaleout_child;HOST=winxp-2:6878';
```

## 1.10.1.6 データベースミラーリングシステムの保守

データベースモニタリングシステムの保守は重要です。

このセクションの内容:

### [データベースミラーリングシステムでのバックアップ \[1681 ページ\]](#)

データベースミラーリングは、データ損失のリスクを最小限に留めますが、データベースミラーリングシステムに参加しているデータベースのバックアップを取り、検証することを強くお奨めします。

### [データベースミラーリングでのロールの切り替え \(フェイルオーバー\) \[1681 ページ\]](#)

プライマリサーバがハードウェアやソフトウェアの障害により使用できなくなった場合、ミラーサーバが監視サーバとネゴシエートしてデータベースの所有権を取得し、プライマリサーバのロールを引き受けます。

### [ミラーリングシステムのデータベースサーバの停止 \(dbstop ユーティリティ\) \[1684 ページ\]](#)

ユーティリティデータベースで、ミラーリングシステムのデータベースを停止します。

### [ミラーリングシステムからのミラーサーバの削除 \[1685 ページ\]](#)

ミラーサーバ定義 (CREATE MIRROR SERVER...AS PARTNER および CREATE MIRROR SERVER...AS MIRROR) を削除することで、ミラーシステムからミラーサーバを削除します。

### [ミラーリングされたデータベースを実行中のミラーリングシステムに追加する \[1686 ページ\]](#)

ミラーリングされたデータベースを、既存のサーバの可用性を低減することなく実行中のミラーリングシステムに追加します。

### [パートナーサーバの移動 \[1689 ページ\]](#)

現在のミラーサーバのミラー定義を削除し、新しいサーバを作成してミラーリングシステムに追加することによって、システムを停止しないでデータベースミラーリングシステム内のサーバを別のサーバに移動します。

### [監視サーバの移動 \[1690 ページ\]](#)

新しいサーバを作成し、新しい監視サーバを使用するように監視ミラーサーバ定義を変更して、ミラーリングシステムを停止することなく監視サーバを移動します。

## 1.10.1.6.1 データベースミラーリングシステムでのバックアップ

データベースミラーリングは、データ損失のリスクを最小限に留めますが、データベースミラーリングシステムに参加しているデータベースのバックアップを取り、検証することを強くお奨めします。

- BACKUP DATABASE 文を使用すると、データベースサーバを基準にしてバックアップを実行できます。BACKUP DATABASE 文はプライマリデータベースサーバ上で実行されるので、指定するファイル名には、プライマリおよびミラーのデータベースサーバ両方で一貫性のあるネットワークドライブまたは UNC 名を指定する必要があります。
- dbbackup ユーティリティを使用して、データベースのクライアント側バックアップを実行できます。

### 関連情報

[トラブルシューティング: プライマリサーバの障害からのリカバリ方法 \[1693 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

## 1.10.1.6.2 データベースミラーリングでのロールの切り替え (フェイルオーバー)

プライマリサーバがハードウェアやソフトウェアの障害により使用できなくなった場合、ミラーサーバが監視サーバとネゴシエートしてデータベースの所有権を取得し、プライマリサーバのロールを引き受けます。

所有権の譲渡、すなわちロールの切り替えを実施するには、ダウンしていないパートナーサーバと監視サーバは、ロールの切り替えを行おうとする時点で、ミラーが最新で同期された状態にあることを合意する必要があります。

- それまで元のプライマリサーバに接続されていたクライアントはすべて切断され、コミットされていないトランザクションはすべて失われます。クライアントがデータベースへのアクセスを続行するには、新しいプライマリサーバ上のデータベースに接続し直す必要があります。元のプライマリサーバは、再び使用可能になると、ミラーサーバのロールを引き受けます。
- 元のミラーデータベースに接続していたクライアントは、フェイルオーバーが発生し、ミラーサーバがプライマリサーバになったときにも維持されます。フェイルオーバー後は、これらのクライアント接続によってデータベースに変更を加えることができます。

データベースサーバのデータベースサーバメッセージウィンドウには、起動時に、そのサーバが引き受けているロールと起動プロセスの進捗状況を示すステータスメッセージが表示されます。また、ミラーリングシステムを構成する他の 1 つ以上のサーバが失われたことが原因でデータベースの再起動が必要になった場合、またはロールがミラーからプライマリに変更された場合に、メッセージが表示されます。

ミラーリングシステムを構成するいずれかのサーバでアサーションが失敗すると、そのサーバはデータベースサーバメッセージログにエラーを出力して終了します。これにより、他のサーバに障害発生が通知され、適切なアクションが実行されます。

高可用性環境のデータベースで、互換性がない、トランザクションログが一致しないなどの問題が発生すると、データベースは停止します。問題が発生したデータベースを実行していたデータベースサーバも、他に実行中のデータベースがないかぎりシャットダウンします。

データベースミラーリングシステムの各データベースのステータスの詳細は、ステータス情報ファイルに格納されます。

- 監視サーバが存在する場合、プライマリからミラーへのフェイルオーバーは自動で実行されます。同期モードで実行している場合、コミットされたトランザクションがフェイルオーバー中に失われることはありません。

- ミラーサーバにトランザクションログが適用済みなので、フェイルオーバーが非常に高速になります。ミラーがプライマリの障害を検知すると、コミットされていないトランザクションをすべてロールバックし、データベースを使用可能にします。

プライマリが同期ステータスでありながら利用できなくなった場合、ミラーはプライマリとして引き継ぐことができません。ミラーリングシステムは、プライマリが利用できるようになるまで待機します。

データベースミラーリングシステムでフェイルオーバーが発生したときに、使用しているモードに関係なく起動するシステムイベントがサポートされています。これらのイベントを使用することで、フェイルオーバーの発生時に管理者に通知することなどができます。

このセクションの内容:

#### [ユーザが開始したロールの切り替え \(フェイルオーバー\) \[1682 ページ\]](#)

プライマリサーバに接続することで、サーバを停止することなくプライマリサーバからミラーサーバにデータベースミラーリングのフェイルオーバー (ロールの切り替え) を開始します。

#### [トラブルシューティング: データベースミラーリングシステムイベントを使用してフェイルオーバー発生時に通知を送信する \[1683 ページ\]](#)

ミラーデータベースでアクションが必要であることを電子メール経由で通知するために、MirrorServerDisconnect および MirrorFailover イベントを使用します。

## 関連情報

[トラブルシューティング: パートナおよび監視サーバのステータス情報ファイル \[1692 ページ\]](#)

### 1.10.1.6.2.1 ユーザが開始したロールの切り替え (フェイルオーバー)

プライマリサーバに接続することで、サーバを停止することなくプライマリサーバからミラーサーバにデータベースミラーリングのフェイルオーバー (ロールの切り替え) を開始します。

次の文を実行します。

```
ALTER DATABASE SET PARTNER FAILOVER;
```

この文を実行すると、データベースに対する既存の接続は、文を実行した接続も含めて、閉じられます。この文がプロシージャまたはイベントに含まれている場合、後続する他の文は実行されない可能性があります。この文の実行に必要な権限は、-gd サーバオプションによって制御されます。

この文は、優先サーバを指定する代わりに使用できます。また、特定のデータベースサーバに対してデータベースの所有権が指定されているときに制御するロジックで使用することができます。たとえば、パートナサーバ (PartnerState データベースプロパティの値で決まります) の可用性、またはデータベースに対する接続数 (ConnCount データベースプロパティの値で決まります) に基づいて、フェイルオーバーを起動できます。

## 関連情報

[データベースミラーリングシステムで使用するデータベースサーバ \[1677 ページ\]](#)

### 1.10.1.6.2.2 トラブルシューティング: データベースミラーリングシステムイベントを使用してフェイルオーバー発生時に通知を送信する

ミラーデータベースでアクションが必要であることを電子メール経由で通知するために、MirrorServerDisconnect および MirrorFailover イベントを使用します。

ただし、プライマリサーバ上で実行されているデータベースが使用できなくなるあらゆる状況でこれらのイベントが起動するわけではありません。たとえば、プライマリサーバとミラーサーバの両方に影響する停電があった場合、どちらのイベントも起動しません。このような監視が必要な場合は、別のコンピュータでスクリプト言語を使用することで実装できます。たとえば、dbping を定期的呼び出してミラーデータベースに接続します。

#### 例

次の例は、フェイルオーバーの発生を管理者に通知するイベントを作成しています。

```
CREATE EVENT mirror_server_unavailable
TYPE MirrorServerDisconnect
HANDLER
BEGIN
  CALL xp_startmail ( mail_user = 'George Smith',
                    mail_password = 'mypwd' );
  CALL xp_sendmail( recipient='DBAdmin',
                  subject='Mirror server disconnect occurred',
                  "message"='The following server is unavailable in the mirroring system: '
                    || event_parameter( 'MirrorServerName' ) );
  CALL xp_stopmail ( );
END;
```

## 関連情報

[Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

## 1.10.1.6.3 ミラーリングシステムのデータベースサーバの停止 (dbstop ユーティリティ)

ユーティリティデータベースで、ミラーリングシステムのデータベースを停止します。

### 前提条件

-su データベースサーバオプションを使用してデータベースサーバを起動し、ユーティリティデータベースにパスワードを設定しておく必要があります。ユーティリティデータベースのパスワードが必要です。

### コンテキスト

ミラーリングシステムのデータベースサーバを停止して、サポートパッケージまたはマイナリリースを適用したり、ミラーリングシステムからサーバを削除したりする必要があるときには、データベースの停止が役立ちます。

### 手順

ユーティリティデータベースに接続して、dbstop ユーティリティを実行します。接続文字列でユーティリティデータベースの接続パラメータを指定する必要があります。

たとえば、次のコマンドを実行するとデータベースサーバ mirror\_server1 が停止します。

```
dbstop -c "UID=DBA;PWD=sql;DBN=utility_db;LINKS=tcPIP" mirror_server1
```

### 結果

データベースサーバが停止します。高可用性システムに 2 台以上のサーバがある場合、ユーザは、いずれかのデータベースのコピーに接続可能である必要があります。ただし、停止するサーバによって、データベースを更新できたり、できなかったりする場合があります。

### 次のステップ

プライマリサーバ、ミラーサーバ、または監視サーバを停止しても、システムは動作し続けますが、高可用性を維持するため、停止したデータベースサーバを再起動するか、置き換える必要があります。

## 関連情報

[データベースサーバの起動方法 \[310 ページ\]](#)

[ユーティリティデータベースへの接続 \(接続ウィンドウの場合\) \[298 ページ\]](#)

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.10.1.6.4 ミラーリングシステムからのミラーサーバの削除

ミラーサーバ定義 (CREATE MIRROR SERVER...AS PARTNER および CREATE MIRROR SERVER...AS MIRROR) を削除することで、ミラーシステムからミラーサーバを削除します。

#### 前提条件

データベースミラーリングシステムの一部として実行されているミラーサーバは削除しないでください。

削除できるのはミラーサーバだけです。削除したいサーバがプライマリサーバの場合は、プライマリサーバとミラーサーバのロールが切り替わるように、フェイルオーバーを開始する必要があります。

MANAGE ANY MIRROR SERVER システム権限が必要です。

#### コンテキスト

同じサーバ名を保持して設定内容を変更する場合、CREATE OR REPLACE MIRROR SERVER 文または ALTER MIRROR SERVER 文を使用できます。

#### 手順

1. プライマリサーバのデータベースに接続します。
2. ミラーサーバに子コピーノードがある場合は、ALTER MIRROR SERVER 文を実行し、子コピーノードを別の親に再割り当てします。
3. DROP MIRROR SERVER 文を実行し、ミラーサーバ名を指定して、ミラーサーバ定義を削除します。

```
DROP MIRROR SERVER mirror_server_name;
```

4. DROP MIRROR SERVER 文を実行し、パートナー名を指定して、パートナーサーバ定義を削除します。

```
DROP MIRROR SERVER partner_server_name;
```

5. (省略可) データベースサーバの停止

## 結果

ミラーサーバが削除されます。コピーノードが残っている場合、データベースミラーリングシステムは読み取り専用スケールアウトシステムになります。

### 例

ミラーリングシステムのパートナーサーバは、次の文で作成されます。

```
CREATE MIRROR SERVER mirror_server1
AS PARTNER
connection_string='SERVER=mirror_server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1.state';
CREATE MIRROR SERVER mirror_server2
AS PARTNER
connection_string='SERVER=mirror_server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2.state';

CREATE MIRROR SERVER myprimary
AS PRIMARY
connection_string='SERVER=myprimary;HOST=localhost:6871,localhost:6872';
CREATE MIRROR SERVER mymirror
AS MIRROR
connection_string='SERVER=mymirror;HOST=localhost:6871,localhost:6872';
```

次の文を実行し、ミラーサーバ (mirror\_server2) を削除します。

```
DROP MIRROR SERVER mirror_server2;
DROP MIRROR SERVER mymirror
```

## 1.10.1.6.5 ミラーリングされたデータベースを実行中のミラーリングシステムに追加する

ミラーリングされたデータベースを、既存のサーバの可用性を低減することなく実行中のミラーリングシステムに追加します。

### 前提条件

-su データベースサーバオプションを使用してプライマリサーバとミラーサーバを起動し、ユーティリティデータベースにパスワードを設定しておく必要があります。ユーティリティデータベースのパスワードが必要です。

このタスクでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

MANAGE MIRROR SERVER 権限も必要です。

監視サーバおよび既存のミラーリングシステムのパートナーの1つが実行されている必要があります。

追加するデータベースにはトランザクションログが必要です。

## コンテキスト

ミラーリングシステムを実行し、既存のパートナーと監視サーバを使用して別のデータベースをミラーリングできます。

ディレクトリにつき1つだけのミラーリングされたデータベースと、関連するログファイルが存在する必要があります。サーバが複数のミラーリングされたデータベースを実行中の場合、各データベースは専用のディレクトリにある必要があります。

## 手順

1. 次を実行し、監視サーバが起動していることを確認します。
  - DBN パラメータによる `-xa` データベースサーバオプション。DBN=\* を指定し、監視サーバがミラーリングされたデータベースの接続を受け入れるようにします。指定しない場合、動的に起動する必要のあるすべてのデータベースを指定し、このサーバを監視サーバとして使用する必要があります。
  - `-su` データベースサーバオプション。監視サーバのユーティリティデータベースに接続できる必要があります。

次に例を示します。

```
dbsrvl7 -n mirror_arbiter -su passwd -x "TCPIP(PORT=6870;DOBRoad=no)" -xf "c:¥arbiter¥arbiter.state" -xa "AUTH=abc;DBN=*" -o c:¥arbiter¥arbiter.conslog
```

2. トランザクションログとともに、ミラーリングしたいデータベースを既存のパートナーコンピュータの1つにコピーします。追加のデータベースが、新しいディレクトリに存在する必要があります。たとえば、C:¥server1¥second です。
3. 既存のパートナーサーバのユーティリティデータベースに接続します。次に例を示します。

```
dbisql -c "UID=dba;PWD=sql;DBN=utility_db;SERVER=mirror_server1"
```

4. START DATABASE 文を MIRROR ON および AUTOSTOP OFF 句とともに使用し、追加するデータベースを起動します。次に例を示します。

```
START DATABASE 'c:¥¥server1¥¥second¥¥second.db'  
AUTOSTOP OFF  
MIRROR ON;
```

データベースが起動します。

5. データベースに接続します。次に例を示します。

```
dbisql -c UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=second
```

SET MIRROR OPTION を使用して、ミラーサーバ定義 (PRIMARY、MIRROR、ARBITER および両方の PARTNERS) および認証文字列を作成します。

新しいミラーリングシステムの認証文字列は、監視サーバの `-xa` オプションで指定した認証文字列と一致している必要があります。ホスト名とポートは、既存のパートナーおよび監視サーバのホスト名とポートに一致している必要があります。PARTNER ミラーサーバ名は、パートナーサーバの `-n` サーバオプションに一致している必要があります。PRIMARY および MIRROR サーバ名は、現在システムが使用しているサーバ名と重複してはいけません。

次に例を示します。

```
CREATE MIRROR SERVER second_primary  
AS PRIMARY  
connection string='SERVER=second_primary;HOST=localhost:6871,localhost:6872';  
CREATE MIRROR SERVER second_mirror
```

```

AS MIRROR
connection_string='SERVER=second_mirror;HOST=localhost:6871,localhost:6872';
CREATE MIRROR SERVER mirror_server1
AS PARTNER
connection_string='SERVER=mirror_server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1.state';
CREATE MIRROR SERVER mirror_server2
AS PARTNER
connection_string='SERVER=mirror_server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2.state';
CREATE MIRROR SERVER arbiter_server
AS ARBITER
connection_string='SERVER=mirror_arbiter;HOST=localhost:6870';
SET MIRROR OPTION authentication_string='abc';

```

6. データベースのバックアップを作成し、そのバックアップを 2 番目のパートナーサーバにコピーします。たとえば、次のように dbbackup ユーティリティを使用します。

```

dbbackup -c "UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=second" "c:
¥server2¥second"

```

7. 2 番目のパートナーサーバのユーティリティデータベースに接続します。次に例を示します。

```

dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server2;DBN=utility_db"

```

8. START DATABASE 文を AUTOSTOP OFF MIRROR ON 句とともに使用して、2 番目のパートナーサーバでデータベースを起動します。

```

START DATABASE 'c:¥¥server2¥¥second¥¥second.db'
AUTOSTOP OFF
MIRROR ON;

```

## 結果

新たに追加された、ミラーリングされたデータベースを両方のパートナーで実行する必要があります。1つのパートナーがプライマリのロールを持ち、別のパートナーがミラーのロールを持つことを（コンソールログまたは DB\_PROPERTY( 'MirrorRole' ) のいずれかを使用して) 確認します。

## 関連情報

[-xa データベースサーバオプション \[502 ページ\]](#)

## 1.10.1.6.6 パートナサーバの移動

現在のミラーサーバのミラー定義を削除し、新しいサーバを作成してミラーリングシステムに追加することによって、システムを停止しないでデータベースミラーリングシステム内のサーバを別のサーバに移動します。

### 前提条件

MANAGE ANY MIRROR SERVER 権限が必要です。

このタスクでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

### コンテキスト

例を試すときには、localhost を実際のコンピュータ名に置き換えます。

### 手順

1. 移動するパートナサーバに接続し、パートナサーバにミラーロールがあることを確認します。移動できるのはミラーロールを持つパートナのみです。移動したいサーバがプライマリサーバの場合は、プライマリサーバとミラーサーバのロールが切り替わるように、フェイルオーバーを開始する必要があります。
2. 新しいパートナサーバの新しいディレクトリを作成します。
3. プライマリサーバに接続します。
4. DROP MIRROR SERVER 文を実行し、移動するサーバのパートナ定義を削除します。

ミラーデータベースが停止します。サーバ上で実行されているデータベースがミラーデータベースのみの場合は、サーバも停止します。

5. 新しいパートナになるサーバのパートナ定義を新規作成します。たとえば、次の文を実行します。

```
CREATE MIRROR SERVER mirror_server3 AS PARTNER
connection_string='SERVER=demo_server3;HOST=localhost:6874'
state_file='c:¥¥server3¥¥server3.state';
```

6. プライマリとミラーの定義を更新します。たとえば、次の文を実行します。

```
ALTER MIRROR SERVER mirror_demo_primary AS PRIMARY
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:
6874';
ALTER MIRROR SERVER mirror_demo_mirror AS MIRROR
connection_string='SERVER=mirror_demo_mirror;HOST=localhost:6871,localhost:6874';
```

7. プライマリデータベースファイルとトランザクションログのコピーを作成し、新しいパートナサーバのディレクトリに追加します。たとえば、次のコマンドを実行します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=mirror_demo" server3
```

- 新しいパートナーがミラーリングシステムに参加できるように、-xp オプションを使用して新しいパートナーサーバを起動します。たとえば、次のコマンドを実行します。

```
dbsrv17 -n mirror_server3 -x "tcpip(PORT=6874)" -su passwd "c:
¥server3¥mirror_demo.db" -xp on
```

- 新しいパートナーサーバに接続し、ミラーサーバであることを確認します。

## 結果

ミラーシステムが新しいパートナーサーバと共に実行されています。

## 次のステップ

プライマリサーバまたはミラーサーバに接続しているクライアントに更新された接続文字列があることを確認し、Host 接続パラメータで両方のパートナーのアドレスを指定します。

## 関連情報

[チュートリアル: パートナサーバの移動 \[1720 ページ\]](#)

### 1.10.1.6.7 監視サーバの移動

新しいサーバを作成し、新しい監視サーバを使用するように監視ミラーサーバ定義を変更して、ミラーリングシステムを停止することなく監視サーバを移動します。

## 前提条件

MANAGE ANY MIRROR SERVER システム権限が必要です。

## 手順

- su、-xa、および -xf の各オプションを使用してサーバを起動し、監視サーバにします。次に例を示します。

```
dbsrv17 -n demo_arbiter2 -x "tcpip(port=6873)" -xf c:¥arbiter2¥arbiter2.state -
xa "AUTH=abc;DBN=mirror_demo" -su passwd
```

2. プライマリサーバに接続し、connection\_string が新しい監視サーバのものになるように、ミラーリングシステムの監視ミラーサーバ定義を変更します。

たとえば、次の文を実行して監視サーバ定義を新しいサーバのものに変更します。

```
ALTER MIRROR SERVER demo_arbiter
AS ARBITER
connection_string='SERVER=demo_arbiter2;HOST=localhost:6873';
```

プライマリサーバとミラーサーバが、その監視サーバから接続を切断して、新しい監視サーバに接続します。

3. 数秒待ってから、古い監視サーバを停止します。

## 結果

監視サーバが新しいサーバに移動します。

## 次のステップ

ミラーリングシステムの ArbiterState データベースプロパティに対して ping を実行し、新しい監視サーバがミラーリングシステムに接続していることを確かめます。たとえば、次のコマンドを実行します。

```
dbping -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary" -pd ArbiterState
```

ArbiterState の値が接続されます。

```
SQL Anywhere Server Ping Utility Version 16.0.4157
Type      Property      Value
-----
Database  ArbiterState  connected
```

## 関連情報

[チュートリアル: 監視サーバの移動 \[1718 ページ\]](#)

### 1.10.1.7 トラブルシューティング: データベースミラーリングシステム

ミラーリングシステムのトラブルシューティングに使用できる拡張データベースプロパティは 3 つあります。

- MirrorState
- PartnerState
- ArbiterState

このセクションの内容:

[トラブルシューティング: パートナおよび監視サーバのステータス情報ファイル \[1692 ページ\]](#)

