

SQL Anywhere

文書バージョン: 17 - 2016-05-11

## SQL Anywhere - 新機能の確認

# 目次

<b>1</b>	<b>SQL Anywhere - 変更点とアップグレード</b> .....	<b>3</b>
1.1	SQL Anywhere 17.0 の最新パッチへのアップグレード.....	4
1.2	SQL Anywhere バージョン 17.0 の新機能 (製品全体).....	13
1.3	SQL Anywhere サーバ 17.0 の新機能.....	20
	パフォーマンスの強化.....	27
	SQL 文、関数、プロシージャ、データ型の変更.....	28
	データ管理の変更.....	32
	カタログの変更.....	36
	プログラミングインタフェースへの変更.....	37
	管理ツールの変更.....	40
	OData サポートへの変更.....	43
	セキュリティへの変更.....	45
	マニュアルの強化.....	48
	その他の変更点と強化点.....	48
	動作変更、廃止機能、およびサポートを終了した機能.....	52
1.4	Mobile Link バージョン 17.0 の新機能.....	57
1.5	Ultra Light バージョン 17.0 の新機能.....	61
1.6	Relay Server バージョン 17.0 の新機能.....	63
1.7	SQL Remote バージョン 17.0 の新機能.....	66
1.8	SQL Anywhere の以前のリリースの新機能.....	67
1.9	SQL Anywhere の最新バージョンへのアップグレード方法.....	68
	SQL Anywhere のアップグレード.....	70
	Mobile Link のアップグレード.....	103
	Ultra Light のアップグレード.....	116
	SQL Remote のアップグレード.....	120
	モニタのアップグレードおよびリソースの移行.....	121
	バージョン 16 のデータベースのアップグレードによる SQL Anywhere コックピットの使用.....	123
	バージョン 17 のデータベースのアップグレードによる SQL Anywhere コックピットの使用.....	124
	バージョン 16 の OData サーバの変換.....	125
1.10	このマニュアルの印刷、再生、および再配布.....	126

# 1 SQL Anywhere - 変更点とアップグレード

このマニュアルでは、*SQL Anywhere 17* とそれ以前のバージョンに含まれる新機能について説明します。*SQL Anywhere* の以前のリリースから最新バージョンへのアップグレード方法についても説明します。

このセクションの内容:

## [SQL Anywhere 17.0 の最新パッチへのアップグレード \[4 ページ\]](#)

ビルド 2000 から、SQL Anywhere 17.0 はいくつかのマイナーな強化点や動作の変更があります。

## [SQL Anywhere バージョン 17.0 の新機能 \(製品全体\) \[13 ページ\]](#)

SQL Anywhere 17.0 には、コンポーネントレベルの変更に加えて、知っておくべき製品全体の新機能および変更された機能がいくつか含まれています。

## [SQL Anywhere サーバ 17.0 の新機能 \[20 ページ\]](#)

SQL Anywhere 17.0 では、多くの新機能が導入され、プログラミングインタフェース、セキュリティ、パフォーマンス、SQL サポート、カタログ変更、管理ツールの領域のソフトウェア全体にわたって変更が加えられています。

## [Mobile Link バージョン 17.0 の新機能 \[57 ページ\]](#)

Mobile Link バージョン 17.0 のいくつかの新機能、変更、廃止、削除された機能についてご紹介します。

## [Ultra Light バージョン 17.0 の新機能 \[61 ページ\]](#)

Ultra Light バージョン 17.0 の新機能、また、変更、廃止、削除された機能についてご紹介します。

## [Relay Server バージョン 17.0 の新機能 \[63 ページ\]](#)

Relay Server バージョン 17.0 のいくつかの新機能、変更、廃止、削除された機能についてご紹介します。

## [SQL Remote バージョン 17.0 の新機能 \[66 ページ\]](#)

SQL Remote バージョン 17.0 では機能強化と動作の変更が行われています。

## [SQL Anywhere の以前のリリースの新機能 \[67 ページ\]](#)

次のリンクを使用して、以前のリリースの SQL Anywhere のマニュアルにアクセスします。そこで、機能がいつどのよう  
に追加、強化、廃止、削除されたかを確認できます。