高可用性システム内の各パートナと監視サーバのそれぞれがステータス情報ファイルを管理し、ミラーリングシステムのステータスについて各サーバの視点から記録を残します。

[トラブルシューティング: プライマリサーバの障害からのリカバリ方法 \[1693 ページ\]](#)

プライマリサーバ障害からのリカバリ手順は、データベースミラーリングシステムで使用している同期実行モードによって異なります。

[トラブルシューティング: プライマリサーバが起動できない \[1694 ページ\]](#)

他の再起動の試行がすべて失敗したときに、ミラーサーバにプライマリサーバを強制的に引き継がせます。

[トラブルシューティング: サーバがデータベースミラーリングシステムで利用できなくなったとき \[1696 ページ\]](#)

データベースのミラーリングシステムでサーバが使用できなくなった場合に何が起るかを理解するのに、いくつかのドキュメント化されたシナリオが役立ちます。

[トラブルシューティング: データベースミラーリングで接続が切断される \[1698 ページ\]](#)

データベースミラーリングシステムでは、データベースサーバ間の接続が切断されると、考慮が必要なことがいくつかあります。

## 1.10.1.7.1 トラブルシューティング: パートナおよび監視サーバのステータス情報ファイル

高可用性システム内の各パートナと監視サーバのそれぞれがステータス情報ファイルを管理し、ミラーリングシステムのステータスについて各サーバの視点から記録を残します。

ステータス情報ファイルは、起動時に、サーバが引き受けるロールの決定に使用されます。サーバのローカルステータスは、データベースミラーリングシステム内の他のサーバのステータスと比較されます。

### 警告

ステータス情報ファイルを変更しないでください。

ミラーリングシステムの各サーバのステータス情報ファイルは、CREATE MIRROR SERVER 文の state\_file オプションを使用して必ず指定する必要があります。

ステータス情報ファイルに含まれている情報は次のとおりです。

フィールド	説明
Owner	どのデータベースサーバがプライマリサーバかを示す。
State	サーバがログページの受信中なのか、または最新状態なのかを示す同期ステータス ("同期中" または "同期" のどちらか) を示す。
Mode	同期実行モード (同期、非同期、非同期フルページのいずれか) を示す。

フィールド	説明
Sequence	データベースミラーリングシステムでフェイルオーバーが発生した回数を示す。このシーケンス番号はロールの切り替えが発生するたびに1ずつ増分されます。これは、サーバの視点でミラーリングシステムの状態が最新かどうかを判断するのに役立ちます。

ステータス情報ファイルの内容の例を次に示します。

```
[demo]
Owner=server2
State=synchronizing
Mode=asynchronous
Sequence=35
```

ステータス情報ファイルがない場合は、自動的に作成されます。ステータス情報ファイルの変更は、データベースサーバのみが行います。ただし、ミラーリングシステムをリセットしたり、再配備するときには、ステータスファイルを削除します。

## 関連情報

[同期ステータスのミラーリング \[1673 ページ\]](#)

[データベースミラーリングモード \[1671 ページ\]](#)

[データベースミラーリング \[1660 ページ\]](#)

### 1.10.1.7.2 トラブルシューティング: プライマリサーバの障害からのリカバリ方法

プライマリサーバ障害からのリカバリ手順は、データベースミラーリングシステムで使用している同期実行モードによって異なります。

#### 同期モード

同期モードで実行している場合は、プライマリサーバに存在するすべてのトランザクションがミラーサーバでもコミットされることが保証されます。ミラーサーバは、ほとんどすべての場合において、ユーザによる介入なしに新しいプライマリサーバになることができます。

#### **i** 注記

まれに、チェックポイントの実行中にプライマリサーバがミラーサーバと監視サーバの両方との接続を失った場合は、次にこれら2つのサーバが接続したときに同期に失敗することがあります。ミラーサーバは、プライマリサーバとのトランザクションログの不一致があり、ミラーデータベースが停止したことを示すエラーをレポートします。この場合は、ミラーサーバが正常に起動して同期するように、現在のプライマリサーバのデータベースをミラーサーバに手動でコピーまたはバックアップする必要があります。

## 非同期モードまたは非同期フルページモード

非同期モードおよび非同期フルページモードでは、プライマリサーバに適用されたコミット済みトランザクションがミラーサーバにすべて適用されているとは限らないので、プライマリサーバからミラーサーバへのフェイルオーバーは自動ではありません。このため、自動フェイルオーバーの実行が指定されていないかぎり、非同期のどちらかのモードを使用する場合は、ミラーサーバはデフォルトで、プライマリに障害が発生してもデータベースの所有権を取得できません。障害が発生したサーバは、再起動後にトランザクションが失われたかどうかを確認します。トランザクションが失われていた場合、データベースサーバメッセージログにメッセージを書き込んで、データベースをシャットダウンします。

元のミラーサーバを新しいプライマリサーバとして起動する場合、両方のサーバ上のデータベースファイルを同じ状態にする方法が2つあります。

- データベースとトランザクションログファイルを元のプライマリサーバからミラーサーバにコピーし、ミラーサーバを新しいプライマリサーバとして起動します。ALTER DATABASE 文を FORCE START 句とともに使用して、サーバを強制的にプライマリサーバにできます。
- 元のミラーサーバで、dbbackup を使用してバックアップを実行します。元のプライマリサーバにファイルをコピーし、データベースサーバを起動します。

## 関連情報

[データベースミラーリングモード \[1671 ページ\]](#)

[バックアップユーティリティ \(dbbackup\) \[1070 ページ\]](#)

### 1.10.1.7.3 トラブルシューティング: プライマリサーバが起動できない

他の再起動の試行がすべて失敗したときに、ミラーサーバにプライマリサーバを強制的に引き継がせます。

## 前提条件

-gd データベースサーバオプションで指定される権限 (デフォルトでは、Server Operator システム権限) が必要です。

ミラーサーバのユーティリティデータベースに接続できるように、パートナーサーバが -su データベースオプションを使用して起動されている必要があります。

すでにプライマリサーバを手動で停止して再起動を試し、ALTER DATABASE 文を SET PARTNER FAILOVER 句とともに実行しています。

## コンテキスト

この方法では、ALTER DATABASE 文を FORCE START 句とともに使用しますが、コミットされたトランザクションを損失する可能性があります。

次の条件に当てはまる場合、ALTER DATABASE FORCE START 文だけを使用できます。

- プライマリサーバがダウンしている (たとえば、ハードウェアの障害が原因で)。
- ミラーサーバがプライマリサーバとしての引き継ぎに失敗した。
- ミラーデータベースサーバのユーティリティデータベースに接続できる。

#### 警告

FORCE START 句を使用する場合、ミラーサーバに存在しないトランザクションがプライマリサーバにあると、結果としてトランザクションが失われることがあります。

プライマリを再起動し、SET PARTNER FAILOVER 句を指定して ALTER DATABASE を実行することで、トランザクションを失わないで失敗を強制することをお奨めします。FORCE START 句は、プライマリをどうしても再起動できない場合にのみ、最後の手段として使用してください。

## 手順

1. ミラーデータベースを停止し、可能であればミラーサーバも停止します。
2. プライマリサーバからデータベースおよびトランザクションログファイルを利用できる場合、それらをバックアップし、そのバックアップしたコピーを検証します。それ以外の場合は、トランザクションログをバックアップして検証します。
3. プライマリサーバからのデータベースファイルおよびトランザクションログファイルが有効である場合、両方のファイルをミラーサーバにコピーします。

プライマリサーバからのデータベースファイルは無効ですが、プライマリトランザクションログファイルは有効である場合は、次の手順を実行します。

- a. プライマリログファイルをミラーデータベースのバックアップに適用し、データベースを検証します。
  - b. データベースおよびトランザクションログファイルをミラーサーバにコピーします。
4. ミラーサーバでミラーリングを有効にして、データベースを起動します (または、ミラーサーバを再起動します)。ミラーサーバを再起動する場合、ユーティリティデータベースに接続できるように、-su データベースオプションを指定します。
  5. ミラーサーバ上のユーティリティデータベースの utility\_db に接続します。
  6. ALTER DATABASE FORCE START 文を実行し、ミラーサーバを強制的にプライマリサーバにします。

## 結果

そのミラーサーバが新しいプライマリサーバになります。

#### 例

次の文により、データベース mymirrored.db のミラーサーバが強制的にプライマリサーバになります。

```
ALTER DATABASE mymirrored FORCE START;
```

## 次のステップ

障害の発生したプライマリマシンと別のプライマリマシンを置き換えるには、障害の発生したパートナーを削除し、新しいパートナーを新しいマシンに追加する必要があります。プライマリとミラーの接続文字列を更新する必要があります。

## 関連情報

[ユーザが開始したロールの切り替え \(フェイルオーバー\) \[1682 ページ\]](#)

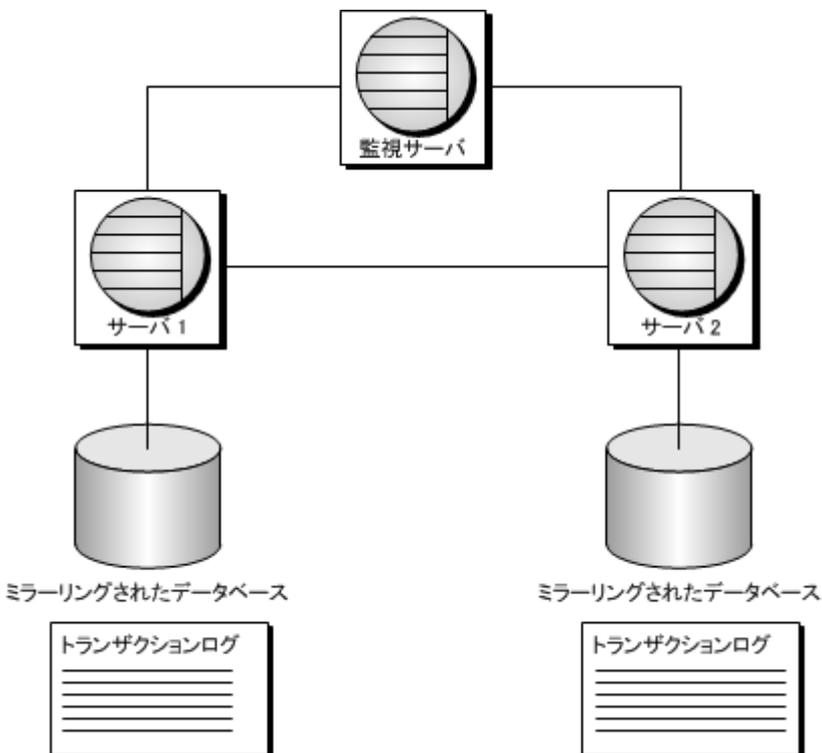
[パートナーサーバの移動 \[1689 ページ\]](#)

[ユーティリティデータベースへの接続 \(接続ウィンドウの場合\) \[298 ページ\]](#)

### 1.10.1.7.4 トラブルシューティング: サーバがデータベースミラーリングシステムで利用できなくなったとき

データベースのミラーリングシステムでサーバが使用できなくなった場合に何が起きるかを理解するのに、いくつかのドキュメント化されたシナリオが役立ちます。

シナリオでは次のデータベースミラーリング構成を使用します。ここで使用するデータベースミラーリングは、同期モードで実行されるサーバ 1、サーバ 2、監視サーバで構成されています。



いつでも MirrorState、PartnerState、および ArbiterState データベースプロパティを使用してミラーリングシステムのデータベースサーバのステータスを判断できます。

プライマリ、ミラー、監視サーバ間の通信は、ミラーリングシステム内で定期的にチェックされます。これら 3 つのサーバ間の通信の中断は、1 つ以上のサーバが利用できなくなることや、サーバ間のネットワークの停止によって引き起こされることがあります。

## シナリオ 1: プライマリサーバを使用できなくなった場合

1. プライマリサーバ (サーバ 1) が使用できなくなります。クライアントはすべて切断されます。
2. 監視サーバとサーバ 2 はサーバ 1 が使用できないことを検知します。
3. 監視サーバとサーバ 2 はクォーラムを形成し、サーバ 2 がプライマリサーバになります。
4. サーバ 2 は、クライアント接続の受け付けを開始します。

このシナリオにおいて、非同期モードまたは非同期フルページモードを使用し、自動フェイルオーバーの実行を指定していない場合、クライアントが再び接続できるようにするには、データベースのコピーを作成し、稼働しているサーバを再起動する必要があります。

## シナリオ 2: プライマリサーバを使用できなくなり、再起動される場合

1. プライマリサーバ (サーバ 1) が使用できなくなります。クライアントはすべて切断されます。
2. 監視サーバとミラーサーバ (サーバ 2) が、プライマリサーバ (サーバ 1) が使用できなくなったことを検知します。
3. 監視サーバとサーバ 2 はクォーラムを形成し、サーバ 2 がプライマリサーバになります。
4. サーバ 2 は、クライアント接続の受け付けを開始します。
5. サーバ 1 が再びオンラインになり、サーバ 2 と監視サーバに再接続します。
6. サーバ 1 がクォーラムを要求しますが、サーバ 2 がすでにプライマリサーバになっています。
7. サーバ 1 はミラーサーバになって、サーバ 2 からの変更を待機します。
8. サーバ 2 は変更をサーバ 1 に送信します。

サーバ 1 がサーバ 2 からのすべてのトランザクションを受信する前に利用できなくなった場合、サーバ 1 は同期ステータスに到達できません。サーバ 2 が再び使用可能になるのを待って、まだ取得していないトランザクションを取得して適用する必要があります。

## シナリオ 3: ミラーサーバを使用できなくなった場合

1. ミラーサーバ (サーバ 2) が使用できなくなります。
2. 監視サーバとサーバ 1 はミラーサーバ (サーバ 2) が使用できないことを検知します。  
クライアント接続は影響を受けません。引き続きプライマリサーバに接続できます。ただし、サーバ 1 または監視サーバのいずれかが利用できなくなった場合、クライアントは接続できません。

## シナリオ 4: ミラーサーバを使用できなくなり、再起動される場合

1. ミラーサーバ (サーバ 2) が使用できなくなります。
2. 可用性は変わらないため、クライアント接続は影響を受けません。引き続きプライマリサーバに接続できます。ただし、サーバ 1 または監視サーバのいずれかが利用できなくなった場合、クライアントは接続できません。
3. サーバ 2 が再びオンラインになり、サーバ 1 と監視サーバに再接続します。
4. サーバ 2 がクォーラムを要求しますが、サーバ 1 がすでにプライマリサーバになっています。
5. サーバ 2 はミラーサーバになって、サーバ 1 からの変更を待機します。
6. サーバ 1 は変更をサーバ 2 に送信します。  
可用性は変わらないため、クライアント接続は影響を受けません。引き続きサーバ 1 に接続します。

## シナリオ 5: 監視サーバを使用できなくなった場合

1. サーバ 1 (プライマリサーバ) とサーバ 2 (ミラーサーバ) が、監視サーバがダウンしたことを検知します。
2. どちらのサーバも引き続き使用できます。クライアント接続は切断されません。  
監視サーバがオンラインに戻ると、サーバ 1 とサーバ 2 はそれを検出して通信を開始します。クライアントにとっては、データベースの可用性に変化はありません。  
監視サーバがない場合にサーバ 1 またはサーバ 2 が利用できなくなると、他のサーバは独自にクォーラムを形成できず、データベースは利用できません。

## シナリオ 6: 監視サーバが再起動する場合

1. 監視サーバが再びオンラインになり、サーバ 1 とサーバ 2 に再接続します。  
可用性は変わらないため、クライアント接続は影響を受けません。

## 関連情報

[トラブルシューティング: プライマリサーバの障害からのリカバリ方法 \[1693 ページ\]](#)

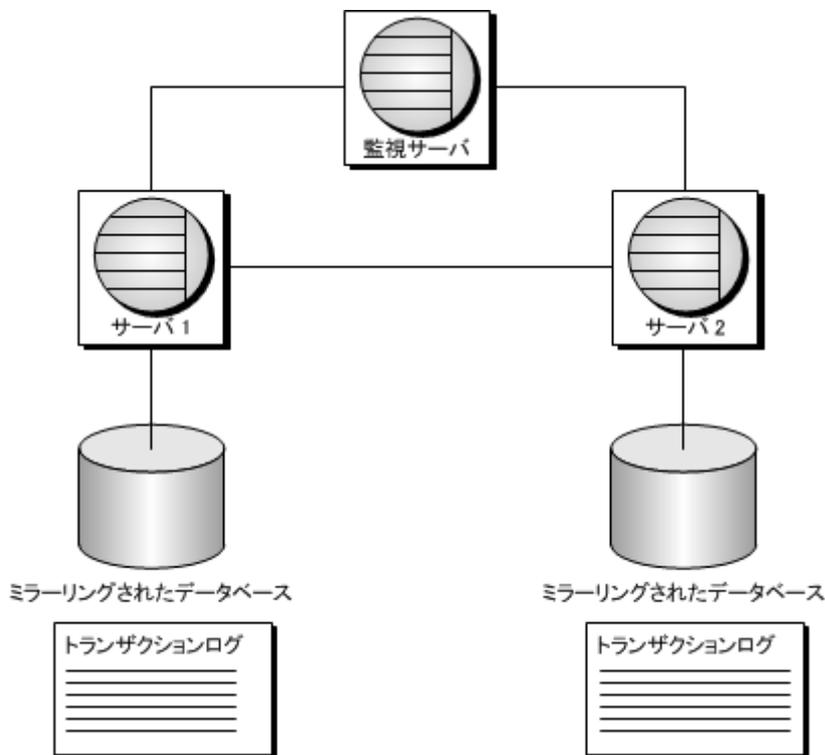
### 1.10.1.7.5 トラブルシューティング: データベースミラーリングで接続が切断される

データベースミラーリングシステムでは、データベースサーバ間の接続が切断されると、考慮が必要なことがいくつかあります。

- 切断されたネットワーク接続を通知するためにサーバに必要な時間は、次の状況によって異なります。

1. 接続がどのように切断されるのか。ネットワークケーブルがコンピュータに差し込まれていない場合、オペレーティングシステムの中にはこの問題を即座に検出して、接続を切断させるものもあります。ネットワークスイッチの障害によって問題が引き起こされた場合は、問題がサーバによって即座に検出される可能性はそれよりも少なくなります。
  2. ミラー接続文字列で指定された活性タイムアウト設定。
- プライマリサーバがミラーリングされたデータベースの所有権を維持するには、少なくとも他のサーバ（ミラーまたは監視サーバ）の1つに接続される必要があります。他のサーバのどちらに対する接続も失われた場合、データベースは再起動され、プライマリサーバは、それらの接続の1つに少なくとも再確立できて、そのステータスを検証できるまで待ちます。
  - 接続の切断を引き起こした問題が修正されたときに、サーバ間の接続は自動的に再確立されます。
  - サーバは、切断した接続を検出するまで、これまで割り当てられていたロールを維持します。この動作は、結果として、これまでミラーサーバであったサーバがプライマリサーバへの切断された接続を検出した後で、プライマリサーバがミラーサーバへの接続が切断されたことを検出した場合、短い期間だけ2つのサーバがプライマリサーバとして動作しようとするようになります。元のプライマリサーバで行われた最初の更新は、サーバが他のサーバにその変更を送信しようとした時点でブロックされます。これによって、この状況が発生したためにトランザクションが失われる可能性が限定されます。元のプライマリは、元のミラーサーバからの切断された接続を検出し、監視サーバのステータスをチェックしたときに、フェイルオーバーが発生していることを確認し、データベースを再起動します。
  - ミラーとして動作しているサーバがデータベースを再起動する必要がある場合、そのサーバ上のデータベースへのアクセスは、プライマリサーバへの接続が再確立されるまで許可されません。
  - ネットワークの障害は、ネットワークのトポロジによってはクライアント接続とミラーサーバ接続の両方に影響する可能性があります。

以下のシナリオは、ミラーリングシステムで接続が切断された場合に何が起こるかを理解するのに役立ちます。ここで使用するデータベースミラーリングは、同期モードで実行されるサーバ1、サーバ2、監視サーバで構成されています。



いつでも MirrorState、PartnerState、および ArbiterState データベースプロパティを使用してミラーリングシステムのデータベースサーバのステータスを判断できます。

## シナリオ 7: プライマリサーバと監視サーバの接続が切断された場合

1. プライマリサーバ (サーバ 1) と監視サーバとの接続が切断されます。
2. サーバ 1 と監視サーバは、互いの接続が失われたことを検出します。
3. プライマリサーバとミラーサーバの両方は利用可能なままです。クライアント接続は切断されません。サーバ 1 は引き続き変更をサーバ 2 に送信します。
  - サーバ 1 が利用不可能になったか、ミラーサーバ (サーバ 2) への接続を失っている場合、サーバ 2 は自身が監視サーバにまだ接続されていることを前提としてプライマリサーバになります。
  - サーバ 1 がサーバ 2 および監視サーバへの接続を失っている場合、サーバ 1 上のデータベースは使用できなくなります。サーバ 1 は、データベースを再起動し、監視サーバとサーバ 2 への再接続が確立するのを待ちます。
  - サーバ 1 とサーバ 2 が、どちらも監視サーバへの接続を失い、その一方で互いへの接続は続いている場合、その動作は監視が利用できなくなった場合のシナリオ 5 で説明したものと同じになります。

## シナリオ 8: ミラーサーバと監視サーバの接続が切断された場合

1. ミラーサーバ (サーバ 2) と監視サーバとの接続が切断されます。
2. サーバ 2 と監視サーバは、互いの接続が失われたことを検出します。
3. プライマリサーバとミラーサーバの両方は利用可能なままです。クライアント接続は切断されません。
4. サーバ 1 は引き続き変更をサーバ 2 に送信します。
5. サーバ 1 とサーバ 2 の両方が監視サーバの再接続を待機します。
  - サーバ 2 が利用不可能になるか、サーバ 1 への接続を失う場合、サーバ 1 へのクライアント接続は、可用性の変更がないため、影響を受けません。サーバ 2 へのクライアント接続は失われます。
  - サーバ 1 とサーバ 2 が、どちらも監視サーバへの接続を失い、その一方で互いへの接続は続いている場合、その動作は監視が利用できなくなった場合のシナリオ 5 で説明したものと同じになります。
6. 監視サーバが再びオンラインになり、サーバ 1 とサーバ 2 は監視サーバに接続します。

## シナリオ 9: プライマリサーバとミラーサーバの接続が切断された場合

1. プライマリサーバ (サーバ 1) とミラーサーバ (サーバ 2) の接続が切断されます。
2. サーバ 1 とサーバ 2 は、接続が失われたことを検出しますが、異なるタイミングで検出する可能性があります。どちらのサーバも監視サーバのステータスを確認してから、監視サーバからの応答に基づいて自身のアクションを実行します。
  - サーバ 1 が、接続の切断を先に検出した場合、監視サーバのステータスを更新して、自身が引き続きプライマリサーバであることと、サーバ 2 のデータベースが最新ではない可能性があることを示します。サーバ 2 は、監視サーバを確認したときに、サーバ 1 が引き続きプライマリであることを理解します。サーバ 2 は自身のデータベースを再起動します。サーバ 2 はサーバ 1 への接続が再確立するのを待ってから、データベースへのアクセスを許可します。

- サーバ 2 が、接続の切断を先に検出した場合、監視サーバのステータスを更新して、フェイルオーバーが発生したと、サーバ 2 がプライマリサーバになったことを示します。サーバ 1 は、監視サーバのステータスを確認して、フェイルオーバーが発生したと、サーバ 2 がプライマリサーバになったことを理解します。サーバ 1 はデータベースを再起動し、サーバ 2 への接続が再確立するのを待ってから、データベースへのアクセスを許可します。

## シナリオ 10: プライマリサーバから、ミラーサーバおよび監視サーバへの接続が切断された場合

1. プライマリサーバ (サーバ 1) とミラーサーバ (サーバ 2) の間の接続が切断し、プライマリサーバ (サーバ 1) と監視サーバとの接続が切断します。
2. サーバ 1 は監視サーバとサーバ 2 が使用できなくなったことを検出します。サーバ 1 はデータベースを再起動して、他のサーバへの再接続が確立するのを待ちます。クライアントはすべて切断されます。
3. 監視サーバとサーバ 2 はサーバ 1 が使用できないことを検出します。
4. 監視サーバとサーバ 2 はクォーラムを形成し、サーバ 2 がプライマリサーバになります。監視サーバとサーバ 2 では、その動作はプライマリサーバが利用できなくなった場合のシナリオ 1 で説明したものと同一になります。
5. サーバ 2 は、クライアント接続の受け付けを開始します。
6. サーバ 1 が再びオンラインになり、サーバ 2 と監視サーバに再接続します。
7. サーバ 1 はミラーサーバになって、サーバ 2 からの変更を待機します。
8. サーバ 2 は変更をサーバ 1 に送信します。

## シナリオ 11: ミラーサーバから、プライマリサーバおよび監視サーバへの接続が切断された場合

1. ミラーサーバ (サーバ 2) およびプライマリサーバ (サーバ 1) の間の接続と、ミラーサーバ (サーバ 2) と監視サーバの接続が切断します。
2. サーバ 2 は監視サーバとサーバ 1 が使用できなくなったことを検出します。サーバ 2 は、データベースを再起動し、プライマリサーバと監視サーバへの再接続が確立するのを待ちます。サーバ 2 へのクライアント接続はすべて切断されます。プライマリサーバへのクライアント接続は影響を受けません。クライアントは引き続きプライマリサーバに接続できます。
3. 監視サーバとサーバ 1 はサーバ 2 が使用できないことを検出します。監視サーバとサーバ 1 では、その動作はミラーサーバが利用できなくなった場合のシナリオ 3 で説明したものと同一になります。
4. サーバ 1 は監視サーバのステータスを更新して、サーバ 2 が最新ではなくなったことを示します。
5. サーバ 2 が再びオンラインになり、サーバ 1 と監視サーバに再接続します。
6. サーバ 2 はミラーサーバのロールを再開し、サーバ 1 からの変更を待機します。
7. サーバ 1 は変更をサーバ 2 に送信します。

## シナリオ 12: 監視サーバから、プライマリサーバおよびミラーサーバへの接続が切断された場合

1. 監視サーバとプライマリサーバとの接続、および監視サーバとミラーサーバ (サーバ 2) との接続が切断します。
2. 監視サーバへの接続が失われた場合、動作は監視が利用できなくなった場合のシナリオ 4 で説明したものと同じです。
3. 接続が復元された場合、動作は監視が再開した場合のシナリオ 4 で説明したものと同じです。

## シナリオ 13: 全サーバ間の接続が切断された場合

1. すべてのサーバ間の接続が切断します。
2. プライマリサーバ (サーバ 1)、ミラーサーバ (サーバ 2)、監視サーバは、互いのサーバへの接続が失われたことを検出します。
3. サーバ 1 は、すべてのクライアントへの接続を切断してデータベースを再起動します。
4. サーバ 2 は、すべてのクライアントへの接続を切断してデータベースを再起動します。
5. サーバ 1 とサーバ 2 は、互いへの再接続を待機して、そのロールを決定します。
6. 接続が再確立されると、サーバ 1 とサーバ 2 は自身のステータスと相手のステータスに基づいて自らのロールを決定します。

### 1.10.1.8 チュートリアル: データベースミラーリングシステムの作成

データベースミラーリングシステムを作成して、フェイルオーバーに対処します。

#### 前提条件

次のシステム権限が必要です。

- MANAGE ANY MIRROR SERVER
- BACKUP DATABASE
- CREATE TABLE
- SERVER OPERATOR

#### コンテキスト

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、運用環境では、各データベースサーバを別々のコンピュータにインストールする必要があります。

異なるコンピュータで実行されているデータベースサーバでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピュータ名に変更する必要があります。

1. [レッスン 1: データベースミラーリングシステムの作成 \[1703 ページ\]](#)  
ミラーリングシステムを作成します。
2. [レッスン 2: データベースミラーリングシステムが正しく実行されていることをテストする \[1706 ページ\]](#)  
データベースおよびデータベースサーバのプロパティを確認し、データベースミラーリングシステムが正しく実行されていることをテストします。
3. [レッスン 3: データベースミラーリングシステムでのロールの切り替え \[1707 ページ\]](#)  
ミラーリングシステムを停止することなく、強制的にサーバのロールを切り替えます。
4. [レッスン 4: データベースミラーリングシステムの停止 \[1709 ページ\]](#)  
ミラーサーバ、プライマリサーバ、監視サーバを停止し、ミラーリングシステムを停止します。
5. [レッスン 5: \(省略可\) ミラーリングシステムの再起動 \[1710 ページ\]](#)  
パートナーサーバと監視サーバを起動して、ミラーリングシステムを起動します。

## 1.10.1.8.1 レッスン 1: データベースミラーリングシステムの作成

ミラーリングシステムを作成します。

### 前提条件

このチュートリアル冒頭の冒頭に一覧されているロールと権限を持っている必要があります。

### 手順

1. 次のディレクトリを作成します。c:\¥server1、c:\¥server2、c:\¥arbiter の 3 つのファイルが含まれています。
2. サンプルデータベースのデータを格納し、トランザクションログを持つ、mirror\_demo.db というデータベースを作成します。トランザクションログを持っていないデータベースをミラーリングモードで起動することはできません。次のコマンドを実行します。

```
newdemo.bat c:\¥server1¥mirror_demo.db
```

3. 1 番目のデータベースサーバを起動します。次のコマンドを実行します。

```
dbsrv17 -n mirror_server1 -x "tcpip(PORT=6871)" -su passwd "c:\¥server1¥mirror_demo.db" -xp on
```

**-n**

データベースサーバに名前を付けます。

**-su**

ユーティリティデータベースにパスワードを指定します。-su オプションを含めてユーティリティデータベースのパスワードを指定することをお奨めします。そうすることで、必要に応じてユーティリティデータベースに接続してデータベースサーバをシャットダウンできるようになります。

-x

データベースサーバが実行されているポートを指定します。

-xp on

データベースサーバがデータベースミラーリングシステムに参加可能であることを示します。

4. Interactive SQL からデータベースに接続します。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server1"
```

5. CREATE MIRROR SERVER 文を使用して、データベースのパートナーサーバと監視サーバを定義します。

次の文を実行して、mirror\_server1 と mirror\_server2 をデータベースミラーリングシステムにおけるパートナーサーバとして定義します。

```
CREATE MIRROR SERVER mirror_server1
AS PARTNER
connection_string='SERVER=mirror_server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1.state';
CREATE MIRROR SERVER mirror_server2
AS PARTNER
connection_string='SERVER=mirror_server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2.state';
```

次の文を実行して定義します

- mirror\_demo\_primary は mirror\_server1 の代替サーバ名として定義し、mirror\_demo\_primary はプライマリサーバとして動作するデータベースサーバに接続するためにクライアントが使用する名前です。
- mirror\_demo\_mirror は mirror\_server2 の代替サーバ名として定義し、mirror\_demo\_mirror はミラーサーバとして動作するデータベースサーバに接続するためにクライアントが使用する名前です。

システムのデータベースサーバを設定するには、プライマリとミラーのロールが必要です。これらのサーバに付けた名前は、クライアントからデータベースサーバに接続するときに代替サーバ名として使用されます。どちらか一方のパートナーサーバは、プライマリサーバまたはミラーサーバとして動作できます。

```
CREATE MIRROR SERVER mirror_demo_primary
AS PRIMARY
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:
6872';
CREATE MIRROR SERVER mirror_demo_mirror
AS MIRROR
connection_string='SERVER=mirror_demo_mirror;HOST=localhost:6871,localhost:6872';
```

次の SQL 文を実行し、データベースミラーリングシステムの監視サーバを定義します。

```
CREATE MIRROR SERVER demo_arbiter
AS ARBITER
connection_string ='SERVER=demo_arbiter;HOST=localhost:6870';
```

6. データベースの認証文字列を設定します。次の文を実行します。

```
SET MIRROR OPTION authentication_string='abc';
```

7. バックアップユーティリティ (dbbackup) を使用して、c:¥¥server1 にあるデータベースファイルとトランザクションログを c:¥¥server2 にコピーします。次のコマンドを実行します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=mirror_demo" c:¥¥server2
```

8. 2 番目のパートナーサーバを起動します。次のコマンドを実行します。

```
dbsrv17 -n mirror_server2 -x "tcpip(PORT=6872;DOBROAD=no)" -su passwd "c:
¥server2¥mirror_demo.db" -xp on
```

9. 監視サーバを起動します。次のコマンドを実行します。

```
dbsrv17 -n demo_arbiter -su passwd -x "tcpip(PORT=6870;DOBROAD=no)" -xf "c:
¥arbiter¥arbiter.state" -xa "AUTH=abc;DBN=mirror_demo"
```

**-xf**

arbiter のステータス情報ファイルのロケーションを指定します。

**-xa**

ミラーリング対象のデータベースの名前と、監視サーバに対する認証文字列 (ここでは abc) を指定します。この認証文字列を、データベースミラーリングシステムのすべてのサーバ (監視サーバ、プライマリサーバ、ミラーサーバ) で使用する必要があります。

10. Interactive SQL を起動し、プライマリサーバの代替サーバ名を使用してプライマリサーバに接続します。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:
6871,localhost:6872"
```

11. データベースにデータを追加します。次の文を実行します。

```
CREATE TABLE test (col1 INTEGER, col2 CHAR(32));
INSERT INTO test VALUES(1, 'Hello from server1');
COMMIT;
```

12. (省略可) 接続先のデータベースサーバの名前を決定するには、ServerName データベースサーバプロパティを使用します。次の文を実行します。

```
SELECT PROPERTY( 'ServerName' );
```

13. すべての Interactive SQL ウィンドウを閉じます。

## 結果

データベースミラーリングシステムが実行されています。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

次のタスク: [レッスン 2: データベースミラーリングシステムが正しく実行されていることをテストする \[1706 ページ\]](#)

## 関連情報

[データベースミラーリング \[1660 ページ\]](#)

[ミラーリングシステムのデータベースサーバの停止 \(dbstop ユーティリティ\) \[1684 ページ\]](#)

[トラブルシューティング: プライマリサーバが起動できない \[1694 ページ\]](#)

[データベースミラーリングシステムの設定 \[1674 ページ\]](#)

[チュートリアル: 監視サーバを共有している複数データベースでデータベースミラーリングシステムを作成する \[1711 ページ\]](#)

### 1.10.1.8.2 レッスン 2: データベースミラーリングシステムが正しく実行されていることをテストする

データベースおよびデータベースサーバのプロパティを確認し、データベースミラーリングシステムが正しく実行されていることをテストします。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

次のコマンドを実行し、各項目を表示します。

- データベースサーバのミラーロール
- 接続先のデータベースサーバの名前
- パートナサーバ、ミラーサーバおよび監視サーバのステータス

```
dbping -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary" -ps ServerName -pd MirrorRole,MirrorState,PartnerState,ArbiterState
```

## 結果

データベースプロパティに関する情報が返されます。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

前のタスク: [レッスン 1: データベースミラーリングシステムの作成 \[1703 ページ\]](#)

次のタスク: [レッスン 3: データベースミラーリングシステムでのロールの切り替え \[1707 ページ\]](#)

## 関連情報

[ServerName \(Server\) 接続パラメータ \[105 ページ\]](#)

[Ping ユーティリティ \(dbping\) \[1140 ページ\]](#)

### 1.10.1.8.3 レッスン 3: データベースミラーリングシステムでのロールの切り替え

ミラーリングシステムを停止することなく、強制的にサーバのロールを切り替えます。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

1. プライマリサーバに接続します。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6872"
```

2. ServerName データベースサーバプロパティを使用し、現在プライマリサーバになっているサーバの名前を判別します。

```
SELECT PROPERTY( 'ServerName' );
```

プライマリサーバの名前 (mirror\_server1) が表示されます。

ServerName データベースサーバプロパティでは、サーバの代替名ではなく実際の名前が返されます。

- ALTER DATABASE SET PARTNER FAILOVER 文を実行してフェイルオーバーを開始します

```
ALTER DATABASE SET PARTNER FAILOVER;
```

プライマリサーバへの接続が閉じられます。server2 のデータベースサーバメッセージウィンドウには、server2 が新しいプライマリサーバであることを示すメッセージが表示されます。

- 現在のプライマリサーバに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6872"
```

- 次の文を実行して、これまでミラーサーバとして機能していたサーバ (mirror\_server2) が、現在はプライマリサーバとして機能していることを確認します。

```
SELECT PROPERTY ( 'ServerName' );
```

- 次の文を実行し、すべてのトランザクションがミラーデータベースにミラーリングされたことを確認します。

```
SELECT * FROM test;
```

- Interactive SQL との接続を切断します。

## 結果

パートナーサーバがロールを切り替えます。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

前のタスク: [レッスン 2: データベースミラーリングシステムが正しく実行されていることをテストする \[1706 ページ\]](#)

次のタスク: [レッスン 4: データベースミラーリングシステムの停止 \[1709 ページ\]](#)

## 関連情報

[トラブルシューティング: データベースミラーリングで接続が切断される \[1698 ページ\]](#)

## 1.10.1.8.4 レッスン 4: データベースミラーリングシステムの停止

ミラーサーバ、プライマリサーバ、監視サーバを停止し、ミラーリングシステムを停止します。

### 前提条件

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

### 手順

1. ミラーサーバ、プライマリサーバ、最後に監視サーバを停止し、データベースミラーリングシステムを停止します。次の順番でコマンドを実行します。

```
dbstop -y -c "UID=DBA;PWD=sql;SERVER=mirror_server2;DBN=utility_db"
```

```
dbstop -y -c "UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=utility_db"
```

```
dbstop -y -c "UID=DBA;PWD=sql;SERVER=demo_arbiter;DBN=utility_db"
```

2. (オプション) `c:¥server1`、`c:¥server2`、および `c:¥arbiter` ディレクトリを削除します。

### 結果

ミラーリングシステムが停止します。

### 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

前のタスク: [レッスン 3: データベースミラーリングシステムでのロールの切り替え \[1707 ページ\]](#)

次のタスク: [レッスン 5: \(省略可\) ミラーリングシステムの再起動 \[1710 ページ\]](#)

## 1.10.1.8.5 レッスン 5: (省略可) ミラーリングシステムの再起動

パートナーサーバと監視サーバを起動して、ミラーリングシステムを起動します。

### 前提条件

ミラーリングシステムのすべてのデータベースサーバを停止する必要があります。

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

### 手順

1. 次のコマンドを実行し、最初のパートナーを再起動します。

```
dbsrv17 -n mirror_server1 -x "tcpip(PORT=6871;DOBROAD=no)" -su passwd "c:
¥server1¥mirror_demo.db" -xp on
```

2. 次のコマンドを実行し、2 番目のパートナーを再起動します。

```
dbsrv17 -n mirror_server2 -x "tcpip(PORT=6872;DOBROAD=no)" -su passwd "c:
¥server2¥mirror_demo.db" -xp on
```

3. 次のコマンドを実行し、監視サーバを再起動します。

```
dbsrv17 -n demo_arbiter -su passwd -x "tcpip(PORT=6870;DOBROAD=no)" -xf "c:
¥arbiter¥arbiter.state" -xa "AUTH=abc;DBN=mirror_demo"
```

### 結果

ミラーリングシステムが再起動します。

タスクの概要: [チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

前のタスク: [レッスン 4: データベースミラーリングシステムの停止 \[1709 ページ\]](#)

### 関連情報

[データベースミラーリングでのクォーラム \[1662 ページ\]](#)

## 1.10.1.9 チュートリアル: 監視サーバを共有している複数データベースでデータベースミラーリングシステムを作成する

プライマリサーバとミラーサーバのそれぞれが、ミラーリングシステムに参加している3つの個別データベースを実行するミラーリング構成を作成します。

### 前提条件

MANAGE ANY MIRROR SERVER、CREATE TABLE、および SERVER OPERATOR のシステム権限が必要です。

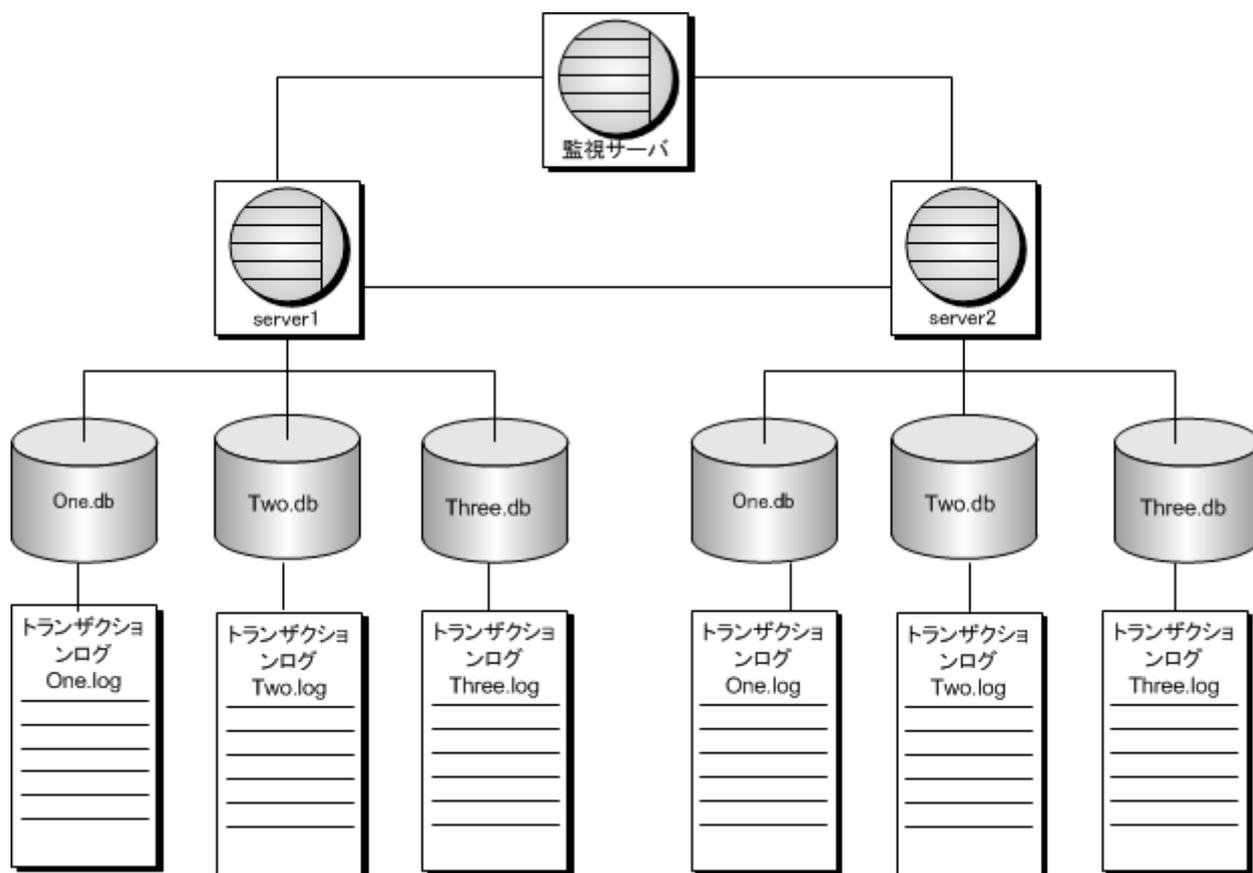
このチュートリアルでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

### コンテキスト

3つのミラーリングシステムはすべて同じ監視サーバと通信します。各ミラーリングシステムでは、CREATE MIRROR SERVER 文の PRIMARY 句と MIRROR 句で指定された一意の代替サーバ名が使用されます。このような構成では、プライマリサーバ、ミラーサーバ、監視サーバを、それぞれ別のコンピュータで実行することができます。

#### 警告

1つのデータベースのトランザクションログファイルは、他のデータベースのトランザクションログファイルと同じディレクトリに格納できません。



プライマリサーバが使用できなくなった場合は、ロールの切り替えが発生し、ミラーサーバがデータベースの所有権を取得し、プライマリサーバになります。クライアントは新しいプライマリサーバへの接続を再確立する必要があります。プライマリサーバへの接続を再確立するために指定する必要があるのは、代替サーバ名だけです。また、この構成には、1つのデータベースに発生した障害からの保護機能もあります。プライマリサーバで実行されているデータベースが使用できなくなると、ロールの切り替えが行われ、障害が発生したデータベースの所有権をミラーサーバが取得します。このデータベースに対してのみ、ミラーサーバがプライマリサーバになります。クライアントは代替サーバ名を使用して、このデータベースのプライマリサーバに対する接続を再確立する必要があります。

`%SQLANYSAMPI7%`¥SQLAnywhere¥DBMirror に、スケールアウトシステムでデータベースミラーリングを使用するサンプルがあります。このサンプルは、1台のコンピュータまたは複数のコンピュータで実行できます。

### i 注記

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、運用環境では、各データベースサーバを別々のコンピュータにインストールする必要があります。

異なるコンピュータで実行されているデータベースサーバでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピュータ名に変更する必要があります。

## 1. レッスン 1: 3つのデータベースと1つの監視サーバで構成されるデータベースミラーリングシステムの作成 [1713 ページ]

同じ監視サーバを使用する3つのミラーリングされたデータベースで構成されるデータベースミラーリングシステムを作成します。

## 2. [レッスン 2: データベースミラーリングシステムの使用とテスト \[1717 ページ\]](#)

フェイルオーバーを開始して、データベースミラーリングシステムの設定をテストします。

### 関連情報

[ミラーリングされたデータベースを実行中のミラーリングシステムに追加する \[1686 ページ\]](#)

## 1.10.1.9.1 レッスン 1: 3 つのデータベースと 1 つの監視サーバで構成されるデータベースミラーリングシステムの作成

同じ監視サーバを使用する 3 つのミラーリングされたデータベースで構成されるデータベースミラーリングシステムを作成します。

### 前提条件

このチュートリアル冒頭に一覧されているロールと権限を持っている必要があります。

### 手順

#### 1. 次のディレクトリを作成します。

- c:¥server1
- c:¥server1¥one
- c:¥server1¥two
- c:¥server1¥three
- c:¥server2
- c:¥server2¥one
- c:¥server2¥two
- c:¥server2¥three
- c:¥arbiter

#### 2. newdemo.bat を使用して、SQL Anywhere17 サンプルデータベース (*demo.db*) を作成します。次のコマンドを実行します。