## [SQL Anywhere の最新バージョンへのアップグレード方法 \[68 ページ\]](#)

メジャーリリース、サービスパック、パッチレベルの更新には、データベースをアップグレードまたは再構築しなければ  
ならない可能性のあるソフトウェアの機能強化が含まれています。

## [このマニュアルの印刷、再生、および再配布 \[126 ページ\]](#)

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

## 1.1 SQL Anywhere 17.0 の最新パッチへのアップグレード

ビルド 2000 から、SQL Anywhere 17.0 はいくつかのマイナーな強化点や動作の変更があります。

これらの強化点や変更を使用するには、データベースのアップグレードまたは再構築が必要です。SQL Anywhere データベースをバージョン 17.0 ビルド 2000 以降にアップグレードまたは再構築した場合、そのデータベースをバージョン 17.0 ビルド 2000 以前のデータベースサーバ上で実行することはできません。

### SQL Anywhere サーバ: 全体的な変更

#### Mac OS X 10.11 のサポート開始

SQL Anywhere で Mac OS X 10.11 をサポートできるようになりました。Mac OS X 10.11 を使用する場合、アプリケーションに次の調整が必要となる場合があります。

- libdbcapi を必要とする API を使用する場合、DYLD\_LIBRARY\_PATH を設定する必要があります。システム整合性保護の有効化により DYLD\_LIBRARY\_PATH が機能しない場合、SQLANY\_API\_DLL を使用します。
- JDBC のように、ネイティブライブラリとともに Java を使用する場合、DYLD\_LIBRARY\_PATH を設定します。システム整合性保護の有効化により DYLD\_LIBRARY\_PATH が機能しない場合、-Djava.library.path=/path/to/sqlanywhere/lib64 コマンドラインオプションを使用します。を参照してください。
- 外部関数を使用している場合、LD\_LIBRARY\_PATH を設定するか、共有ライブラリへのフルパスを提供する必要があります。
- カスタムインストールレイアウトを使用する場合、DYLD\_LIBRARY\_PATH を設定する必要があります。システム整合性保護の有効化により DYLD\_LIBRARY\_PATH が機能しない場合、SQLANY\_API\_DLL を使用します。DYLD\_LIBRARY\_PATH を使用せずに追加の検索パスを提供するには、install\_name\_tool を使用します。

ステートメントパフォーマンスサマリの更新 (データベースのアップグレードまたは再構築が必要)

ステートメントパフォーマンスサマリを使用して、速度が遅い文を特定し、トラブルシューティングします。収集されたデータを表示するには、SYS\_RUN\_PROFILER\_ROLE が必要です。

このアップデートをサポートするために、次のような変更が加えられています。

新しい **GTSYSPERFCACHESTMT** システムビュー GTSYSPERFCACHESTMT システムビューには、コストの高い文に対する SQL テキストが含まれています。

新しい **GTSYSPERFCACHEPLAN** システムビュー

GTSYSPERFCACHEPLAN システムビューは、コストの高い文に対するグラフィカル実行プランを含んでいます。

新しい **sp\_top\_k\_statements** システムプロシージャ

sp\_top\_k\_statements システムプロシージャは、最も長いランタイムを持つ文/プランの組み合わせを、指定された数だけ特定します。

新しい **sp\_find\_top\_statements** システムプロシージャ

sp\_find\_top\_statements システムプロシージャは、ログに記録されたそれぞれの文/プランプロシージャに対して収集されたパフォーマンス統計の詳細を示します。

-k データベースサーバオプションの強化

データベースサーバの開始時に -k を指定すると、Windows パフォーマンスモニタのステートメントパフォーマンスサマリが無効化されます。

## CREATE TEMPORARY TRACE EVENT SESSION 文の強化

新しい WHERE 句により、そのプロパティに基づく条件に応じてイベントを追跡できます。

新しい **-pf** オプション SQL Anywhere データベースサーバ実行ファイル (dbsrv17、dbeng17) または Broadcast Repeater ユーティリティ (dbns17) を使用するとき、**-pf** オプションを指定することで、指定されたファイルにプロセス ID を書き込むことができます。

プライマリサーバとミラーサーバ間のタイムラグを制御する新機能 (データベースのアップグレードまたは再構築が必要)