```
newdemo.bat c:¥server1¥one¥one.db
```

#### 3. 次のコマンドを実行します。

```
newdemo.bat c:¥server1¥two¥two.db
```

4. c:\¥server1 ディレクトリから次のコマンドを実行します。

```
newdemo.bat c:\¥server1¥three¥three.db
```

5. server1 という名前のデータベースサーバを起動します。

```
dbsrv17 -n server1 -x tcpip(PORT=6871) -su passwd c:\¥server1¥one¥one.db -xp on c:\¥server1¥two¥two.db -xp on c:\¥server1¥three¥three.db -xp on
```

-su オプションを含めてユーティリティデータベースのパスワードを指定することをお奨めします。そうすることで、必要に応じてユーティリティデータベースに接続してデータベースサーバをシャットダウンできるようになります。

6. Interactive SQL からデータベース one に接続し、必要なミラーリングオブジェクトを定義します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=server1;DBN=one"
```

- a. server1 をデータベース one のパートナーサーバとして定義します。

```
CREATE MIRROR SERVER server1
AS PARTNER
connection_string='SERVER=server1;host=localhost:6871'
state_file='c:\¥server1¥server1state.txt';
```

- b. データベースサーバ primary\_one をデータベース one の論理プライマリサーバとして定義します。

```
CREATE MIRROR SERVER primary_one
AS PRIMARY
connection_string='SERVER=primary_one;host=localhost:6871,localhost:6872';
```

- c. server2 をデータベース one のパートナーサーバとして定義します。

```
CREATE MIRROR SERVER server2
AS PARTNER
connection_string='SERVER=server2;host=localhost:6872'
state_file='c:\¥server2¥server2state.txt';
```

- d. データベースサーバ mirror\_one をデータベース one の論理ミラーサーバとして定義します。

```
CREATE MIRROR SERVER mirror_one
AS MIRROR
connection_string='SERVER=mirror_one;host=localhost:6871,localhost:6872';
```

- e. 監視サーバを定義します。

```
CREATE MIRROR SERVER arbiter
AS ARBITER
connection_string='SERVER=arbiter;HOST=localhost:6870';
```

- f. データベースミラーリングシステムのミラーリングオプションを設定します。

```
SET MIRROR OPTION authentication_string='abc';
```

- g. Interactive SQL との接続を切断します。

- h. c:\¥server2¥one ディレクトリにデータベースのバックアップコピーを作成します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=server1;DBN=one" c:\¥server2¥one
```

7. Interactive SQL からデータベース two に接続し、必要なミラーリングオブジェクトを定義します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=server1;DBN=two"
```

- a. server1をデータベース two のパートナーサーバとして定義します。

```
CREATE MIRROR SERVER server1
AS PARTNER
connection_string='SERVER=server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1state.txt';
```

- b. データベースサーバ primary\_two をデータベース two の論理プライマリサーバとして定義します。

```
CREATE MIRROR SERVER primary_two
AS PRIMARY
connection_string='SERVER=primary_two;host=localhost:6871,localhost:6872';
```

- c. server2をデータベース two のパートナーサーバとして定義します。

```
CREATE MIRROR SERVER server2
AS PARTNER
connection_string='SERVER=server2;host=localhost:6872'
state_file='c:¥¥server2¥¥server2state.txt';
```

- d. データベースサーバ mirror\_two をデータベース two の論理ミラーサーバとして定義します。

```
CREATE MIRROR SERVER mirror_two
AS MIRROR
connection_string='SERVER=mirror_two;host=localhost:6871,localhost:6872';
```

- e. 監視サーバを定義します。

```
CREATE MIRROR SERVER arbiter
AS ARBITER
connection_string='SERVER=arbiter;HOST=localhost:6870';
```

- f. データベースミラーリングシステムのミラーリングオプションを設定します。

```
SET MIRROR OPTION authentication_string='def';
```

- g. Interactive SQL との接続を切断します。

- h. c:¥server2¥two ディレクトリにデータベースのバックアップコピーを作成します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=server1;DBN=two" c:¥server2¥two
```

8. Interactive SQL からデータベース three に接続し、必要なミラーリングオブジェクトを定義します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=server1;DBN=three"
```

- a. server1をデータベース three のパートナーサーバとして定義します。

```
CREATE MIRROR SERVER server1
AS PARTNER
connection_string='SERVER=server1;host=localhost:6871'
state_file='c:¥¥server1¥¥server1state.txt';
```

- b. データベースサーバ primary\_three をデータベース three の論理プライマリサーバとして定義します。

```
CREATE MIRROR SERVER primary_three
AS PRIMARY
connection_string='SERVER=primary_three;host=localhost:6871,localhost:6872';
```

- c. server2をデータベース three のパートナーサーバとして定義します。

```
CREATE MIRROR SERVER server2
AS PARTNER
```

```
connection_string='SERVER=server2;host=localhost:6872'  
state_file='c:¥¥server2¥¥server2state.txt';
```

- d. データベースサーバ mirror\_three をデータベース three の論理ミラーサーバとして定義します。

```
CREATE MIRROR SERVER mirror_three  
AS MIRROR  
connection_string='SERVER=mirror_three;host=localhost:6871,localhost:6872';
```

- e. 監視サーバを定義します。

```
CREATE MIRROR SERVER arbiter  
AS ARBITER  
connection_string='SERVER=arbiter;HOST=localhost:6870';
```

- f. データベースミラーリングシステムのミラーリングオプションを設定します。

```
SET MIRROR OPTION authentication_string='ghi';
```

- g. Interactive SQL との接続を切断します。

- h. c:¥server2¥three ディレクトリにデータベースのバックアップコピーを作成します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=server1;DBN=three" c:¥server2¥three
```

9. server2 という名前のデータベースサーバを起動します。

```
dbsrv17 -n server2 -x tcpip(PORT=6872) -su passwd c:¥server2¥one¥one.db -xp on c:  
¥server2¥two¥two.db -xp on c:¥server2¥three¥three.db -xp on
```

10. 監視サーバを起動します。

```
dbsrv17 -n arbiter -su passwd -x tcpip(port=6870) -xf c:¥arbiter  
¥arbiterstate.txt -xa "AUTH=abc,def,ghi;DBN=one,two,three"
```

## 結果

server2 を起動した後、server1 のデータベースサーバメッセージウィンドウに、server1 がデータベース one、two、three のミラーリングシステムにおけるプライマリサーバであることが表示されます。また、メッセージウィンドウには、one、two、three のミラーデータベース (パートナー) が server1 に接続されていることも表示されます。

arbiter サーバのメッセージウィンドウには、server1 と server2 の両方が接続されていることが表示されます。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: 監視サーバを共有している複数データベースでデータベースミラーリングシステムを作成する \[1711 ページ\]](#)

次のタスク: [レッスン 2: データベースミラーリングシステムの使用とテスト \[1717 ページ\]](#)

## 1.10.1.9.2 レッスン 2: データベースミラーリングシステムの使用とテスト

フェイルオーバーを開始して、データベースミラーリングシステムの設定をテストします。

### 前提条件

このチュートリアルこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

### 手順

1. 次のコマンドを実行して、Interactive SQL を起動し、プライマリサーバにあるデータベース one に接続します。

```
dbisql -c "UID=DBA;PWD=sql;Server=primary_one;LINKS=TCPIP"
```

2. 次の文を実行して、サンプルデータをデータベースに追加します。

```
CREATE TABLE test (col1 INTEGER, col2 CHAR(32));  
INSERT INTO test VALUES(1, 'Hello from server1');  
COMMIT;
```

3. 次の文を実行し、接続先のデータベースサーバを判別します。

```
SELECT PROPERTY( 'ServerName' );
```

プライマリサーバの名前が出力されます。

4. Interactive SQL との接続を切断します。
5. フェイルオーバーを開始します。次のコマンドを実行します。

```
dbstop -y -c "UID=DBA;PWD=sql;Server=server1"
```

データベースサーバの 1 つの接続が有効であるという警告メッセージが表示された場合は、**はい**をクリックしてシャットダウンします。server2 のデータベースサーバメッセージウィンドウには、server2 が新しいプライマリサーバであることを示すメッセージが表示されます。

6. 次のコマンドを実行して、Interactive SQL を再起動します。

```
dbisql -c "UID=DBA;PWD=sql;Server=primary_one;LINKS=tcPIP"
```

7. 次の文を実行して、server2 に接続されているかどうかを確認します。

```
SELECT PROPERTY ( 'ServerName' );
```

8. 次の文を実行し、すべてのトランザクションがミラーデータベースにコピーされたことを確認します。

```
SELECT * FROM test;
```

9. Interactive SQL から切断し、監視サーバおよび server2 データベースサーバを停止します。

## 結果

データベースミラーリングシステムが正常にフェイルオーバーされ、server2 がプライマリサーバになります。その後、データベースミラーリングシステムが停止します。

## 次のステップ

(オプション) c:¥server1、c:¥server2、および c:¥arbiter ディレクトリを削除します。

タスクの概要: [チュートリアル: 監視サーバを共有している複数データベースでデータベースミラーリングシステムを作成する \[1711 ページ\]](#)

前のタスク: [レッスン 1: 3 つのデータベースと 1 つの監視サーバで構成されるデータベースミラーリングシステムの作成 \[1713 ページ\]](#)

### 1.10.1.10 チュートリアル: 監視サーバの移動

新しいサーバを作成し、新しい監視サーバを使用するように監視サーバ定義を変更して、ミラーリングシステムを停止することなく監視サーバを移動します。

#### 前提条件

このチュートリアルでは、データベースミラーリングシステムの作成に関するチュートリアルで説明しているデータベースミラーリングシステムを利用します。

MANAGE ANY MIRROR SERVER および SERVER OPERATOR のシステム権限が必要です。

#### コンテキスト

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、運用環境では、各データベースサーバを別々のコンピュータにインストールする必要があります。

異なるコンピュータで実行されているデータベースサーバでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピュータ名に変更する必要があります。

## 手順

1. `-su`、`-xa`、および `-xf` の各オプションを使用してサーバを起動し、監視サーバにします。次に例を示します。

```
mkdir arbiter2
```

```
dbsrv17 -n demo_arbiter2 -x "tcpip(port=6873)" -xf c:¥arbiter2¥arbiter2.state -  
xa "AUTH=abc;DBN=mirror_demo" -su passwd
```

オプション	アクション
<code>-su</code>	ユーティリティデータベースにパスワードを指定します。
<code>-xa</code>	監視サーバのデータベース名および認証文字列を指定します。
<code>-xf</code>	ミラーリングシステムのステータス情報ファイルのロケーションを指定します。

2. プライマリサーバに接続し、ミラーリングシステムの監視サーバ定義を変更します。
  - a. 次のコマンドを実行し、プライマリサーバに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:  
6871,localhost:6872"
```

- b. 次のコマンドを実行して、監視サーバ定義を新しいサーバのものに変更します。

```
ALTER MIRROR SERVER demo_arbiter  
AS ARBITER  
connection_string='SERVER=demo_arbiter2;HOST=localhost:6873';
```

プライマリサーバとミラーサーバが、その監視サーバから接続を切断して、新しい監視サーバに接続します。

3. 数秒待ってから、古い監視サーバを停止します。

```
dbstop -y -c "UID=DBA;PWD=sql;SERVER=demo_arbiter;HOST=localhost:  
6870;DBN=utility_db"
```

4. ミラーリングシステムの `ArbiterState` データベースプロパティに対して `ping` を実行し、新しい監視サーバがミラーリングシステムに接続していることを確かめます。次のコマンドを実行します。

```
dbping -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary" -pd ArbiterState
```

`ArbiterState` プロパティの値が接続されます。

```
SQL Anywhere Server Ping Utility Version 17.0.4.1691  
Type          Property          Value  
-----  
Database     ArbiterState     connected
```

## 結果

監視サーバが新しいサーバに移動します。

## 関連情報

[チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

### 1.10.1.11 チュートリアル: パートナサーバの移動

システムを停止しないでデータベースミラーリングシステム内のサーバを別のサーバに移動し、現在のミラーサーバのミラー定義を削除し、新しいサーバを作成し、新しいサーバをミラーリングシステムに追加します。

#### 前提条件

このチュートリアルでは、データベースミラーリングシステムの作成に関するチュートリアルで説明しているデータベースミラーリングシステムを利用します。

MANAGE ANY MIRROR SERVER 権限が必要です。

このタスクでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

#### コンテキスト

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、各データベースサーバが運用環境の別個のコンピュータにインストールされ、接続文字列の localhost への参照が実際のコンピュータ名に変更されている必要があります。

#### 手順

1. 移動するパートナサーバ mirror\_server2 に接続し、パートナサーバにミラーロールがあることを確認します。移動できるのはミラーロールを持つパートナのみです。接続したいサーバがプライマリサーバの場合は、ミラーサーバとプライマリサーバのロールが切り替わるように、フェイルオーバーを開始する必要があります。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server2;HOST=localhost:6872" SELECT DB_PROPERTY( 'MirrorRole' )
```

現在のサーバのミラーリングロールが返されます。

- Mirror が返された場合、mirror\_server2 が現在のミラーです。
- Primary が返された場合、mirror\_server2 が現在のプライマリであり、フェイルオーバーを開始して mirror\_server2 をミラーにする必要があります。次のコマンドを実行し、フェイルオーバーを開始します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server2;HOST=localhost:6871,localhost:6872" ALTER DATABASE SET PARTNER FAILOVER
```

2つのパートナーサーバのロールが切り替わります。

2. 新しいパートナーサーバの新しいディレクトリを作成します。次のコマンドを実行します。

```
mkdir c:¥server3
```

3. プライマリサーバに接続します。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6872"
```

4. DROP MIRROR SERVER 文を実行し、移動するサーバのパートナー定義を削除します。次の文を実行します。

```
DROP MIRROR SERVER mirror_server2;
```

ミラーデータベースが停止します。サーバ上で実行されているデータベースがミラーデータベースのみの場合は、サーバも停止します。

5. 新しいパートナーサーバになるサーバのパートナー定義を新規作成します。次の文を実行します。

```
CREATE MIRROR SERVER mirror_server3 AS PARTNER  
connection_string='SERVER=demo_server3;HOST=localhost:6874'  
state_file='c:¥¥server3¥¥server3.state';
```

6. プライマリとミラーの定義を更新します。次の文を実行します。

```
ALTER MIRROR SERVER mirror_demo_primary AS PRIMARY  
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6874';  
ALTER MIRROR SERVER mirror_demo_mirror AS MIRROR  
connection_string='SERVER=mirror_demo_mirror;HOST=localhost:6871,localhost:6874';
```

7. プライマリサーバからのミラーリングされたデータベースファイルとトランザクションログのコピーを作成し、次のコマンドを実行して c:¥server3 に追加します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=mirror_demo" c:¥server3
```

8. 新しいパートナーサーバがミラーリングシステムに参加するように `-xp on` オプションで起動します。次のコマンドを実行します。

```
dbsrv17 -n mirror_server3 -x "tcpip(PORT=6874)" -su passwd "c:¥server3¥mirror_demo.db" -xp on
```

9. 新しいパートナーサーバに接続し、ミラーサーバであることを確認します。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server3;HOST=localhost:6874" "SELECT DB_PROPERTY('MirrorRole')"
```

このコマンドでは、Mirror が返されます。

## 結果

ミラーシステムが新しいパートナーサーバと共に実行されています。

## 次のステップ

プライマリサーバまたはミラーサーバに接続しているクライアントに更新された接続文字列があることを確認し、Host 接続パラメータで両方のパートナーのアドレスを指定します。

プライマリサーバのデータベースへの接続文字列は次のとおりです。

```
'UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6873';
```

ミラーサーバのデータベースへの接続文字列は次のとおりです。

```
'UID=DBA;PWD=sql;SERVER=mirror_demo_mirror;HOST=localhost:6871,localhost:6873';
```

## 関連情報

[チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

## 1.10.2 SQL Anywhere Veritas Cluster Server エージェント

クラスタとは、一連のアプリケーションを実行するために連携して動くコンピュータ (ノードと呼ばれる) によるグループです。

### i 注記

SQL Anywhere Veritas Cluster Server エージェントには、別途ライセンスが必要です。

クラスタ上で実行されているアプリケーションに接続するクライアントは、クラスタを単一のシステムとして扱います。あるノードに障害が発生すると、クラスタ内の別のノードが、障害のあったノードが提供するサービスを自動的に引き継ぎます。クライアント側では、可用性がわずかに悪化したように見えることがありますが (残りのノードでサービスを再開するまでの時間)、ノードに障害が発生したことまではわかりません。

クラスタリングを使用する場合、データベースまたはデータベースサーバがクラスタ内の別のノードにフェイルオーバーされると、コミットされていないトランザクションはすべて失われます。また、フェイルオーバーが発生した場合は、クライアントはデータベースに再接続する必要があります。

クラスタソフトウェアが任意のアプリケーションを自動フェイルオーバーの対象となる汎用リソースにできる、さまざまなクラスタ環境がサポートされているため、高可用性が実現されます。ただし、フェイルオーバーできるのはデータベースサーバプロセスのみで、監視プロセスや制御プロセスについては制限があります。

ほとんどのクラスタソフトウェアには、特定のアプリケーション用のカスタムリソースを作成するための API が用意されています。Veritas Cluster Server の 2 つのカスタムフェイルオーバーリソースが含まれます。SAServer と SADatabase です。SAServer エージェントはデータベースサーバのフェイルオーバーを担当し、SADatabase エージェントは個々のデータベースファイルのフェイルオーバーを担当します。エージェントは、アプリケーションに応じて、いずれか一方または両方とも使用できます。

SQL Anywhere Veritas Cluster Server エージェントを使用するには、次のようにシステムを設定する必要があります。

- Veritas Cluster Server 4.1 以降を使用する必要があります。

- クラスタ内の各システムノードに SQL Anywhere を同じ構成でインストールします。
- データベースファイルが、クラスタ内のすべてのシステムからアクセスできる共有記憶装置に格納されている必要があります。
- ユーティリティデータベースのパスワードが、クラスタ内のすべてのシステムで同じである必要があります。  
SADatabase エージェントはユーティリティデータベースを使用して、特定のデータベースファイルを起動および停止します。クラスタに属しているすべてのシステムには、同じユーティリティデータベースパスワードを使用する必要があります。データベースサーバの起動時に `-su` データベースサーバオプションを指定して、ユーティリティデータベースのパスワードを設定します。  
UNIX の場合、VCS エージェントは `$$SQLANY17/vcsagent/saserver` にインストールされます。

新しいエージェントを設定して Veritas Cluster Server に追加するには、次の 3 つの方法があります。

1. Cluster Manager を使用します。
2. コマンドラインユーティリティを使用します。
3. テキストエディタを使用して、`main.cf` 設定ファイルを編集します。

これらのマニュアルでは Cluster Manager を使用しています。

使用可能なユーティリティの詳細については、*Veritas Cluster Server Administration Guide* を参照してください。

テキストエディタを使用して `main.cf` を手動で設定するには、`main.cf` ファイルを編集する前にすべての Veritas Cluster Server サービスを停止する必要があります。そうしないと、変更が反映されません。

このセクションの内容:

[SAServer エージェント設定 \[1723 ページ\]](#)

SAServer エージェントを設定する方法がいくつかあります。

[SADatabase エージェントの設定 \[1728 ページ\]](#)

SADatabase エージェントを設定できます。

## 関連情報

[Setting Up SQL Anywhere as a Cluster Database Service \(クラスタデータベースサービスとしての SQL Anywhere の設定\)](#)

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.10.2.1 SAServer エージェント設定

SAServer エージェントを設定する方法がいくつかあります。

このセクションの内容:

[SAServer エージェントの設定 \[1724 ページ\]](#)

SAServer エージェントを設定して、データベースサーバによるクラスタ内の別のノードへのフェイルオーバーを制御できるようにします。

[SAServer エージェントを使用してフェイルオーバーするデータベースサーバの設定 \[1725 ページ\]](#)

Veritas Cluster Server を使用して別のデータベースサーバにフェイルオーバーするようにデータベースを設定します。

[SAServer エージェントのテスト \[1727 ページ\]](#)

Veritas Cluster Server を使用した SAServer エージェントにおけるテストフェイルオーバー

## 1.10.2.1.1 SAServer エージェントの設定

SAServer エージェントを設定して、データベースサーバによるクラスタ内の別のノードへのフェイルオーバーを制御できるようにします。

### 前提条件

SQL Anywhere Veritas Cluster Server エージェントを使用するには、次のようにシステムを設定する必要があります。

- Veritas Cluster Server 4.1 以降を使用する必要があります。
- クラスタ内の各システムノードに SQL Anywhere を同じ構成でインストールします。
- データベースファイルが、クラスタ内のすべてのシステムからアクセスできる共有記憶装置に格納されている必要があります。
- ユーティリティデータベースのパスワードが、クラスタ内のすべてのシステムで同じである必要があります。  
SADatabase エージェントはユーティリティデータベースを使用して、特定のデータベースファイルを起動および停止します。クラスタに属しているすべてのシステムには、同じユーティリティデータベースパスワードを使用する必要があります。ユーティリティデータベースのパスワードを設定するには、データベースサーバの起動時に `-su` サーバオプションを指定します。
- UNIX の場合、VCS エージェントは `$$SQLANY17/vcsagent/saserver` にインストールされます。

### コンテキスト

SAServer エージェントは、データベースサーバによるクラスタ内の別のノードへのフェイルオーバーを制御します。

### 手順

1. クラスタの各ノードで実行されているデータベースサーバをすべてシャットダウンします。
2. クラスタからノードを選択して、SAServer というディレクトリをそのノードの `%VCS_HOME%\bin` ディレクトリの下に作成します。このフォルダ内には他の Veritas Cluster Server エージェント (NIC や IP) が存在することがあります。
3. 次のファイルを `%SQLANY17%\VCSAgent\SAServer` ディレクトリから手順 2 で作成した SAServer ディレクトリにコピーします。

- Online.pl
  - Offline.pl
  - Monitor.pl
  - Clean.pl
  - SAServer.xml
4. %VCS\_HOME%\bin\VCSdefault.dll ファイルを %VCS\_HOME%\bin\SAServer ディレクトリにコピーして、SAServer.dll に名前を変更します。
  5. %SQLANY17%\VCSAgent\SAServer\SAServerTypes.cf ファイルを %VCS\_HOME%\conf\config ディレクトリにコピーします。
  6. クラスタ内のその他すべてのノードについて、手順 1～5 を繰り返します。
  7. Veritas Cluster Server Manager を起動し、ユーザ名とパスワードを入力してクラスタに接続します。
  8. 次の手順で SAServer エージェントを追加します。
    - a. **File > Import Types** をクリックします。
    - b. %VCS\_HOME%\conf\config\SAServerTypes.cf に移動して、**インポート** をクリックします。

## 結果

SAServer エージェントが設定されます。

## 関連情報

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.10.2.1.2 SAServer エージェントを使用してフェイルオーバーするデータベースサーバの設定

Veritas Cluster Server を使用して別のデータベースサーバにフェイルオーバーするようにデータベースを設定します。

## 前提条件

SQL Anywhere Veritas Cluster Server エージェントを使用するには、次のようにシステムを設定する必要があります。

- Veritas Cluster Server 4.1 以降を使用する必要があります。
- クラスタ内の各システムノードに SQL Anywhere を同じ構成でインストールします。
- データベースファイルが、クラスタ内のすべてのシステムからアクセスできる共有記憶装置に格納されている必要があります。

- ユーティリティデータベースのパスワードが、クラスタ内のすべてのシステムで同じである必要があります。SADatabase エージェントはユーティリティデータベースを使用して、特定のデータベースファイルを起動および停止します。クラスタに属しているすべてのシステムには、同じユーティリティデータベースパスワードを使用する必要があります。ユーティリティデータベースのパスワードを設定するには、データベースサーバの起動時に `-su` サーバオプションを指定します。
- UNIX の場合、VCS エージェントは `$SQLANY17/vcsagent/saserver` にインストールされます。

ネットワークサーバを停止する場合は、サーバを停止するパーミッションのあるユーザ ID を接続文字列に指定します。デフォルトでは、ネットワークサーバを停止するには SERVER OPERATOR システム権限が必要です。-gk サーバコマンドラインオプションを使用するとデフォルトの動作を変更できます。

## 手順

1. Veritas Cluster Server Manager を起動し、ユーザ名とパスワードを入力して接続します。
2. 次の手順で、SAServer をリソースとしてサービスグループに追加します。
  - a. **Edit > Add > Resource** をクリックします。
  - b. **Resource Type** リストから、**SAServer** をクリックします。

Windows では、**Resource Type** リストで Windows の下に SAServer が表示されない場合、`SAServer.xml` ファイルを `%VCS_ROOT%\cluster manager\attrpool\Win2K\400` に追加して、クラスタサービスを再起動する必要があります。

- c. **Resource Name** フィールドに名前を入力します。
- d. 以下の属性に次のように属性値を追加します。

### cmdStart