新しいラグタイムミラーリングオプションにより、トランザクションログの適用時にプライマリサーバに対するミラーサーバのおよそのラグタイムを指定できます。ラグタイムが設定した値に近づくと、プライマリサーバはトランザクションのレートを減少させます。

新しい **MILLISECOND**、**MICROSECOND**、**EXTRACT** 関数 (データベースのアップグレードまたは再構築が必要)

以前、TIMESTAMP 式のミリ秒およびマイクロ秒要素を取得するには、DATEPART 関数を使用していました。新たに、MILLISECOND および MICROSECOND 関数が使用できるようになりました。また、新しい EXTRACT 関数は TIMESTAMP および TIMESTAMP WITH TIMEZONE 式のさまざまな日付要素を返します。EXTRACT 関数は、ANSI/ISO SQL 標準に含まれています。

64 ビットになったパフォーマンスカウンタ (データベースのアップグレードと再構築が必要)

パフォーマンスカウンタが 32 ビットから 64 ビットに変更されました。

エラー後のアンロードの継続を可能にするアンロードユーティリティ (dbunload) の強化

以前は、データベースの再構築中 (dbunload -ac、-an、または -ar オプション) にエラーが発生した場合、再構築操作が停止し、エラーに対処しなければ再度再構築操作を試みることができませんでした。エラーの数が多くと、それだけ時間がかかっていました。新しい **-ae** オプションを使用すると、エラーをとりあえず無視してアンロード操作を継続し、操作が完了してからエラーの対処法を決めることができます。reload.sql スクリプト内の文が失敗した場合、リロード操作は停止することなく次の文へと継続されます。失敗した文は unprocessed.sql スクリプト内に記録され、**-q** オプションが指定されていない限りコマンドライン出力に表示されます。この強化により、エラーのすべてを一度に表示でき、報告されたエラーのすべてを修正するか、データベースを一から再構築するか、または完了していないデータベースに unprocessed.sql スクリプトを適用するかを決定できます。

グローバルテンポラリテーブルへの強化

以前は、接続によってグローバルテンポラリテーブルが参照されると、ロックが必要か否かに関わらず、接続が閉じるまでテーブル上にロックに相当するものが配置されていました。ロックを開始した接続が終了するまで、グローバルテンポラリテーブル上の他の接続による ALTER 文は禁止されていました。今では、接続が最初にテーブルを参照するとき、テーブル上に明示的なスキーマロックが配置されるようになりました。接続がカーソル内でテーブルを参照している場合、またはテーブルにローがある場合のみ、ロックは維持されます。テーブルがカーソルによって参照されていない場合やローを持たない場合、コミットまたはロールバック時にロックは削除され、他の接続がテーブル上で ALTER 文を実行できます。どのグローバルテンポラリテーブルにスキーマロックが配置されているかを判別するには、sa\_locks システムプロシージャを用います。

xp\_startsmtp システムプロシージャの強化

以前は、電子メールセッションを開始する際、データベースサーバは、SMTP サーバを使用したユーザ ID とパスワードの PLAIN 認証のみをサポートしていました。今では、データベースは PLAIN 認証だけでなく、CRAM-MD5 認証もサポートできるようになりました。

smtp\_auth\_username と smtp\_auth\_password パラメータとともに xp\_startsmtp システムプロシージャを使用する場合、データベースサーバは CRAM-MD5 認証を使用します。SMTP サーバが CRAM-MD5 認証をサポートしない場合、データベースサーバは PLAIN 認証を使用します。**-fips** データベースサーバオプションを指定する場合、PLAIN 認証のみが使用されます。**-fips** データベースサーバオプションは CRAM\_MD5 認証をサポートしません。

新しい `sp_disk_info` システムプロシージャ このシステムプロシージャを使用すれば、特定のパスに対するディスクドライブ情報を取得できます。

新しい **BEGIN PARALLEL WORK** 文を使用した、**CREATE INDEX** 文と **LOAD TABLE** 文の並列実行

**BEGIN PARALLEL WORK** 文は、**CREATE INDEX** 文のリストまたは **LOAD TABLE** 文のリストを並行で実行することで、パフォーマンスを向上させることができます。データベースサーバが実行されているコンピュータ上で使用可能な論理プロセッサの数、ならびに `-gtc` オプションおよび `max_parallel_statements` オプションの設定は、同時に実行できる文の数を制限します。

このアップデートをサポートする文、オプション、およびユーティリティは次のとおりです。

- **BEGIN PARALLEL WORK** 文
- **CREATE INDEX** 文
- **LOAD TABLE** 文
- 新しい `max_parallel_statements` オプションは、**BEGIN PARALLEL WORK** 文内で同時に実行できる文の数を制限します。
- `Unload` ユーティリティ (`dbunload`) は、**CREATE INDEX** 文や **LOAD TABLE** 文の並列実行のために最適化された `reload.sql` ファイルを作成するための、新しい `-bp` オプションをサポートします。`-bp` オプションが指定されているとき、**CREATE INDEX** 文と **LOAD TABLE** 文は `reload.sql` ファイルの **BEGIN PARALLEL WORK** 文内にリストされます。

特定のビューでの **LOAD TABLE** 文の使用

**LOAD TABLE** 文を使用することで、通常のビューにコンテンツをロードできるようになりました。ビューはベーステーブルを基礎とし、次のように定義されている必要があります。

```
CREATE VIEW
[owner.]view-name
AS SELECT * FROM base-table-name
```

このアップデートをサポートするため、**GRANT** 文によってビューに **LOAD** 権限を付与できるようになりました。

**CREATE TABLE** 文と **CREATE EXISTING TABLE** 文の **AT** ロケーション文字列での **ESCAPE CHARACTER** 句のサポート

まれな例として、ロケーション句内の区切り文字の 1 つを文字どおりに解釈する必要がある場合、ロケーション区切り文字をエスケープするためのエスケープ文字を指定できます。このような場合、このカラムにはプロキシテーブルの作成時に指定されたエスケープ文字が含まれます。

## SQL Anywhere サーバ:セキュリティの変更

新しい **ACCESS DISK INFORMATION** システム権限

新たに追加されたシステム権限 **ACCESS DISK INFORMATION** により、ユーザは `sp_disk_info` システムプロシージャを使用することで、合計ディスクサイズと残りのディスク領域に関する情報にアクセスできます。このシステム権限は、ユーザ定義の `COCKPIT_ROLE` ロールに含まれています。これは、新しい `sp_disk_info` システムプロシージャの実行に必要です。を参照してください。

クライアントとデータベースサーバ間におけるパスワード交換のセキュリティの強化

データベースサーバに接続するとき、今やクライアントは RSA 暗号化を使用して、より安全にユーザパスワードを暗号化してから、データベースサーバに送信できます。このアップデートは TDS 接続には適用されません。ユーザの作成や変

更の際に GRANT CONNECT、CREATE USER、または ALTER USER 使用するとき、文内のパスワードは暗号化されません。クライアントおよびデータベースサーバ両方のソフトウェアのバージョンは、バージョン 17.0、ビルド 2000 以降である必要があります。

PKCS#5 標準の一部である PBKDF2 (パスワードベースのキー導出機能 2 番) を使用した、強力な暗号化キーの作成 (データベースのアップグレードと再構築が必要)

データベースを暗号化したり、データベース内のテーブル暗号化を有効化する際、今やソフトウェアは PKCS#5 標準の一部である PBKDF2 (パスワードベースのキー導出機能 2 番) を使用するようになりました。当て推量攻撃から暗号化キーを保護するため、ソフトウェアは暗号化キーに対して暗号化ハッシュを繰り返して適用します。デフォルトでは、繰り返しが 2000 回適用されます。暗号化データベースの作成時に、適用される繰り返しの数を変更できます。

このアップデートをサポートするために、次のような機能が変更されました。

**SQL Anywhere** クライアントサーバとデータベースサーバによる、クライアント側証明書のサポート

以前は、クライアントが信頼されているデータベースサーバに接続していることを保証するサーバ側認証のみがサポートされていました。現在は、クライアント側認証もサポートされているため、クライアントがデータベースサーバ信頼済みソースによって署名された証明書を使用している場合、データベースサーバのみに接続することができます。