```
dbsrv17-x tcpip database-file-on-shared-disk -n server-name
```

### cmdMonitor

```
dbping -c "Server=server-name"
```

### cmdStop

```
dbstop -c user-id,password -y
```

- e. **Enabled** をクリックします。

これにより、リソースの使用準備が整ったことを示します。

- f. **OK** をクリックします。
3. リソースの依存性が適切に設定されていることを確認します。共有ディスクリソースや IP アドレスリソースなど、SAServer を起動するために起動してグループ化しておく必要があるリソースが他にも存在します。
  4. サービスグループを右クリックして、**Online > node-name** をクリックします。ここで、`node-name` はリソースの実行に使用する、クラスタ内のコンピュータの名前です。

## 結果

これで、サービスグループがオンラインになります。

## 関連情報

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.10.2.1.3 SAServer エージェントのテスト

Veritas Cluster Server を使用した SAServer エージェントにおけるテストフェイルオーバー

#### 前提条件

SQL Anywhere Veritas Cluster Server エージェントを使用するには、次のようにシステムを設定する必要があります。

- Veritas Cluster Server 4.1 以降を使用する必要があります。
- クラスタ内の各システムノードに SQL Anywhere を同じ構成でインストールします。
- データベースファイルが、クラスタ内のすべてのシステムからアクセスできる共有記憶装置に格納されている必要があります。
- SADatabase エージェントはユーティリティデータベースを使用して、特定のデータベースファイルを起動および停止します。クラスタに属しているすべてのシステムには、同じユーティリティデータベースパスワードを使用する必要があります。ユーティリティデータベースのパスワードを設定するには、データベースサーバの起動時に `-su` サーバオプションを指定します。
- UNIX の場合、VCS エージェントは `$$SQLANY17/vcsagent/saserver` にインストールされます。

選択元となるテーブルに対する SELECT object-level 権限、または SELECT ANY TABLE システム権限が必要です。

#### 手順

1. Interactive SQL からデータベースに接続します。次に例を示します。

```
dbisql -c "UID=DBA;PWD=passwd;Server=VCS;HOST=HostName"
```

2. 次のクエリを実行します。

```
SELECT * FROM Departments;
```

このクエリはエラーなく実行されます。

3. データベースサーバを実行しているシステムをシャットダウンします。
4. 同じ接続文字列を使用して Interactive SQL から再接続し、クエリを再度実行します。接続もクエリの実行も正常に行われます。

## 結果

フェイルオーバーが行われ、すべてのリソースが代替サーバで起動されます。

## 関連情報

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.10.2.2 SADBatabase エージェントの設定

SADBatabase エージェントを設定できます。

このセクションの内容:

[SADBatabase エージェントの設定 \[1728 ページ\]](#)

SADBatabase エージェントを設定して、データベースによるクラスタ内の別のノードへのフェイルオーバーを制御します。

[SADBatabase エージェントを使用してフェイルオーバーするデータベースの設定 \[1730 ページ\]](#)

フェイルオーバーの設定を行う前に、データベースをサービスグループに追加します。

[SADBatabase エージェントのテスト \[1731 ページ\]](#)

Interactive SQL を使用して SADBatabase エージェントのフェイルオーバーをテストします。

#### 1.10.2.2.1 SADBatabase エージェントの設定

SADBatabase エージェントを設定して、データベースによるクラスタ内の別のノードへのフェイルオーバーを制御します。

### 前提条件

SQL Anywhere Veritas Cluster Server エージェントを使用するには、次のようにシステムを設定する必要があります。

- Veritas Cluster Server 4.1 以降を使用する必要があります。
- クラスタ内の各システムノードに SQL Anywhere を同じ構成でインストールします。
- データベースファイルが、クラスタ内のすべてのシステムからアクセスできる共有記憶装置に格納されている必要があります。
- ユーティリティデータベースのパスワードが、クラスタ内のすべてのシステムで同じである必要があります。  
SADBatabase エージェントはユーティリティデータベースを使用して、特定のデータベースファイルを起動および停止します。クラスタに属しているすべてのシステムには、同じユーティリティデータベースパスワードを使用する必要があります。ユーティリティデータベースのパスワードを設定するには、データベースサーバの起動時に `-su` サーバオプションを指定します。

- UNIX の場合、VCS エージェントは `$$SQLANY17/vcsagent/saserver` にインストールされます。

データベースを停止するには、`offline.pl` ファイルにデフォルトで SERVER OPERATOR システム権限が必要です。`-gk` サーバコマンドラインオプションを使用するとデフォルトの動作を変更できます。

ネットワークサーバでデータベースを起動するには、`online.pl` ファイルにデフォルトで SERVER OPERATOR システム権限が必要です。`-gk` サーバコマンドラインオプションを使用するとデフォルトの動作を変更できます。

## 手順

1. クラスタの各ノードで実行されているデータベースサーバをすべてシャットダウンします。
2. クラスタ内のいずれか 1 つのノードに `%VCS_HOME%\bin\SADatabase` というディレクトリを作成します。
3. 次のファイルを `%SQLANY17%\VCSAgent\SADatabase` ディレクトリから手順 2 で作成した `%VCS_HOME%\bin\SADatabase` ディレクトリにコピーします。
  - `Online.pl`
  - `Offline.pl`
  - `Monitor.pl`
  - `Clean.pl`
  - `SADatabase.xml`
4. `%VCS_HOME%\bin\VCSdefault.dll` ファイルを `%VCS_HOME%\bin\SADatabase` ディレクトリにコピーして、`SADatabase.dll` に名前を変更します。
5. `%SQLANY17%\VCSAgent\SADatabase\SADatabaseTypes.cf` ファイルを `%VCS_HOME%\conf\config` ディレクトリにコピーします。
6. クラスタに属するその他すべてのシステムについて、手順 1 ~ 5 を繰り返します。
7. Veritas Cluster Server Manager を起動し、ユーザ名とパスワードを入力してクラスタに接続します。
8. 次の手順で SADatabase エージェントを追加します。
  - a. **File > Import Types** をクリックします。
  - b. `%VCS_HOME%\conf\config` に移動して、**インポート** をクリックします。

## 結果

SADatabase エージェントが設定されます。

## 関連情報

[-su データベースサーバオプション \[480 ページ\]](#)

## 1.10.2.2 SADBatabase エージェントを使用してフェイルオーバーするデータベースの設定

フェイルオーバーの設定を行う前に、データベースをサービスグループに追加します。

### 前提条件

SQL Anywhere Veritas Cluster Server エージェントを使用するには、次のようにシステムを設定する必要があります。

- Veritas Cluster Server 4.1 以降を使用する必要があります。
- クラスタ内の各システムノードに SQL Anywhere を同じ構成でインストールします。
- データベースファイルが、クラスタ内のすべてのシステムからアクセスできる共有記憶装置に格納されている必要があります。
- ユーティリティデータベースのパスワードが、クラスタ内のすべてのシステムで同じである必要があります。  
SADBatabase エージェントはユーティリティデータベースを使用して、特定のデータベースファイルを起動および停止します。クラスタに属しているすべてのシステムには、同じユーティリティデータベースパスワードを使用する必要があります。ユーティリティデータベースのパスワードを設定するには、データベースサーバの起動時に `-su` サーバオプションを指定します。
- UNIX の場合、VCS エージェントは `$SQLANY17/vcsagent/saserver` にインストールされます。

### 手順

1. 次の手順で、SADBatabase をリソースとしてサービスグループに追加します。

- a. **Edit** > **Add** > **Resource** をクリックします。
- b. **Resource Type** リストから、**SADBatabase** をクリックします。

Windows では、**Resource Type** リストで Windows の下に SADBatabase が表示されない場合、SADBatabase.xml ファイルを `%VCS_ROOT%\cluster manager\attrpool\Win2K¥400` に追加して、クラスタサービスを再起動する必要があります。

- c. **Resource Name** フィールドに名前を入力します。
- d. 以下の各属性について、**Edit** 列のボタンをクリックして、次のように属性値を追加します。

#### DatabaseFile

データベースファイルのロケーション。たとえば、`E:\demo.db`。

#### DatabaseName

データベースの名前。

#### ServerName

データベースサーバの名前。クラスタ内の各システムには異なるサーバ名を指定できます。属性の範囲は、Global ではなく、Per System にします。

#### UtilDBpwd

クラスタ内のすべてのシステムで使用されるユーティリティデータベースのパスワード。

e. *Enabled* をクリックします。

これにより、リソースの使用準備が整ったことを示します。

f. *OK* をクリックします。

- リソースの依存性が適切に設定されていることを確認します。共有ディスクリソースや IP アドレスリソースなど、SADatabase を起動するために起動/グループ化しておく必要があるリソースが他にも存在します。
- サービスグループを右クリックして、▶ *Online* ▶ *node-name* ▶ をクリックします。ここで、*node-name* はリソースの実行に使用する、クラスタ内のコンピュータの名前です。

## 結果

これで、サービスグループがオンラインになります。

## 関連情報

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.10.2.2.3 SADatabase エージェントのテスト

Interactive SQL を使用して SADatabase エージェントのフェイルオーバーをテストします。

## 前提条件

SQL Anywhere Veritas Cluster Server エージェントを使用するには、次のようにシステムを設定する必要があります。

- Veritas Cluster Server 4.1 以降を使用する必要があります。
- クラスタ内の各システムノードに SQL Anywhere を同じ構成でインストールします。
- データベースファイルが、クラスタ内のすべてのシステムからアクセスできる共有記憶装置に格納されている必要があります。
- ユーティリティデータベースのパスワードが、クラスタ内のすべてのシステムで同じである必要があります。  
SADatabase エージェントはユーティリティデータベースを使用して、特定のデータベースファイルを起動および停止します。クラスタに属しているすべてのシステムには、同じユーティリティデータベースパスワードを使用する必要があります。ユーティリティデータベースのパスワードを設定するには、データベースサーバの起動時に `-su` サーバオプションを指定します。
- UNIX の場合、VCS エージェントは `$SQLANY17/vcsagent/saserver` にインストールされます。

選択元となるテーブルに対する `SELECT object-level` 権限、または `SELECT ANY TABLE` システム権限が必要です。

ネットワークサーバ上のデータベースを停止するのに必要な権限は、データベースサーバの `-gd` オプションによって決まります。ネットワークサーバ上のデータベースを停止するためのデフォルトのシステム権限は `SERVER OPERATOR` です。

## 手順

1. Interactive SQL からデータベースに接続します。次に例を示します。

```
dbisql -c "UID=DBA;PWD=sql;Server=VCS;HOST=HostName"
```

2. 次のクエリを実行します。

```
SELECT * FROM Departments;
```

このクエリはエラーなく実行されます。

3. データベースに障害が発生し、第 1 システムノードで実行されているデータベースサーバがデータベースファイルにアクセスできないとします。これにより、第 2 システムノードで起動されたデータベースサーバへのデータベースファイルのフェイルオーバーが発生します。次のコマンドに類似のコマンドを発行して、最初のノードのデータベースファイルのフェイルオーバーを行うことができます。

```
dbisql -q -c "UID=DBA;PWD=sql;Server=VCS1;DBN=utility_db" STOP DATABASE DEMO ON VCS1 UNCONDITIONALLY;
```

## 結果

第 1 コンピュータ上のデータベースファイルに障害が発生します。Veritas Cluster Server がこのファイルに障害が発生したことを認識するまでには遅延があります。これは、Veritas Cluster Server がリソースの状態を監視する間隔がデフォルトでは 60 秒ごとになっているからです (この間隔は、リソース設定で短くできます)。その後、データベースファイルは 2 番目のコンピュータにフェイルオーバーし、そのデータベースファイルは、元のデータベースサーバとは異なる名前を持つ可能性のある 2 番目のコンピュータ上のデータベースサーバを使用して起動されます。

たとえば、新しいデータベースサーバが VCS2 の場合、クライアントでは次のようにして接続文字列にこの新しいデータベースサーバ名を指定する必要があります。