TLS および HTTPS のための新しい `allow_expired_certs` 暗号化オプション

クライアント上でこのオプションを設定することで、クライアントライブラリは、期限切れまたは有効化されていないルート証明書やデータベースサーバ証明書を信頼できます。データベースサーバ上でこのオプションを設定することで、データベースサーバは期限切れまたは有効化されていないファイルを特定し、受け入れることができます。を参照してください。

クライアントファイルセキュリティへのアップデート

新しい接続パラメータ `ClientFileValidator` を使用すると、データベースサーバがローカルファイルを読み込むタイミングを指定できます。リクエストがクライアントから直接出されたものとデータベースサーバが判断すると、またはコールバックが TRUE を返すと、ファイルは転送されます。

以前のリリースでは、一部のクライアントに対して、未検証のソースからファイルリクエストが出されると呼び出されるバリデータが配置されていました。今回のアップデートにより、この動作が変更されます。以前の動作を使用するには、クライアント上で `ClientFileValidator` 接続パラメータを `TrustedServer` に設定してください。

## SQL Anywhere サーバ:カタログの変更

**SYSPROCEDURE** システムビューと **SYSVIEWS** 互換ビューに追加された新カラム (データベースのアップグレードと再構築が必要)

**SYSPROCEDURE** システムビューに次の 6 つのカラムが追加されました。

- `dialect`
- `is_deterministic`
- `is_external`
- `external_language`
- `external_name`
- `sql_security`

さらに、**SYSVIEW** 互換ビューに新しい `check_option` カラムが追加されました。

## SQL Anywhere サーバ:Web サービスの変更

同じ名前で複数のヘッダを返す Web サービスプロシージャ (データベースのアップグレードと再構築が必要)

以前、Web サービスプロシージャでは、HTTP サーバが同じ名前を持つ複数のヘッダを返した場合、ヘッダ値の最初のインスタンスのみを返していました。現在は、HTTP Web プロシージャ結果セットの 3 番目のカラムでヘッダのインスタンスを示すことにより、同じ名前を持つ複数のヘッダを扱うことができるようになりました。今では HEADERS 句で同じ名前を持つ複数のヘッダを指定する Web サービスプロシージャは、これらのすべてのヘッダを Web サーバに送信します。このアップデートを使用するには、データベースをバージョン 17 SP1 にアップグレードしてください。

新しいデータベースプロパティである HttpClientMultipleHeaders は、データベースが同じ名前を持つ複数のヘッダをサポートするかどうかを示します。

ステータスコードに対する細かな調整を可能にする Web サービスプロシージャ

SET 句の EXCEPTIONS オプションにより、ステータスコードの処理を制御できます。

HTTP/HTTPS 接続キューのサポート

以前は、データベースが HTTP/HTTPS 接続の限界数に達すると、データベースへのさらなる接続の試みは拒否されていました。

今では、データベースサーバが HTTP/HTTPS 接続の限界数に達すると、さらなる接続の試みはキューに入れられません。接続が可能になると、キュー内の最初の接続が処理されます。

標準接続はキューに入れられません。したがって、データベースサーバの接続数が限界に達すると、標準接続の新しい試みは失敗します。

パーソナルデータベースサーバでは、標準接続のために最低でも 3 つの接続が予約されています。

ネットワークデータベースサーバは、標準接続のための接続の予約は行われません。reserved\_connections データベースオプションを使用することで、標準接続を予約します。

このアップデートをサポートするため、次のプロパティとオプションが追加されました。

- HttpConnectionsQueued、HttpQueueCount、HttpQueueMaxCount、HttpQueueTimedOut データベースサーバプロパティ
- reserved\_connections オプション

## SQL Anywhere サーバ:プログラミングの変更

Embedded SQL:FETCH に対する 32765 バイト長の制限廃止

Embedded SQL は、DT\_VARCHAR、DT\_NVARCHAR、および DT\_BINARY のフェッチの最大長を 32765 バイトから 32767 バイトに変更しました。Embedded SQL は今後、32767 バイトのプット (挿入) をサポートします。