```
"UID=DBA;PWD=sql;Server=VCS2;DBN=DEMO;HOST=HostName"
```

## 次のステップ

Interactive SQL から再接続します。接続もクエリの実行も正常に行われます。

## 関連情報

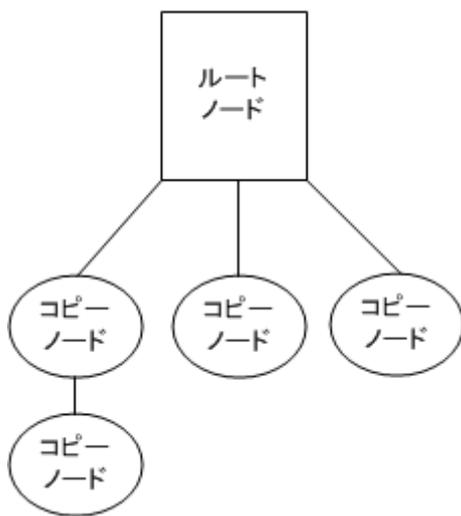
[-su データベースサーバオプション \[480 ページ\]](#)

### 1.10.3 読み込み専用スケールアウト

読み込み専用スケールアウトによって、データベースへの読み込み専用アクセスを必要とするレポートなどの操作の負荷を軽減できます。

スケールアウトツリーは、ルートノードとコピーノードで構成されます。

- スケールアウトシステムのルートノードには、書き込み可能なデータベースの唯一のコピーがあります。ルートノードには、1つのデータベースまたはデータベースミラーリングシステムを指定できます。ルートノードの下に、コピーノードの1つまたは多数の分岐を指定できます。
- コピーノードは、読み込み専用アクセス用にデータベースのコピーを実行するデータベースサーバであり、コピーノードに他のコピーノードの親を指定できます。



読み込み専用スケールアウトには別途ライセンスが必要です。

このセクションの内容:

#### [スケールアウトの動作 \[1734 ページ\]](#)

データベースの可用性を保証するために、スケールアウトとデータベースミラーリングを一緒に使用することができます。

#### [データベースミラーリングシステムの読み込み専用スケールアウトシステム \[1735 ページ\]](#)

データベースミラーリングシステムでは、読み込み専用の接続を使用して、ミラーサーバ上で実行されているデータベースにアクセスできます。

#### [Relay サーバおよび Outbound Enabler を使用する、Web サービスの高可用性とスケールアウトソリューション \[1736 ページ\]](#)

次の SQL Anywhere コンポーネントは、Web アプリケーションをホストするための高可用性やスケールアウトソリューションの設定に必要です。

#### [読み込み専用スケールアウトシステムにおけるデータベースへの接続 \[1740 ページ\]](#)

スケールアウトシステムを設定するには、ルートノードとして動作するデータベースサーバに対して CREATE MIRROR SERVER 文で 2 つの定義を作成します。

#### [読み込み専用スケールアウト設定 \[1742 ページ\]](#)

スケールアウトシステムを設定するために、実行する必要があるステップがいくつかあります。

[読み込み専用のスケールアウトシステムを保守するには、次の手順に従います。 \[1748 ページ\]](#)

SQL Anywhere モニタを使用してプライマリデータベースをモニタリングすることによって、スケールアウトシステムのデータベースサーバのステータスを確認できます。

[チュートリアル: 読み込み専用のスケールアウトシステムの作成 \[1753 ページ\]](#)

コピーノードを自動的に追加するルートデータベースサーバを設定してモニタします。

[チュートリアル: 1つのサーバをコピーノードおよび監視サーバとして使用 \[1759 ページ\]](#)

サーバを監視サーバおよびコピーノードとして実行するように設定します。

[チュートリアル: パートナサーバのコピーノードへの変換 \[1761 ページ\]](#)

データベースミラーリングシステムのパートナサーバを、システムを停止することなく、読み込み専用のスケールアウトシステムのコピーノードに変換します。

[チュートリアル: 読み込み専用のスケールアウトシステムへのミラーリングシステムの追加 \[1763 ページ\]](#)

ミラーリングシステムを既存の読み込み専用のスケールアウトシステムへ追加します。読み込み専用のスケールアウトシステムのルートも、ミラーサーバになります。

## 関連情報

[データベースの正常性と統計情報 \[1058 ページ\]](#)

[レッスン 8: \(省略可\) SQL Anywhere モニタからの読み込み専用スケールアウトシステムのモニタリング \[1298 ページ\]](#)

### 1.10.3.1 スケールアウトの動作

データベースの可用性を保証するために、スケールアウトとデータベースミラーリングを一緒に使用することができます。

ルートノードは、読み込み要求と書き込み要求の両方を受け入れる唯一のデータベースサーバです。追加のコピーノードを起動すると、ルートデータベースサーバは、子として定義されたコピーノードにトランザクションログページを送信します (コピーノードが接続済みでトランザクションログページを受信できる場合)。通常、ページは応答を待たずに送信されますが、処理しきれない量の非同期要求がコピーノードで受信されないようにするため、データベースサーバから確認が要求されることもあります。

コピーノードは、ページを受信するとそれをディスクに書き込み、子ノード (存在する場合) に送信します。

子ノードが使用できなくなったことを親ノードが検出すると、子へのトランザクションログページの送信が停止します。子が再起動されると、受信されていないトランザクションログページが要求され、親から子へのトランザクションログページの送信が再開します。子はコピーノードのステータス変更を親に通知し、ステータス情報はツリーをたどって最終的にルートデータベースサーバに到達します。

ルートデータベースサーバが使用できなくなっても、スケールアウトシステム内のすべての子は引き続き動作しますが、プライマリデータベースサーバからは更新を受信しなくなります。コピーノード接続では、古くなったデータが取得される可能性があります。ルートデータベースサーバが再び使用可能になると、子は接続を再確立し、トランザクションログページの受信を再開します。

コピーノードで、互換性がない、トランザクションログが一致しないなどの問題が発生すると、データベースは停止します。問題が発生したデータベースを実行していたデータベースサーバも、他に実行中のデータベースがないかぎりシャットダウンします。

## 関連情報

[データベースミラーリングシステムの読み込み専用スケールアウトシステム \[1735 ページ\]](#)

[データベースの正常性と統計情報 \[1058 ページ\]](#)

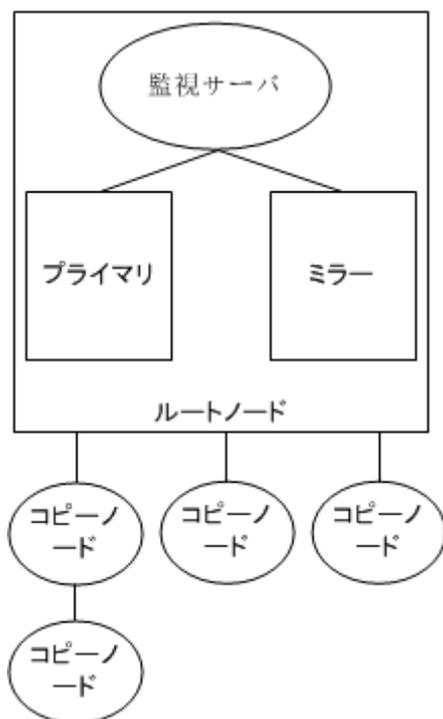
[レッスン 8: \(省略可\) SQL Anywhere モニタからの読み込み専用スケールアウトシステムのモニタリング \[1298 ページ\]](#)

### 1.10.3.2 データベースミラーリングシステムの読み込み専用スケールアウトシステム

データベースミラーリングシステムでは、読み込み専用の接続を使用して、ミラーサーバ上で実行されているデータベースにアクセスできます。

この機能は、このデータベースへの読み込み専用アクセスを必要とするレポートなどの操作の負荷を軽減したい場合に役立ちます。ミラーサーバとコピーノードの唯一の違いは、コピーノードがプライマリサーバまたはミラーサーバとしてデータベースミラーリングシステムに加わることができない点です。コピーノードは、それがコピーするデータベースの監視サーバとして動作することができます。

読み込み専用スケールアウトでデータベースミラーリングを使用することで、ルートノードの可用性を保証できます。データベースミラーリングでスケールアウトを使用する場合、スケールアウトシステムのルートノードは、単一のデータベースサーバで構成されるのではなく、プライマリサーバ、ミラーサーバ、監視サーバで論理的に構成されます。



ミラーリングシステムでデータベースサーバを起動すると、参加しているミラーサーバに他のデータベースサーバからアクセスして、データベースの読み込み専用コピーを管理できます。必要に応じて、コピーノードの分岐をいくつでもミラーサーバに追加します。

## コピーノードとプライマリサーバ/ミラーサーバの違い

データベースミラーリングシステムでコピーノードを使用する場合、現在のプライマリサーバ、現在のミラーサーバ、または別のコピーノードの子としてコピーノードを指定できます。コピーノードは、その親からトランザクションログページを取得します。ミラーノードと異なり、コピーノードにはステータス情報ファイルはありません。これは、どのデータベースサーバにデータベースの所有権を与えるかについて、コピーノードのステータスが影響しないためです。コピーノードは、コピーしているデータベースの監視サーバとしても動作することができます。コピーノードが監視サーバとして動作する場合、高可用性システムのいずれのデータベースサーバ名とも異なる監視ミラーサーバ名を持つ必要があります。

## 関連情報

[データベースミラーリング \[1660 ページ\]](#)

[Relay サーバおよび Outbound Enabler を使用する、Web サービスの高可用性とスケールアウトソリューション \[1736 ページ\]](#)

[監視サーバとしてコピーノードを使用する \[1678 ページ\]](#)

### 1.10.3.3 Relay サーバおよび Outbound Enabler を使用する、Web サービスの高可用性とスケールアウトソリューション

次の SQL Anywhere コンポーネントは、Web アプリケーションをホストするための高可用性やスケールアウトソリューションの設定に必要です。

#### Relay Server (RS)

Relay Server は、HTTP 要求のエントリポイント (通常は Internet から) であり、仮想ホストと負荷分散機能を提供します。Relay Server は、サーバファームという仮想エンティティをターゲットとする HTTP 要求を受け入れ、ルート指定します。各サーバファームは、1 つ以上のデータベース/Web サーバで構成されています。Relay Server は、処理用に適したバックエンドサーバにすべての HTTP 要求をリレーする HTTP/WEB アクセスポイントです。

#### Outbound Enabler (OE)

Outbound Enabler は、Relay Server とバックエンドデータベースサーバ間のブリッジの役割を果たします。サーバファーム内にあるバックエンドサーバごとに、1 つの Outbound Enabler プロセスがあります。その目的は、指定されたバックエンドサーバが稼働していることを確認し、Outbound Enabler が Relay Server と間の通信チャネルを確立して要求をバックエンドサーバに転送することにあります。Outbound Enabler は Relay Server への接続を開始するため、ファイアウォール経由で LAN から DMZ にある Relay Server にアウトバウンド接続を簡単に行うことができます。

#### SQL Anywhere (バックエンド) サーバ

SQL Anywhere (バックエンド) サーバは、完全に SQL で作成された Web アプリケーションをホストできる、高度なデータベースであり Web サーバです。サーバは、高可用性とスケールアウトソリューションを必要とする、負荷の高い環境を緩和するために配備できます。高可用性により、サービスの中断から保護するフォールトトレランス機能が提供され、スケールアウトにより、負荷のピーク時にスループットを向上するための負荷分散が行われます。

## 高可用性

高可用性を提供するには、3つのサーバ(1つの監視サーバと2つのパートナーサーバ)が必要です。1つのパートナーサーバにはプライマリロールが割り当てられ、もう1つにはミラーロールが割り当てられます。グローバルテーブルへの挿入などの書き込み操作はプライマリサーバでのみ許可されますが、選択などの読み込み操作はどちらのタイプのサーバでも許可されます。プライマリサーバは、更新を受信するたびにミラーサーバを同期します。監視サーバは、他の2つのサーバのどちらがデータベースの所有権を持つのかを判断する必要があるときに使用されます。

Relay Server 経由ですべての HTTP 要求をルート指定することにより、Web サービス環境内で高可用性を活用できます。Relay Server では、HTTP 要求の URL を処理して、ターゲットサーバファームが抽出されます。わかりやすくするため、HTTP 要求を readFarm または writeFarm のどちらかにルート指定する Relay Server 設定について考えてみます。プライマリバックエンドサーバでは読み込み要求と書き込み要求の両方を処理できますが、サーバ障害が発生したときに読み込み機能を処理する他の読み込み専用コピーノードがある場合には書き込み要求のみを処理するよう設定し、そうでない場合には読み込み要求と書き込み要求の両方を処理するように設定できます。ただし、ミラーサーバで読み込み要求を処理して読み込み負荷を分散したり、ミラーサーバでのハードウェア使用を削減するために要求をいっさい処理しないように設定したりできます。クライアントアプリケーションでは、読み込み要求と書き込み要求用にアクセスする物理サーバを認識する必要はなく、要求を readFarm または writeFarm のどちらかにダイレクトするように URL を構成します。

Relay Server に加え、パートナーである各バックエンドサーバには、2つの Outbound Enabler が必要です(バックエンドサーバが参加している各ファームに1つずつ)。2つの Outbound Enabler は、プライマリパートナーへの通信を制御し、ミラーパートナーとの通信を確立します。この設定では、プライマリサーバは常に1つしか存在しません。プライマリサーバは、writeFarm に割り当てられた Outbound Enabler をアクティブにし、コピーノード以外のメンバーが readFarm に存在する場合は、オプションとして readFarm に関連付けられている Outbound Enabler を非アクティブにします。ミラーサーバは、writeFarm に関連付けられている Outbound Enabler を非アクティブにし、オプションとして readFarm に関連付けられている Outbound Enabler をアクティブにします。

ハードウェアやソフトウェアの障害によりプライマリサーバが使用できなくなった場合、ミラーサーバが監視サーバとネゴシエートしてデータベースのロールを引き受け、これにより writeFarm のメンバーとなります。また、他の読み込み専用コピーノードがシステムに存在しない場合は、readFarm のメンバーにもなります。障害が発生すると、元のプライマリサーバが writeFarm と readFarm のメンバーである場合は、その両方への参加が突然解消されます。元のプライマリサーバは、障害の原因が究明するまで回復しないことがあります。回復した場合にはミラーロールが想定され、オプションとして readFarm に加わります。サーバの突然の障害により、そのサーバで処理されていた要求が失敗することがあります。ミラーサーバからの writeFarm の回復の遅延は、ミラーのロール切り替えの遅延と、ミラーの writeFarm Outbound Enabler で指定されたステータスポーリング間隔によって異なります。このステータスポーリング間隔は、明示的に指定されていない場合は設定できません。デフォルトは5秒です。

## スケールアウト

スケールアウト機能を提供するには、ルートノードおよびコピーノードの少なくとも2つのサーバが必要です。コピーノードは、ミラーサーバと同様に読み込み専用ですが、プライマリロールが想定されることがないという点でミラーサーバとは異なります。読み込み要求のサービスに限定されています。プライマリサーバは writeFarm に関連付けられ、コピーノードは readFarm メンバーとしてミラーサーバに置き換わります。1つのルートノードと1つのコピーノードの場合は、合計2つの Outbound Enabler のみが必要です(各サーバに1つずつ)。すべてのコピーノードが停止しても環境が完全に機能できるようにするため、オプションとしてプライマリサーバを readFarm に追加できます。ルートノードに関連付けられている Outbound Enabler は、writeFarm への要求を受信するように設定されます。一方、コピーノードの Outbound Enabler は、readFarm への要求を受け入れるように設定されます。ルートノードは、書き込み更新を受信した時点でコピーノードを継続

的に更新します。コピーノードを readFarm に追加することで、設定を拡張できます。readFarm に対して 10 個のコピーノードを実行すると、負荷分散が行われ、readFarm 内で発生するサーバ障害が緩和されます。

## 高可用性とスケールアウト

上記の設定シナリオを組み合わせ、高可用性とスケールアウトの両方を提供することができます。このような設定では、書き込み可能 (プライマリ/ミラー) と読み込み可能 (ミラーとコピー) のバックエンドサーバに対してフォールトトレランスが提供されます。また、コピーノードをさらに追加することで、HTTP 読み込み専用要求に対する潜在的なスループットが高まります。この設定では、プライマリサーバとミラーサーバがルートノードを構成し、プライマリノード、ミラーノード、コピーノードのすべてに Outbound Enabler がそれぞれ 1 つずつ関連付けられて readFarm に加わります。プライマリサーバとミラーサーバには、writeFarm に加わるための追加の Outbound Enabler があります。オプションとして、プライマリサーバとミラーサーバを readFarm から削除できます。

## Outbound Enabler の設定

この説明では、次の項目を識別するために Outbound Enabler が設定されます。

- 関連付けられたサーバファーム
- Relay Server 接続パラメータ
- SQL Anywhere (バックエンドサーバ) 接続パラメータ

次のコマンドで、-cr オプションで指定されたパラメータに基づいて Relay Server への接続を設定する Outbound Enabler を示します。Outbound Enabler がアクティブの場合は、readFarm に対して行われた要求を、-cs オプションに基づいて設定されたバックエンドサーバに転送します。

```
rsoe2.exe -f readFarm -id saHost1 -cr url_suffix="/rs/server/rs.dll";host=rs_server;port=80 -cs host=sahost1;port=8080;
```

上記の例に基づいて、`http://rs_server/rs/client/rs.dll/readFarm/service_path[?service_query]` という形式の要求の URL は、rs\_server サーバでホストされている Relay Server に要求をダイレクトします。Relay Server は、URL からドメインとファームの各コンポーネントを削除し、ファームに関連付けられた Outbound Enabler に要求を Relay Server-Outbound Enable ジャンクション経由でリレーします。次に、Outbound Enabler は要求をバックエンドサーバに転送します。

Outbound Enabler は、最初にジャンクションをバックエンドサーバに接続できた場合のみ、ジャンクションを Relay Server に接続します。この動作はクライアントからの要求が届いたときにアクセスの遅延時間を減らしたり、バックエンドサーバと通信できなくなった場合にトラフィックが Outbound Enabler にダイレクトされるのを防いだりします。

## SQL Anywhere configuration

Outbound Enabler の設定の項で説明している Relay Server Outbound Enabler (RSOE) 起動例を活用して、oe\_read\_status サービスを次のように作成できます。

```
// READ-ONLY SA-OE ping service
```

```

// optionally includes primary as a READ node.
call sa_make_object( 'service', 'oe_read_status' );
alter service oe_read_status
  type 'raw'
  authorization off
  secure off
  user DBA
  as call sp_oe_read_status(:ro);
// if ro is 1, then only read-only servers are activated: mirror or copy
// if ro is 0, then include all servers: primary, mirror, copy...
create or replace procedure sp_oe_read_status( ro int )
begin
  declare readonly long varchar;
  declare res long varchar;
  set res = 'AVAILABLE=';
  set ro = isnull(ro, 0);
  if ro = 1 then
    select db_property('ReadOnly') into readonly;
    if readonly = 'On' then
      set res = res || 'TRUE';
    else
      set res = res || 'FALSE';
    end if;
  end if;;
  call sa_set_http_header('Content-Length', length(res) );
  select res;
end;

```

上記の Web サービスとストアドプロシージャは、プライマリノード、ミラーノード、コピーノードのロールを持つ任意のサーバに使用できます。ReadOnly データベースプロパティを On に設定すると、AVAILABLE=TRUE が返されます。ミラーを変更して、ReadOnly を Off に設定した場合にのみ、writeFarm などの書き込み可能なサーバファーム内で使用するサービスから AVAILABLE=TRUE が返されます。オプションとして、グローバルテーブルのエントリなどの他の要因に基づいて、Outbound Enabler に対する ping サービスの可用性を決めることもできます。たとえば、段階的な配備では、プライマリノードでテーブルやプロシージャを更新し、ミラーノードとコピーノードに伝達された時点で、各サーバのサーバファームメンバーシップをアクティブにしたり非アクティブにしたりします。

## Relay Server 設定

Relay Server を設定するには、次の手順に従います。

1. readFarm と writeFarm の各バックエンドファームを作成します。
2. プライマリノード、ミラーノード、コピーノードを readFarm Relay Server バックエンドファームのメンバーとして設定します。
3. プライマリノードとミラーノードを writeFarm Relay Server バックエンドファームのメンバーとして設定します。
4. どちらのファームも、active\_cookie 設定オプションを **yes** に設定し、active\_header 設定オプションを **no** に設定します。アフィニティを必要とせず、パフォーマンスを向上させる場合、active\_cookie 設定オプションを no に設定します。

## クライアント/アプリケーションの設定

### 制限事項

バックエンドサーバは、単一のデータベース上で実行する高可用性をサポートします。高可用性のフェイルオーバーの最大遅延は、次の項目の最大値になります。

- Outbound Enabler の構文で -d オプションで制御される Outbound Enabler のポーリング頻度。
- ミラーがプライマリになるまでの所要時間。

### 1.10.3.4 読み込み専用スケールアウトシステムにおけるデータベースへの接続

スケールアウトシステムを設定するには、ルートノードとして動作するデータベースサーバに対して CREATE MIRROR SERVER 文で 2 つの定義を作成します。

- (1 つはパートナーロール用)  
スケールアウトシステムでは、1 つのパートナーサーバを定義する必要があります。データベースサーバを起動するとき、パートナーサーバ定義で指定された名前が接続文字列で使用されます。
- (1 つはプライマリロール用)  
プライマリサーバとして定義されたデータベースサーバは、スケールアウトシステムにおけるコピーノードのデフォルトの親になります。プライマリサーバが定義されていない場合は、システムに追加するコピーノードごとに親を指定してください。

読み込み専用スケールアウトでは、アプリケーションはルートデータベースサーバに接続され、アプリケーションの接続文字列をコピーノードのステータスと負荷情報と一緒に使用して、アプリケーションの接続先となるノードをルートデータベースサーバによって決定することをお奨めします。NodeType=COPY 接続パラメータを使用して、負荷が最も低いコピーノードにアプリケーションを接続することができます。これを指定すると、ルートデータベースサーバがクライアントをそのノードにリダイレクトします。

このような接続と切断をアプリケーションで短時間に何度も行うと、接続がプールされ、どのコピーノードを使用するかがルートデータベースサーバに問われなくなります。この動作により、ルートデータベースサーバでの負荷は削減されますが、予想する結果にならない可能性があります。接続のたびに接続先となるコピーノードをルートサーバで確実に判断するため、接続のプールを禁止するようアプリケーションで指定することができます。

#### ➔ ヒント

SQL Anywhere モニタを使用してプライマリデータベースをモニタリングすることによって、スケールアウトシステムのデータベースサーバのステータスを確認します。

また、プライマリデータベースへの接続と、**正常性と統計情報**ウィンドウ枠にあるデータベースサーバのステータスのチェックによって、SQL Central からスケールアウトシステムをモニタします。

このセクションの内容:

#### [コピーノードへの接続 \[1741 ページ\]](#)

プライマリサーバに接続し、接続文字列の一部として NodeType 接続パラメータを指定すると、コピーノードに接続します。

## 関連情報

### [データベースミラーリングシステムの読み込み専用スケールアウトシステム \[1735 ページ\]](#)

[接続プールによるアプリケーションパフォーマンスの向上 \[244 ページ\]](#)

[データベースの正常性と統計情報 \[1058 ページ\]](#)

[レッスン 8: \(省略可\) SQL Anywhere モニタからの読み込み専用スケールアウトシステムのモニタリング \[1298 ページ\]](#)

### 1.10.3.4.1 コピーノードへの接続

プライマリサーバに接続し、接続文字列の一部として NodeType 接続パラメータを指定すると、コピーノードに接続します。

NodeType 接続パラメータには、次の値を指定できます。

#### **DIRECT**

これは、データベースサーバに接続する場合のデフォルト設定です。

NodeType 接続パラメータが DIRECT に設定されている場合、データベースサーバは、負荷分散またはリダイレクションを行わずに接続を受け入れます。

#### **PRIMARY**

プライマリサーバに接続したときに、NodeType 接続パラメータが PRIMARY に設定されている場合、接続は受け入れられます。非プライマリサーバ (ミラーサーバまたはコピーノードなど) に接続した場合、データベースサーバは接続をプライマリサーバにリダイレクトします。

#### **COPY**

NodeType 接続パラメータが COPY にセットされているとき、データベースサーバは負荷分散を行い、コピーノードを選択します。読み込み専用スケールアウトシステムでは、データベースサーバはその分岐内のコピーノード (ルートノードでない場合はサーバ自体も) を検査し、負荷が最も軽いコピーノードを選択します。データベースサーバがサーバ自体を選択しない場合、選択したデータベースサーバにクライアントをリダイレクトします。

#### **MIRROR**

ミラーサーバに接続したときに、NodeType 接続パラメータが MIRROR に設定されている場合、接続は受け入れられません。ミラーでないサーバに接続した場合、データベースサーバは接続をミラーサーバにリダイレクトします。

#### **READONLY**

NodeType 接続パラメータが READONLY に設定されているとき、コピーノードかミラーかに関わらず、任意の読み込み専用サーバに接続されます。コピーノードがない場合は、MIRROR 設定と同様になります。コピーノードのみある場合は、COPY 設定と同様になります。

接続されるコピーノードがルートデータベースサーバで判断されると、そのノードに接続が自動的に (必要に応じて) リダイレクトされます。

## 関連情報

[ミラーリングシステムでのデータベースへの接続 \[1665 ページ\]](#)

[NodeType \(NODE\) 接続パラメータ \[96 ページ\]](#)

## 1.10.3.5 読み込み専用スケールアウト設定

スケールアウトシステムを設定するために、実行する必要があるステップがいくつかあります。

1. `-xp on` データベースオプションを使用して、データベースサーバを起動します。このデータベースサーバは、スケールアウトシステムのルートノードです。
2. ルートデータベースにスケールアウトのオブジェクト定義を追加します。サーバを定義し、サーバのオプションを設定してください。
3. ルートデータベースのバックアップコピーを作成します。
4. コピーノード (データベースのバックアップコピー) を起動します。

### ユーザ認証の考慮事項

NodeType 接続パラメータを使用して、次のユーザ認証方法のうちのいずれかを使用する接続をリダイレクトしている場合、スケールアウトツリーのすべてのデータベースサーバを次のように設定する必要があります。

#### 統合化ログイン

(Windows のみ) スケールアウトツリーのすべてのデータベースサーバは、同一の Windows ユーザグループの一部であるか、同一の Windows ユーザがアクセスすることを許可するように設定する必要があります。

#### Kerberos ユーザ認証

スケールアウトツリーのすべてのデータベースサーバでは、Kerberos ユーザ認証を有効にしておく必要があります。スケールアウトツリーのすべてのサーバに対する Kerberos サーバプリンシパルは、同一の領域を持つ必要があります。

#### LDAP ユーザ認証

スケールアウトツリーのすべてのデータベースサーバでは、LDAP ユーザ認証を有効にしておく必要があります。LDAP サーバとの通信で TLS 暗号化を有効化するには、各データベースサーバは信頼できる証明書ファイルへアクセスする必要があります。このため、各コンピュータの同じファイルパスを使用して、信頼できる認証ファイルを格納する必要があります。

LDAP サーバでの認証が成功したときは常に、格納された値が異なっていれば、データベースのユーザのパスワードを更新しようとします。これはデータベースが読み込み専用かどうかによって、成功する場合と失敗する場合があります。また、更新したパスワードは他のノードに複写されません。このため、パスワードが変更され LDAP サーバが利用できない場合は、標準のユーザ認証へのフェイルオーバーが失敗することがあります。

#### PAM ユーザ認証

(UNIX のみ) スケールアウトツリーのすべてのデータベースサーバでは、PAM ユーザ認証を有効にしておく必要があります。このため、各コンピュータには適切な PAM 設定が定義されている必要があります。

PAM サーバでの認証が成功したときは常に、格納された値が異なっていれば、データベースのユーザのパスワードを更新しようとします。これはデータベースが読み込み専用かどうかによって、成功する場合と失敗する場合があります。また、更新したパスワードは他のノードに複写されません。このため、パスワードが変更され PAM サービスが利用できない場合は、標準のユーザ認証へのフェイルオーバーが失敗することがあります。

このセクションの内容:

#### [ルートノードの設定 \[1743 ページ\]](#)

コピーするデータベースを使用して、読み込み専用スケールアウトシステムのルートノードを作成します。

### コピーノードの追加 [1745 ページ]

読み込み専用のスケールアウトシステムにコピーノードを追加します。

### 子コピーノードを追加する方法 [1746 ページ]

SET MIRROR OPTION 文の child\_creation オプションは、読み込み専用スケールアウトシステムに子ノードを追加する方法を制御します。

### コピーノードの親の割り当て [1746 ページ]

コピーノードには親が必要です。

## 1.10.3.5.1 ルートノードの設定

コピーするデータベースを使用して、読み込み専用スケールアウトシステムのルートノードを作成します。

### 前提条件

MANAGE ANY MIRROR SERVER システム権限が必要です。

### コンテキスト

ルートノードを設定する場合、ルートデータベースサーバをパートナーサーバとして定義します。また、プライマリサーバも定義できます。

### 手順

1. 読み込み専用コピーにするデータベースが稼働しているデータベースサーバを起動します。データベースサーバを起動する場合は、`-xp on` データベースオプションを指定してください。

たとえば、次のコマンドはポート 6871 で起動する `scaleout_root_demo` というデータベースサーバを起動します。`-x` オプションを使用してポート番号を指定することが推奨されています。これは、このポート番号が後で CREATE MIRROR SERVER 文に使用されるからです。