Embedded SQL プリプロセッサ (SQLPP) への修正には、DECL\_VARCHAR、DECL\_NVARCHAR、または DECL\_BINARY を使用するアプリケーションをプリプロセッサで再コンパイルする必要があります。

Embedded SQL:ワイドマージへのサポート

Embedded SQL では、EXECUTE 文 [ESQL] の ARRAY 句を使用したワイドマージがサポートされるようになりました。

ODBC:ワイドマージへのサポート



ODBC では、SQLSetStmtAttr (SQL\_ATTR\_PARAMSET\_SIZE) および SQLBindParameter を使用したワイドマージがサポートされるようになりました。

#### JDBC:バッチマージへのサポート

JDBC では、PreparedStatement.addBatch() および PreparedStatement.executeBatch() を使用したワイドマージとバッチマージがサポートされるようになりました。

#### SQL Anywhere C API:ワイドマージに対するサポートの追加

SQL Anywhere C API のバージョン 4 で、ワイドマージがサポートされるようになりました。

#### .NET 4.0 のサポート終了

SQL Anywhere .NET Data Provider の .NET 4.0 バージョンは削除されました。Microsoft は今後、.NET 4.0 に対するセキュリティアップデート、テクニカルサポート、またホットフィックスの提供を行いません。.NET 4 以降のバージョンは、引き続き SQL Anywhere .NET Data Provider によってサポートされます。

#### Node.js SQL Anywhere サーバの複数のバージョンをサポート

Node.js の 0.10、0.12、4.x.y、5.x.y など、さまざまなバージョンに対してドライバが提供されています。

#### DataNucleus と OpenJPA のサポート

DataNucleus と OpenJPA 用のアダプタが実装されました。

SQL Anywhere 向けアダプタが DataNucleus に追加されました。<http://www.datanucleus.org/products/datanucleus/datastores/rdbms.html> を参照してください。

SQL Anywhere 用のアダプタが OpenJPA に対して書き込まれました (OpenJPA プロジェクトによる採用は保留)。<https://github.com/sqlanywhere/OpenJPA> を参照してください。

Jetty サポートのアップグレード OData サーバで Jetty 9.3.7 が使用されるようになりました。

Node.js の 0.10、0.12、4.x.y、5.x.y など、さまざまなバージョンに対してドライバが提供されています。

## SQL Anywhere サーバ:動作の変更

### 最小通信パケットサイズを 1000 バイトに変更

クライアント接続やデータベースサーバによって許可される通信パケットの最小サイズが 1000 バイトになりました。以前の最小サイズは 500 バイトでした。

このアップデートの使用にあたって、クライアントをアップデートする必要はありません。バージョン 17 SP1 データベースサーバに接続する旧クライアント (最小パケットサイズが 1000 未満) には、新しい最小値が適用されます。しかし、17 SP1 ソフトウェアを実行するクライアントは、以前のバージョンのソフトウェアを実行し、かつ 1000 バイト未満というデフォルトのパケットサイズを使用しているデータベースサーバに接続することはできません。

### 空間参照系の境界を越えて作成できるようになった空間ジオメトリ

以前、ジオメトリによる SRS 境界越えの上限は、各方向に 50 パーセントのみで、それを超過するとエラーが発生していました。曲面 SRS でも、ジオメトリポイントが許可された範囲を超えると、同じエラーが発生していました。今では、すべてのポイントが SRS 座標系で表示可能かつジオメトリが自明に有効である限り、ジオメトリは SRS 境界を越えることができるようになりました。SRS 境界を 50 パーセント以上越えるジオメトリは、インデックス化できません。

新しい ST\_Geometry メソッドである ST\_IsIndexable は、インデックス化できないジオメトリを検出します。

## 管理ツールの変更

### コックピットの強化

コックピットには次のような変更が加えられました。

イベント履歴情報の表示 以前は、履歴情報の表示はイベントがアクティブである間に限られていました。今では、イベントの現在のステータスに関わらず、履歴情報を表示できるようになりました。

#### 接続カウントの単純化

以前、データベースサーバへのすべての接続は、コックピット自体による接続を含め、データベースサーバに対する接続カウントに含まれていました。今は、コックピットによって行われた接続が、接続カウントから除外されるようになりました。