```
dbsrv17 -n scaleout_root_demo -x TCPIP(port=6871) "c:¥scaleoutdemo
¥scaleoutdemo.db" -xp on
```

2. CREATE MIRROR SERVER 文を使用して、ルートデータベースでスケールアウトオブジェクトを定義します。

ルートデータベースサーバに対して、次の 2 つの定義を作成することが推奨されています。CREATE MIRROR SERVER...AS PARTNER および CREATE MIRROR SERVER...AS PRIMARY。両方の定義に対して、`connection_string` パラメータを指定する必要があります。

- a. AS PARTNER 句を指定する文でデータベースサーバに付ける名前は、データベースサーバの起動コマンドとクライアント接続文字列で使用する名前です。次に例を示します。

```
CREATE MIRROR SERVER "scaleout_root_demo"  
AS PARTNER  
connection_string = 'SERVER=scaleout_root_demo;HOST=localhost:6871';
```

- b. AS PRIMARY 句を指定する文でデータベースサーバに付ける名前は、スケールアウトシステムに追加するコピーノードのデフォルトの親となるデータベースサーバの名前です。プライマリサーバを定義しないと、コピーノードを作成するときに親サーバ名の指定が必要となるため、プライマリサーバを定義することをおすすめします。

```
CREATE MIRROR SERVER "scaleout_primary_demo"  
AS PRIMARY  
connection_string = 'SERVER=scaleout_primary_demo;HOST=localhost:6871';
```

3. SET MIRROR OPTION 文を使用して、データベースにスケールアウトオプションを設定します。authentication\_string パラメータに対して値を指定することと、child\_creation パラメータを自動 (デフォルト) に設定する必要があります。次に例を示します。

```
SET MIRROR OPTION child_creation='automatic';  
SET MIRROR OPTION authentication_string='abc';
```

## 結果

これで、ルートノードが使用可能になりました。

## 次のステップ

読み込み専用スケールアウトシステムに子ノードを追加するのに使用するルートデータベースのバックアップコピーを作成します。

## 関連情報

[子コピーノードを追加する方法 \[1746 ページ\]](#)

[-xp データベースオプション \[537 ページ\]](#)

[-x データベースサーバオプション \[500 ページ\]](#)

## 1.10.3.5.2 コピーノードの追加

読み込み専用のスケールアウトシステムにコピーノードを追加します。

### 前提条件

このタスクでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

ルートデータベースでは、`child_creation` の `SET MIRROR OPTION` が自動的に設定されています。

### コンテキスト

コピーノードの定義がデータベースに保存されます。スケールアウトシステムのコピーノードを事前に定義するか、コピーノードの接続時にルートデータベースサーバによって定義することができます。ルートデータベースサーバによってコピーノードを定義すると、定義時のエラーの発生率が抑えられるため、この方法をお奨めします。

### 手順

1. ルートデータベースのバックアップコピーを作成します。
2. データベースのコピーをコピーノードとして起動します。コピーノードを実行するデータベースサーバを起動するとき、ユニークな名前とポートを指定すると同時に、`-xp on` データベースオプションを指定する必要があります。次に例を示します。

```
dbsrv17 -n scaleout_child_demo -x TCPIP(port=6873) "c:¥scaleoutdemo¥copynode
¥scaleoutdemo.db" -xp on
```

### 結果

データベースは、コピーノードとして起動されます。

### 関連情報

[子コピーノードを追加する方法 \[1746 ページ\]](#)

[-xp データベースオプション \[537 ページ\]](#)

### 1.10.3.5.3 子コピーノードを追加する方法

SET MIRROR OPTION 文の child\_creation オプションは、読み込み専用スケールアウトシステムに子ノードを追加する方法を制御します。

child\_creation オプションでは、次の値がサポートされています。

#### 自動

ルートデータベースサーバでは、コピーノードが起動したときにそれを認証するか、またはコピーノードが不明の場合は新しく作成します。コピーノードを手動で作成しなくても、不明なコピーノードの定義がルートサーバで作成されるため、この設定をお奨めします。

#### オフ

ルートデータベースサーバに接続し、CREATE MIRROR SERVER 文を実行して新しいコピーノードを作成します。

#### 手動

コピーノードに接続し、そのデータベースサーバに対して CREATE MIRROR SERVER 文を実行することによって、コピーノードをツリーに追加します。この文では、MANAGE ANY MIRROR SERVER システム権限が必要です。コピーノードは、新しいコピーノードの定義要求をルートデータベースサーバに送信します。コピーノードが定義されると、ルートデータベースサーバにより新しいコピーノードからログページを要求できるようになります。

データベースには、システム内でプライマリデータベースサーバに関連付けられている接続文字列が格納されます。前に定義されていない新しいコピーノードを起動すると、この接続文字列を使用してルートデータベースサーバに接続されます。ルートデータベースサーバでは、データベースに格納されている authentication\_string オプションの値を使用して、コピーノードが認証されます。

コピーノードはミラー接続要求の一部として、コピーデータベースサーバの名前と、その IP アドレスとポートを含む文字列をルートデータベースサーバに送信します。コピーノードが認証されると、それが認識されるかどうかルートデータベースサーバで判断されます。コピーノードが不明の場合は、ルートデータベースサーバで CREATE MIRROR SERVER 文が実行され、新しいコピーとその接続文字列が定義されます。コピーが認識されると、ルートデータベースサーバはコピーへの接続を確立できます。次に、コピーにまだ存在しないすべてのトランザクションログページがコピーから要求され、それがすべて取得された時点で、ルートデータベースサーバは新しいトランザクションログページをコピーノードに送信し始めます。

すべてのコピーノードサーバの名前がユニークになるように指定します。

### 1.10.3.5.4 コピーノードの親の割り当て

コピーノードには親が必要です。

ルートデータベースサーバまたは別のコピーノードを親として指定できます。親が自動的に割り当てられるように、読み込み専用スケールアウトシステムを設定できます。また、スケールアウト階層内でのノードの位置をいつでも変更できます。

このセクションの内容:

#### [コピーノードへの親の自動割り当て \[1747 ページ\]](#)

SET MIRROR OPTION 文では、トランザクションログページの分散処理をノード間でバランスが取れるようにするため、新しいコピーノードをツリー内の親に割り当てるときに使用できる 2 つのオプションがサポートされています。

#### [ノードの位置 \[1747 ページ\]](#)

コピーノードがいったん実行中になると、ルートデータベースサーバで ALTER MIRROR SERVER 文を使用して、新しい親をコピーノードに割り当てることができます。

#### コピーノードの親の判断 [1748 ページ]

コピーノードは、ISYSMIRRORSERVER システムテーブルでコピーノードのミラーサーバ定義と親の値を確認することによって、接続先となるサーバを特定します。

### 1.10.3.5.4.1 コピーノードへの親の自動割り当て

SET MIRROR OPTION 文では、トランザクションログページの分散処理をノード間でバランスが取れるようにするため、新しいコピーノードをツリー内の親に割り当てるときに使用できる 2 つのオプションがサポートされています。

#### auto\_add\_server

自動割り当てツリーの最上位ノードとして動作するデータベースサーバの名前を指定します。

auto\_add\_server オプションが指定されている場合は、そのオプションで指定されたデータベースサーバが最上位のノードとして使用されます。add\_auto\_server オプションが指定されていない場合は、自動割り当てを使用してルートデータベースサーバがすべてのコピーノードの親として使用され、auto\_fan\_out オプションは無視されます。

#### auto\_add\_fan\_out

ツリー内の各分岐の最大コピーノード数を指定します。指定できる最小値は 2 で、デフォルトは 10 です。次の操作の結果作成される新しいノードには、親が自動的に割り当てられます。

- child\_creation オプションが Automatic に設定されている場合のプライマリサーバへの接続
- child\_creation が手動に設定されている場合の、コピーノードでの CREATE MIRROR SERVER...USING AUTO PARENT の実行
- ルートデータベースサーバでの CREATE MIRROR SERVER...USING AUTO PARENT の実行

ルートデータベースサーバは、ツリー内のどのノードを新しいコピーノードの親として割り当ててかを、定期的なステータス更新によってルートデータベースサーバに通知される各ノードの子の数とそのステータスに基づいて決定します。コピーノードでは、ルートデータベースサーバから受け取ったトランザクションログ操作を適用するときに、新しい親の割り当てがわかります。したがって、コピーノードは、トランザクションログ操作を受け取った時点で親の接続を変更します。

### 1.10.3.5.4.2 ノードの位置

コピーノードがいったん実行中になると、ルートデータベースサーバで ALTER MIRROR SERVER 文を使用して、新しい親をコピーノードに割り当てることができます。

これは次のような文を実行することで行います。

```
ALTER MIRROR SERVER "copy-server-name"  
FROM SERVER "new-parent-name";
```

この文は、ルートデータベースサーバ上のトランザクションログに記録されます。変更内容がコピーノードに適用されると、コピーノードで定義の変更が認識され、文で指定された新しい親に接続されます。

### 1.10.3.5.4.3 コピーノードの親の判断

コピーノードは、ISYSMIRRORSERVER システムテーブルでコピーノードのミラーサーバ定義と親の値を確認することによって、接続先となるサーバを特定します。

- NULL は、コピーノードの親がルートノードであることを示します。
- 親カラムに ID が表示されている場合は、その ID を持つサーバがコピーノードの親になります。

コピーノードで親が特定されると、データベースに格納されている connection\_string オプションを使用して、正しい親に接続されます。

### データベースミラーリングシステムでのコピーノードの親の特定

スケールアウトとデータベースミラーリングシステムを一緒に使用している場合は、コピーノードのパートナーがプライマリサーバまたはミラーサーバとして現在動作しているサーバとなるようにコピーノードを定義できます。この設定では、ISYSMIRRORSERVER で、1 つのサーバがタイプ PRIMARY、もう 1 つのサーバがタイプ MIRROR として定義されます。どちらのサーバも、コピーノードの親として選択できます。

### 1.10.3.6 読み込み専用のスケールアウトシステムを保守するには、次の手順に従います。

SQL Anywhere モニタを使用してプライマリデータベースをモニタリングすることによって、スケールアウトシステムのデータベースサーバのステータスを確認できます。

また、プライマリデータベースに接続することと、[正常性と統計情報ウィンドウ](#)にあるデータベースサーバのステータスをチェックすることによって、SQL Central からスケールアウトシステムをモニタすることもできます。

このセクションの内容:

#### [読み込み専用のスケールアウトシステムからのコピーノードの削除 \[1749 ページ\]](#)

コピーノードのミラーサーバ定義を削除することにより、読み込み専用のスケールアウトシステムからコピーノードのデータベースを削除します。

#### [読み込み専用のスケールアウトシステムにおける親接続の消失の処理方法 \[1750 ページ\]](#)

コピーノードの親サーバが使用できなくなっても、コピーノードのデータベースは引き続き使用できます。

#### [パートナーサーバのコピーノードへの変換 \[1751 ページ\]](#)

データベースミラーリングシステムのパートナーサーバを、システムを停止することなく、読み込み専用のスケールアウトシステムのコピーノードに変換します。

#### [読み込み専用のスケールアウトシステムへのミラーリングシステムの追加 \[1752 ページ\]](#)

監視サーバを作成し、コピーノードをミラーリングシステムの第 2 パートナーに変換することによって、読み込み専用のスケールアウトシステムのルートノードを、ミラーリングシステムのパートナーサーバになるよう設定します。

## 1.10.3.6.1 読み込み専用のスケールアウトシステムからのコピーノードの削除

コピーノードのミラーサーバ定義を削除することにより、読み込み専用のスケールアウトシステムからコピーノードのデータベースを削除します。

### 前提条件

MANAGE ANY MIRROR SERVER システム権限が必要です。

### 手順

1. プライマリ (ルート) サーバに接続します。
2. 削除するコピーノードに子コピーノードがある場合は、ALTER MIRROR SERVER 文を実行し、子コピーノードを別の親に再割り当てします。

```
ALTER MIRROR SERVER "child-copy-server-name"  
FROM SERVER "new-parent-name";
```

3. DROP MIRROR SERVER 文を実行して、削除するコピーノードからミラーサーバ定義を削除します。

```
DROP MIRROR SERVER "copy-server-name";
```

ミラーデータベースが停止します。サーバ上で実行されているデータベースがミラーデータベースのみの場合は、サーバも停止します。

### 結果

読み込み専用のスケールアウトシステムからコピーノードが削除されます。

### 次のステップ

データベースファイルを削除します。

### 関連情報

[コピーノードの親の割り当て \[1746 ページ\]](#)

## 1.10.3.6.2 読み込み専用のスケールアウトシステムにおける親接続の消失の処理方法

コピーノードの親サーバが使用できなくなっても、コピーノードのデータベースは引き続き使用できます。

コピーノードは、max\_retry\_connect\_time ミラーオプションで定義された期間 (デフォルトでは 120 秒) 中は引き続き親に接続しようとします。指定期間内に接続を確立できない場合、代替の親 (CREATE MIRROR SERVER STATEMENT の OR SERVER 句を使用して定義) が定義されていれば、さらに max\_retry\_connect\_time 秒間その親に接続しようとします。

接続ができない場合、コピーノードはルートデータベースサーバに接続を試み、ログページを取得します。元の親または代替の親に接続できずに promotion\_time が経過した場合、コピーノードはデータベースサーバに、コピーノードの以前の親をコピーノード自身で置き換え、コピーノードの兄弟を子として割り当てるよう要求します。この動作により、auto\_add\_fan\_out 設定で指定された個数を超える子がコピーノードに存在するという状態になることがあります。

コピーノードが親、代替の親、またはルートデータベースサーバに max\_disconnected\_time 秒間接続できない場合、データベースは停止します。デフォルト max\_disconnected\_time は、決して停止しない時間です。

コピーノードから別のデータベースサーバへの接続を確立しようとする間は、元の親への接続を引き続き試みます。この接続に成功すると、コピーノードは元の親データベースサーバからログページの取得を再開します。

コピーノードが最初に起動された場合は、ツリー内の他のノード (親も含む) が使用できなくても、読み込み専用の接続に関してはデータベースが使用可能になります。

### スケールアウトシステムでのサーバ状態の特定

次の文を使用して、コピーノードの親への接続状態を判断します。

```
SELECT DB_EXTENDED_PROPERTY ('MirrorServerState', server_name );
```

次の文を実行して、任意のノードにおけるトランザクションログの処理量を把握できます。

```
SELECT DB_PROPERTY ('CurrentRedoPos');
```

子ノードの CurrentRedoPos プロパティの値を、プライマリサーバの LastWrittenRedoPos データベースプロパティと比較して、コピーノードがどの程度最新の状態であるかを判断できます。

sa\_mirror\_server\_status システムプロシージャから、プロシージャを呼び出したサーバの下にあるすべてのサーバの接続状況が返されます。

### 1.10.3.6.3 パートナサーバのコピーノードへの変換

データベースミラーリングシステムのパートナサーバを、システムを停止することなく、読み込み専用のスケールアウトシステムのコピーノードに変換します。

#### 前提条件

MANAGE ANY MIRROR SERVER システム権限が必要です。

#### 手順

1. 変換するパートナサーバに接続し、パートナサーバにミラーロールがあることを確認します。変換できるのはミラーロールを持つパートナのみです。コピーノードに変換したいサーバがプライマリサーバの場合は、プライマリサーバとミラーサーバのロールが切り替わるように、フェイルオーバーを開始する必要があります。
2. プライマリサーバに接続します。
3. ミラー定義を変更します。
  - タイプ PRIMARY のミラー定義を更新して、変換されたパートナの接続情報を削除します。
  - タイプ MIRROR のミラー定義を削除します。
  - コピーノードに変換されるサーバのパートナー定義を変更します。

たとえば、次の文を実行します。

```
ALTER MIRROR SERVER mirror_demo_primary
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871';

DROP MIRROR SERVER mirror_demo_mirror;

ALTER MIRROR SERVER mirror_server2 AS COPY FROM SERVER PRIMARY;
```

ルートサーバおよび変換されたパートナサーバ上のデータベースは、これらの設定の変更の一部としてデータベースを再開しますが、サーバ自身は実行し続けます。データベースの再開中、データベースへの接続が削除されます。

4. (オプション) モニタリングシステムを削除する場合、監視サーバを削除します。

#### 結果

これでパートナーサーバは読み込み専用のスケールアウトコピーノードになりました。

#### 次のステップ

SYSMIRRORSERVER システムビューの内容を表示して、パートナーがコピーノードに変換されたことを検証します。

## 関連情報

[チュートリアル: パートナサーバのコピーノードへの変換 \[1761 ページ\]](#)

### 1.10.3.6.4 読み込み専用のスケールアウトシステムへのミラーリングシステムの追加

監視サーバを作成し、コピーノードをミラーリングシステムの第 2 パートナーに変換することによって、読み込み専用のスケールアウトシステムのルートノードを、ミラーリングシステムのパートナーサーバになるよう設定します。

#### 前提条件

MANAGE ANY MIRROR SERVER システム権限が必要です。

#### 手順

1. 読み込み専用のスケールアウトシステムのルートノードに接続します。たとえば、次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=passwd;SERVER=scaleout_root_demo"
```

2. 既存のパートナーおよびミラーサーバの定義を次に従って調整します。
  - 新しいミラーサーバのミラーサーバ定義を作成します。
  - 既存のプライマリ (ルート) サーバのミラーサーバ定義を、ルートサーバおよびミラーサーバ両方のホストとポートを含むように変更します。
  - 既存のパートナー (ルート) サーバのミラーサーバ定義を、ステータスファイルの場所を含むように変更します。ステータスファイルは自動的に作成されます。

たとえば、次の文を実行します。

```
-- Define mirror server definition for the mirror server
CREATE MIRROR SERVER scaleout_mirror_demo AS MIRROR
  connection_string='SERVER=scaleout_mirror_demo;HOST=localhost:6871,localhost:6873';
-- Alter existing primary to include both partners
ALTER MIRROR SERVER scaleout_primary_demo
  connection_string='SERVER=scaleout_primary_demo;HOST=localhost:6871,localhost:6873';
-- Alter the existing Partner to include state file
ALTER MIRROR SERVER scaleout_root_demo state_file='c:¥¥scaleoutdemo¥¥server1¥¥server1.state';
```

3. 既存のコピーノードをミラーサーバに変換します。たとえば、次の文を実行します。

```
ALTER MIRROR SERVER scaleout_child_demo AS PARTNER state_file='c:¥¥scaleoutdemo¥¥copynode¥¥server3.state'
```

- サーバを起動して、ミラーリングシステムの監視サーバにします。たとえば、次のコマンドを実行します。

```
mkdir c:¥scaleoutdemo¥arbiter
```

```
dbssrv17 -n scaleout_arbiter_demo -su passwd -x "TCPIP(PORT=6870)" -xf "c:¥  
¥scaleoutdemo¥¥arbiter¥¥arbiter.state" -xa "AUTH=abc;DBN=scaleoutdemo"
```

- プライマリ (ルート) サーバから、監視ミラーサーバ定義を追加します。たとえば、次の文を実行します。

```
CREATE MIRROR SERVER scaleout_arbiter_demo  
AS ARBITER  
connection_string ='SERVER=scaleout_arbiter_demo;HOST=localhost:6870';
```

## 結果

これで読み込み専用のスケールアウトシステムも、ミラーリングシステムの一部になりました。

## 次のステップ

なし。

## 関連情報

[チュートリアル: 読み込み専用のスケールアウトシステムへのミラーリングシステムの追加 \[1763 ページ\]](#)

[パートナーサーバの移動 \[1689 ページ\]](#)

### 1.10.3.7 チュートリアル: 読み込み専用のスケールアウトシステムの作成

コピーノードを自動的に追加するルートデータベースサーバを設定してモニタします。

## 前提条件

MANAGE ANY MIRROR SERVER および MANAGE ANY DBSPACE のシステム権限が必要です。

このチュートリアルでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

## コンテキスト

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、運用環境では、各データベースサーバを別々のコンピュータにインストールする必要があります。

異なるコンピュータで実行されているデータベースサーバでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピュータ名に変更する必要があります。

`%SQLANYSAMP17%`¥SQLAnywhere¥DBMirror に、スケールアウトシステムでデータベースミラーリングを使用するサンプルがあります。

### 1. [レッスン 1: 読み込み専用のスケールアウトシステムの作成 \[1754 ページ\]](#)

ルートノード (ルートデータベースサーバ) とコピーノード (データベースの読み込み専用バックアップコピー) を含むスケールアウトシステムを作成します。

### 2. [レッスン 2: 読み込み専用スケールアウトシステム \(SQL Central\) のモニタリング \[1756 ページ\]](#)

SQL Central を起動して `scaleoutdemo.db` に接続し、読み込み専用スケールアウトシステムをモニタします。

### 3. [レッスン 3: 読み込み専用のスケールアウトシステムのシャットダウン \[1758 ページ\]](#)

読み込み専用のスケールアウトシステムをシャットダウンします。これには `scaleout_child_demo` および `scaleout_root_demo` データベースサーバのシャットダウンが含まれます。

## 関連情報

[チュートリアル: 読み込み専用のスケールアウトシステムへのミラーリングシステムの追加 \[1763 ページ\]](#)

## 1.10.3.7.1 レッスン 1: 読み込み専用のスケールアウトシステムの作成

ルートノード (ルートデータベースサーバ) とコピーノード (データベースの読み込み専用バックアップコピー) を含むスケールアウトシステムを作成します。

## 前提条件

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## コンテキスト

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、運用環境では、各データベースサーバを別々のコンピュータにインストールする必要があります。

異なるコンピュータで実行されているデータベースサーバでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピュータ名に変更する必要があります。

## 手順

1. ディレクトリ `c:¥scaleoutdemo` および `c:¥scaleoutdemo¥copynode` を作成します。
2. 次のコマンドを実行して、データベース `scaleoutdemo.db` を作成します。これには、次のようにサンプルデータベースのデータが含まれています。

```
newdemo.bat c:¥scaleoutdemo¥scaleoutdemo.db
```

3. スケールアウトシステムのルートデータベースサーバを起動します。

```
dbsrv17 -n scaleout_root_demo -su passwd -x TCPIP(port=6871) "c:¥scaleoutdemo¥scaleoutdemo.db" -xp on
```

-su オプションを含めてユーティリティデータベースのパスワードを指定することをお奨めします。そうすることで、必要に応じてユーティリティデータベースに接続してデータベースサーバをシャットダウンできるようになります。

4. Interactive SQL からデータベースに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=scaleout_root_demo;DBN=scaleoutdemo"
```

5. Interactive SQL で、スケールアウトシステムにルートデータベースサーバを定義します。

```
CREATE MIRROR SERVER "scaleout_primary_demo"  
AS PRIMARY  
connection_string = 'SERVER=scaleout_primary_demo;HOST=localhost:6871';
```

6. スケールアウトシステムでルートデータベースサーバをパートナーとして定義します。パートナーサーバ名は、データベースサーバの起動コマンドで使用したデータベースサーバ名と一致している必要があります。

```
CREATE MIRROR SERVER "scaleout_root_demo"  
AS PARTNER  
connection_string = 'SERVER=scaleout_root_demo;HOST=localhost:6871';
```

7. スケールアウトシステムのルートデータベースサーバにオプションを設定します。

```
SET MIRROR OPTION auto_add_server='scaleout_primary_demo';  
SET MIRROR OPTION child_creation='automatic';  
SET MIRROR OPTION authentication_string='abc';  
SET MIRROR OPTION auto_add_fan_out='10';
```

8. データベースのバックアップコピーを作成し、ディレクトリ `c:¥¥scaleoutdemo¥¥copynode` に格納します。

```
BACKUP DATABASE DIRECTORY 'c:¥¥scaleoutdemo¥¥copynode';
```

9. コマンドプロンプトで、データベースのバックアップコピーを `scaleout_root_demo` データベースサーバのコピーノードとして起動します。

```
dbsrv17 -n scaleout_child_demo -su passwd -x TCPIP(port=6873) "c:¥scaleoutdemo¥copynode¥scaleoutdemo.db" -xp on
```

10. Interactive SQL からコピーノードに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=scaleout_child_demo"
```

コピーノードがルートデータベースサーバに接続すると、コピーノードがデータベースの読み込み専用コピーであることを示す警告が表示されます。これでコピーノードに接続し、これに対してクエリを実行できるようになります。

11. Interactive SQL で、次のクエリを実行すると、スケールアウトシステムでミラーサーバを表示できます。

```
SELECT * FROM SYSMIRRORSERVER;
```

12. すべての Interactive SQL ウィンドウを閉じます。

## 結果

読み込み専用スケールアウトシステムを作成しました。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: 読み込み専用のスケールアウトシステムの作成 \[1753 ページ\]](#)

次のタスク: [レッスン 2: 読み込み専用スケールアウトシステム \(SQL Central\) のモニタリング \[1756 ページ\]](#)

## 関連情報

[スケールアウトの動作 \[1734 ページ\]](#)

[読み込み専用スケールアウト設定 \[1742 ページ\]](#)

[読み込み専用のスケールアウトシステムを保守するには、次の手順に従います。 \[1748 ページ\]](#)

[-su データベースサーバオプション \[480 ページ\]](#)

### 1.10.3.7.2 レッスン 2: 読み込み専用スケールアウトシステム (SQL Central) のモニタリング

SQL Central を起動して `scaleoutdemo.db` に接続し、読み込み専用スケールアウトシステムをモニタします。

## 前提条件

このチュートリアルのこれまでのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

## 手順

1. SQL Central で [接続](#) ウィンドウを開き、テストデータベース `scaleoutdemo.db` に接続するための情報を以下のフィールドに入力します。
  - a. [ユーザ ID](#) フィールドに、`DBA` と入力します。
  - b. [パスワード](#) フィールドに、`sql` と入力します。
  - c. [アクション](#) フィールドで、[このコンピュータで稼働しているデータベースに接続](#) を選択します。
  - d. [サーバ名](#) フィールドに `scaleout_root_demo` と入力します。
  - e. [接続](#) をクリックします。
2. メインウィンドウにある [概要](#) タブをクリックして、データベース `scaleoutdemo.db` をモニタします。
3. [データベースのミラーリングおよびスケールアウトドロップダウンリスト](#) をクリックして、スケールアウトシステムのプライマリサーバ名を、スケールアウトシステムのコピーノードに関する情報と併せて表示します。

## 結果

SQL Central を使用して、読み込み専用スケールアウトシステムに接続し、モニタしています。

## 次のステップ

次のレッスンに進みます。

タスクの概要: [チュートリアル: 読み込み専用のスケールアウトシステムの作成 \[1753 ページ\]](#)

前のタスク: [レッスン 1: 読み込み専用のスケールアウトシステムの作成 \[1754 ページ\]](#)

次のタスク: [レッスン 3: 読み込み専用のスケールアウトシステムのシャットダウン \[1758 ページ\]](#)

## 関連情報

[SQL Anywhere モニタ \[1207 ページ\]](#)

[モニタを使用するためのクイックスタート \[1211 ページ\]](#)

### 1.10.3.7.3 レッスン 3: 読み込み専用のスケールアウトシステムのシャットダウン

読み込み専用のスケールアウトシステムをシャットダウンします。これには scaleout\_child\_demo および scaleout\_root\_demo データベースサーバのシャットダウンが含まれます。

#### 前提条件

このチュートリアルの前までのレッスンを完了している必要があります。

このチュートリアルの冒頭に一覧されているロールと権限を持っている必要があります。

#### 手順

読み込み専用のスケールアウトシステムをシャットダウンします。

- システムトレイで scaleout\_child\_demo データベースサーバのネットワークサーバアイコンをダブルクリックします。
- データベースサーバメッセージウィンドウで、[シャットダウン](#)をクリックします。
- [はい](#)をクリックしてクラスタを停止します。
- この手順を繰り返して、scaleout\_root\_demo データベースサーバをシャットダウンします。

#### 結果

読み込み専用のスケールアウトシステムがシャットダウンされます。

タスクの概要: [チュートリアル: 読み込み専用のスケールアウトシステムの作成 \[1753 ページ\]](#)

前のタスク: [レッスン 2: 読み込み専用スケールアウトシステム \(SQL Central\) のモニタリング \[1756 ページ\]](#)

## 1.10.3.8 チュートリアル: 1つのサーバをコピーノードおよび監視サーバとして使用

サーバを監視サーバおよびコピーノードとして実行するように設定します。

### 前提条件

MANAGE ANY MIRROR SERVER システム権限が必要です。

このタスクでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

### コンテキスト

このチュートリアルでは、読み込み専用のスケールアウトにも関わる、ミラーリングシステムの設定方法について説明します。ミラーリングシステムの監視サーバは、別のサーバの物理的に異なるコンピュータ上で実行することが推奨されていますが、それは容易ではありません。

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、運用環境では、各データベースサーバを別々のコンピュータにインストールする必要があります。

異なるコンピュータで実行されているデータベースサーバでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピュータ名に変更する必要があります。

### 手順

1. 次のディレクトリを作成します。c:¥server1、c:¥server2、および c:¥arbiter のファイルが含まれています。
2. サンプルデータベースのデータを格納し、トランザクションログを持つ、mirror\_demo.db というデータベースを作成します。トランザクションログを持っていないデータベースをミラーリングモードで起動することはできません。次のコマンドを実行します。

```
newdemo.bat c:¥server1¥mirror_demo.db
```

3. 1番目のデータベースサーバを起動します。次のコマンドを実行します。

```
dbsrv17 -n mirror_server1 -x "tcpip(PORT=6871)" -su passwd "c:¥server1¥mirror_demo.db" -xp on
```

4. mirror\_demo データベースに接続します。たとえば、次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server1"
```

5. ミラーサーバ定義を作成します。たとえば、次の SQL 文を実行します。

```
CREATE MIRROR SERVER mirror_demo_primary AS PRIMARY
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6872';
CREATE MIRROR SERVER mirror_demo_mirror AS MIRROR
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6872';

CREATE MIRROR SERVER mirror_server1 AS PARTNER
connection_string='SERVER=mirror_server1;HOST=localhost:6871' state_file='c:¥server1¥¥server1.state';

CREATE MIRROR SERVER mirror_server2 AS PARTNER
connection_string='SERVER=mirror_server2;HOST=localhost:6872' state_file='c:¥server2¥¥server2.state';
```

6. CREATE MIRROR SERVER 文で、監視サーバの定義を作成します。監視サーバの `mirror-server-name` は、接続文字列で指定したサーバ名とは異なる必要があります。

```
CREATE MIRROR SERVER demo_arbiter AS ARBITER
connection_string='SERVER=demo_server3;HOST=localhost:6870';
```

7. ミラーリングシステムのミラーリングオプションを設定します。認証文字列を指定して、`auto_add_server` をオプションをプライマリサーバの名前に設定する必要があります。次の文を実行します。

```
SET MIRROR OPTION auto_add_server='mirror_demo_primary';
SET MIRROR OPTION authentication_string='abc';
```

監視サーバの定義では、`arbiter-server-name` は監視サーバの実際のサーバ名とは異なる必要があります。クライアントの監視サーバへの接続文字列には、ServerName [SERVER] 接続パラメータと実際のサーバ名が含まれる必要があります。

8. ミラーサーバを作成します。

- a. `c:¥server1` にあるデータベースファイルとトランザクションログのコピーを作成し、`c:¥server2` と `c:¥arbiter` に追加します。次のコマンドを実行します。

```
dbbackup -c "UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=mirror_demo" server2
dbbackup -c "UID=DBA;PWD=sql;SERVER=mirror_server1;DBN=mirror_demo" arbiter
```

- b. ミラーサーバを起動します。

```
dbsrv17 -n mirror_server2 -x "tcpip(port=6872)" server2¥mirror_demo.db -su passwd -xp on
```

9. `-xa`、`-xp`、および `-xf` サーバオプションを指定して、監視サーバをコピーノードおよび監視サーバとして起動します。

```
dbsrv17 -n demo_server3 -x "tcpip(port=6870)" arbiter¥mirror_demo.db -xp on -xf arbiter¥arbiter.state -xa "AUTH=abc;DBN=mirror_demo" -su passwd
```

10. `dbping` ユーティリティを使用して、HA 設定の準備が整う時間を決定します。

```
dbping -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary" -pd
MirrorState,PartnerState,ArbiterState
```

```
SQL Anywhere Server Ping Utility Version 17.0.4.1691
Type          Property          Value
-----
Database      MirrorState      synchronized
Database      PartnerState     connected
```

```
Database ArbiterState connected
Ping database successful.
```

11. 次のクエリを使用して、この HA システムでミラーリングされるノードを表示します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server1" "SELECT server_name,state FROM sa_mirror_server_status()"
```

12. SYSMIRRORSERVER テーブルの内容を検査することで、demo\_server3 が COPY ノードで、監視サーバが接続していることを検証します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary" "SELECT server_name, server_type FROM SYSMIRRORSERVER"
```

## 結果

監視サーバを実行するサーバはコピーノードも実行します。

### 1.10.3.9 チュートリアル: パートナサーバのコピーノードへの変換

データベースミラーリングシステムのパートナサーバを、システムを停止することなく、読み込み専用のスケールアウトシステムのコピーノードに変換します。

#### 前提条件

このチュートリアルでは、データベースミラーリングシステムの作成に関するチュートリアルで説明しているデータベースミラーリングシステムを利用します。

MANAGE ANY MIRROR SERVER システム権限が必要です。

デフォルトでは、ALTER DATABASE...SET PARTNER FAILOVER 文を使用するには SERVER OPERATOR システム権限が必要です。-gd データベースサーバオプションを使用すると、必要な権限を変更できます。

#### コンテキスト

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、運用環境では、各データベースサーバを別々のコンピュータにインストールする必要があります。

異なるコンピュータで実行されているデータベースサーバでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピュータ名に変更する必要があります。

## 手順

1. 変換するパートナーサーバに接続し、パートナーサーバにミラーロールがあることを確認します。変換できるのはミラーロールを持つパートナーのみです。コピーノードに変換したいサーバがプライマリサーバの場合は、プライマリサーバとミラーサーバのロールが切り替わるように、フェイルオーバーを開始する必要があります。たとえば mirror\_server2 に接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_server2;HOST=localhost:6872"
```

データベースサーバのプロパティ MirrorRole は、サーバの現在のロールを返します。次の文を実行します。

```
SELECT DB_PROPERTY( 'MirrorRole' );
```

Primary が返された場合は、次の文を実行してフェイルオーバーを開始します。

```
ALTER DATABASE SET PARTNER FAILOVER;
```

現在のプライマリデータベースが停止して再開されます。

2. プライマリサーバに接続します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary;HOST=localhost:6871,localhost:6872"
```

3. 定義を変更します。

- タイプ PRIMARY のミラー定義を更新して、変換されたパートナーの接続情報を削除します。
- タイプ MIRROR のミラー定義を削除します。
- コピーノードに変換されるサーバのパートナー定義を変更します。

次の文を実行します。

```
ALTER MIRROR SERVER mirror_demo_primary
connection_string='SERVER=mirror_demo_primary;HOST=localhost:6871';

DROP MIRROR SERVER mirror_demo_mirror;

ALTER MIRROR SERVER mirror_server2 AS COPY FROM SERVER PRIMARY;
```

両方のパートナーサーバ上のデータベースは、これらの設定の変更の一部としてデータベースを再開しますが、サーバ自身は実行し続けます。データベースの再開中、データベースへの接続が削除されます。

4. (オプション) モニタリングシステムを削除する場合、監視サーバを削除します。次の文を実行します。

```
DROP MIRROR SERVER demo_arbiter;
```

5. SYSMIRRORSERVER システムビューの内容を表示して、mirror\_server2 がコピーノードに変換されたことを検証します。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=mirror_demo_primary" "SELECT server_name,
server_type FROM SYSMIRRORSERVER"
```

## 結果

これでミラーサーバは読み込み専用のスケールアウトコピーノードになりました。

## 関連情報

[チュートリアル: データベースミラーリングシステムの作成 \[1702 ページ\]](#)

### 1.10.3.10 チュートリアル: 読み込み専用のスケールアウトシステムへのミラーリングシステムの追加

ミラーリングシステムを既存の読み込み専用のスケールアウトシステムへ追加します。読み込み専用のスケールアウトシステムのルートも、ミラーサーバになります。

#### 前提条件

このチュートリアルは、読み込み専用のスケールアウトシステムの作成に関するチュートリアルの説明に従って、読み込み専用スケールアウトシステムを実行していることを前提としています。

ミラーサーバを設定し、監視サーバを作成し、コピーノードをミラーリングシステムのミラーパートナーに変換することによって、読み込み専用のスケールアウトシステムのルートノードを、ミラーリングシステムのプライマリサーバになるよう設定します。

MANAGE ANY MIRROR SERVER システム権限が必要です。

このチュートリアルでは、データベースのコピーのバックアップを作成します。選択するバックアップ方法に応じて、その方法に対して適切な権限を参照してください。

#### コンテキスト

このチュートリアルでは、すべてのデータベースサーバが同じコンピュータ上で実行されていることを想定しています。ただし、運用環境では、各データベースサーバを別々のコンピュータにインストールする必要があります。

異なるコンピュータで実行されているデータベースサーバでこのチュートリアルを使用する場合は、接続文字列内のローカルホストへの参照を、実際のコンピュータ名に変更する必要があります。

#### 手順

1. 読み込み専用のスケールアウトシステムのルートノードに接続します。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=scaleout_root_demo"
```

2. プライマリサーバとミラーサーバを定義します。
  - a. `c:¥scaleoutdemo¥server1` ディレクトリを作成します。
  - b. 次の文を実行します。

```
-- Alter existing primary to include both partners
```

```
ALTER MIRROR SERVER scaleout_primary_demo
connection_string='SERVER=scaleout_primary_demo;HOST=localhost:6871,localhost:
6873';
-- Alter existing Partner to include state file
ALTER MIRROR SERVER scaleout_root_demo state_file='c:¥¥scaleoutdemo¥¥server1¥
¥server1.state';
-- Define mirror
CREATE MIRROR SERVER scaleout_mirror_demo AS MIRROR
connection_string='SERVER=scaleout_mirror_demo;HOST=localhost:6871,localhost:
6873';
--Convert an existing copy node to the mirror server
ALTER MIRROR SERVER scaleout_child_demo
connection_string='SERVER=scaleout_child_demo;HOST=localhost:6873'
state_file='c:¥¥scaleoutdemo¥¥copynode¥¥server3.state';
```

3. 既存のコピーノードをミラーサーバに変換します。次の文を実行します。

```
--Convert an existing copy node to the mirror server
ALTER MIRROR SERVER scaleout_child_demo
connection_string='SERVER=scaleout_child_demo;HOST=localhost:6873'
state_file='c:¥¥scaleoutdemo¥¥copynode¥¥server3.state';
```

4. サーバを起動して、ミラーリングシステムの監視サーバにします。次のコマンドを実行します。

```
mkdir c:¥scaleoutdemo¥arbiter
```

```
dbsrvl7 -n scaleout_arbiter_demo -su passwd -x "TCPIP(PORT=6870)" -xf "c:¥
¥scaleoutdemo¥¥arbiter¥¥arbiter.state" -xa "AUTH=abc;DBN=scaleoutdemo"
```

5. ルートノードに接続して、管理サーバを追加します。次の文を実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=scaleout_primary_demo"
```

```
CREATE MIRROR SERVER scaleout_arbiter_demo
AS ARBITER
connection_string = 'SERVER=scaleout_arbiter_demo;HOST=localhost:6870';
```

6. 既存のコピーノードをミラーサーバに変換します。次の文を実行します。

```
ALTER MIRROR SERVER scaleout_child_demo AS PARTNER state_file='c:¥¥scaleoutdemo¥
¥copynode¥¥server3.state'
```

7. SYSMIRRORSERVER テーブルの内容を検査することで、scaleout\_child\_demo がミラーサーバであることを検証します。次のコマンドを実行します。

```
dbisql -c "UID=DBA;PWD=sql;SERVER=scaleout_primary_demo" "SELECT server_name,
server_type FROM SYSMIRRORSERVER"
```

これで scaleout\_child\_demo サーバは 2 番目のパートナーサーバになります。

8. クライアントアプリケーションの接続文字列を更新して、ホスト接続パラメータにおける両方のパートナーのアドレスを指定します。次に例を示します。

```
dbping -c "UID=DBA;PWD=sql;SERVER=scaleout_primary_demo;HOST=localhost:
6871,localhost:6873"
```

9. (オプション) 他のコピーノードを追加します。

- a. データベースのバックアップコピーを作成し、ディレクトリ c:¥scaleoutdemo¥copynode2 に格納します。

- b. 次の文を実行します。

```
BACKUP DATABASE DIRECTORY 'c:¥¥scaleoutdemo¥¥copynode2';
```

- c. データベースのバックアップコピーを scaleout\_root\_demo データベースサーバの子 (コピーノード) として起動します。次のコマンドを実行します。

```
dbsrv17 -n scaleout_child2_demo -su passwd -x TCPIP(port=6874) "c:¥scaleoutdemo¥copynode2¥scaleoutdemo.db" -xp on
```

## 結果

これで読み込み専用のスケールアウトシステムも、ミラーリングシステムの一部になりました。

## 関連情報

[チュートリアル: 読み込み専用のスケールアウトシステムの作成 \[1753 ページ\]](#)

[読み込み専用のスケールアウトシステムへのミラーリングシステムの追加 \[1752 ページ\]](#)

## 1.11 このマニュアルの印刷、再生、および再配布

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。

1. ここに示したものとそれ以外のすべての著作権と商標の表示をすべてのコピーに含めること。
2. マニュアルに変更を加えないこと。
3. SAP 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

ここに記載された情報は事前の通知なしに変更されることがあります。

# 重要免責事項および法的情報

## コードサンプル

この文書に含まれるソフトウェアコード及び / 又はコードライン / 文字列 (「コード」) はすべてサンプルとしてのみ提供されるものであり、本稼動システム環境で使用することが目的ではありません。「コード」は、特定のコードの構文及び表現規則を分かりやすく説明及び視覚化することのみを目的としています。SAP は、この文書に記載される「コード」の正確性及び完全性の保証を行いません。更に、SAP は、「コード」の使用により発生したエラー又は損害が SAP の故意又は重大な過失が原因で発生させたものでない限り、そのエラー又は損害に対して一切責任を負いません。

## アクセシビリティ

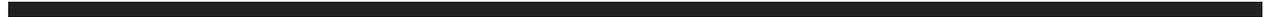
この SAP 文書に含まれる情報は、公開日現在のアクセシビリティ基準に関する SAP の最新の見解を表明するものであり、ソフトウェア製品のアクセシビリティ機能の確実な提供方法に関する拘束力のあるガイドラインとして意図されるものではありません。SAP は、この文書に関する一切の責任を明確に放棄するものです。ただし、この免責事項は、SAP の意図的な違法行為または重大な過失による場合は、適用されません。さらに、この文書により SAP の直接的または間接的な契約上の義務が発生することは一切ありません。

## ジェンダーニュートラルな表現

SAP 文書では、可能な限りジェンダーニュートラルな表現を使用しています。文脈により、文書の読者は「あなた」と直接的な呼ばれ方をされたり、ジェンダーニュートラルな名詞 (例:「販売員」又は「勤務日数」) で表現されます。ただし、男女両方を指すとき、三人称単数形の使用が避けられない又はジェンダーニュートラルな名詞が存在しない場合、SAP はその名詞又は代名詞の男性形を使用する権利を有します。これは、文書を分かりやすくするためです。

## インターネットハイパーリンク

SAP 文書にはインターネットへのハイパーリンクが含まれる場合があります。これらのハイパーリンクは、関連情報を見いだすヒントを提供することが目的です。SAP は、この関連情報の可用性や正確性又はこの情報が特定の目的に役立つことの保証は行いません。SAP は、関連情報の使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。すべてのリンクは、透明性を目的に分類されています (<http://help.sap.com/disclaimer> を参照)。



[go.sap.com/registration/  
contact.html](http://go.sap.com/registration/contact.html)

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

本書のいかなる部分も、SAP SE 又は SAP の関連会社の明示的な許可なくして、いかなる形式でも、いかなる目的にも複製又は伝送することはできません。本書に記載された情報は、予告なしに変更されることがあります。SAP SE 及びその頒布業者によって販売される一部のソフトウェア製品には、他のソフトウェアベンダーの専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。

これらの文書は、いかなる種類の表明又は保証もなしで、情報提供のみを目的として、SAP SE 又はその関連会社によって提供され、SAP 又はその関連会社は、これら文書に関する誤記脱落等の過失に対する責任を負うものではありません。SAP 又はその関連会社の製品及びサービスに対する唯一の保証は、当該製品及びサービスに伴う明示的な保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。

本書に記載される SAP 及びその他の SAP の製品やサービス、並びにそれらの個々のロゴは、ドイツ及びその他の国における SAP SE (又は SAP の関連会社) の商標若しくは登録商標です。本書に記載されたその他のすべての製品およびサービス名は、それぞれの企業の商標です。

商標に関する詳細の情報や通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx> をご覧ください。