#### コックピットアラートの電子メール通知

アラートを電子メールで知らせようコックピットを設定します。どのタイプのアラートを、どれくらいの頻度で電子メール通知するかを指定します。

#### 新しいコックピット向けアラート

データベースファイル、dbspace、またはトランザクションログといったファイルを含むドライブ上でディスク領域が低下すると、アラートが発生します。

データベースページの検索において、キャッシュ内のページ検索で見つからない割合が高くなると、アラートが発生します。

#### COCKPIT\_ROLE ユーザ定義ロールに新しい権限を追加

COCKPIT\_ROLE ユーザ定義ロールに ACCESS DISK INFORMATION システム権限が追加されました。コックピットの全機能にアクセスするには、ACCESS DISK INFORMATION システム権限が必要です。

#### コックピットデータベースから新しいデータベースへの変換

コックピットがテンポラリーデータベースとともに実行中であるとき、設定やアラート履歴を失うことなく、恒久的なデータベースに変換できます。

このアップデートをサポートするため、sa\_server\_option システムプロシージャの cockpitdb オプション、-cdb データベースサーバオプション、および CockpitDB データベースサーバプロパティが更新されました。

### プロファイラの強化

#### 負荷の自動分析

プロファイラは自動的に負荷を分析し、パフォーマンスを向上させるために集中すべき領域を示します。このレポートは、パフォーマンスが次のどの要素により制限されているかといった質問に対処します。

- ブロッキング
- データベースサーバのハードウェア
- サーバワーカー
- バックアップ、チェックポイント、またはほかの内部的なサーバアクティビティ

プロファイリングセッションで収集される情報のターゲットを絞るためのプロファイリングオプションが新たに追加されました。

以前は、データベース内で発生するすべてのアクティビティについての情報を収集する、包括的なプロファイリングしか実行できませんでした。今では、指定された条件を満たすデータベースサーバパフォーマンス統計や SQL 文に対して収集される情報の量を減らし、ターゲットを絞ったプロファイリングを実行できます。ターゲットを絞ることにより、

本稼働環境にあるデータベースなどを、長時間にわたってプロファイリングすることが可能になります。を参照してください。

#### サーバの負荷パネルの改良

サーバの負荷パネルに、新しい2つの値、**スケジュールされていない要求とキャッシュサイズ**が追加されました。サーバの負荷パネルはまた、グラフ化する値を選択するための事前設定や、グラフを構成するためのオプションを含んでいます。

#### データベースサーバに関する SAP サポート診断情報の送信

サポートによってリクエストされた場合、プロファイラを使用してデータベースサーバの診断情報のスナップショットをとり、ファイルに保存し、それをサポートに電子メールで送信できるようになりました。さらにプロファイラを使用して、データベースサーバのステートメントパフォーマンスサマリを表示することもできます。

#### 検索条件式を編集する機能の改良

検索条件式の編集ウィンドウで検索条件式を編集する際、フィルタ構文文法内のすべての句にアクセスし、使用できるようになりました。さらに、ユーザインタフェースにサンプル付きのヘルプが追加されました。

[スタート]メニューまたはコマンドラインからのプロファイラの開始

▶ **スタート** ▶ **プログラム** ▶ **SQL Anywhere 17** ▶ **管理ツール** ▶ **SQL Anywhere Profiler** ▶ をクリックして、プロファイラを起動します。プロファイラユーティリティ (dbprof) を使用してプロファイラを開始することもできます。を参照してください。

### Interactive SQL の強化

Interactive SQL ユーティリティ (dbisql) の **-c** オプションと **-f** オプションの併用が可能に 以前は、**-f** オプションを指定すると **-c** オプションは無視されていました。今は、両方のオプションを一緒に使用できます。

#### マルチライン SQL 文におけるエラー報告機能の改良

ファイルから SQL 文を実行して、それらの文のいずれかが失敗すると、Interactive SQL はエラーが発生したファイル名と行番号を報告します。

#### ツールバーに接続ボタンと切断ボタンを追加

Interactive SQL のツールバーメニュー項目を使用して、データベースやサーバとの接続や切断を行います。

Interactive SQL オプションは、SQL 文の警告をエラーとして処理し、表示します。

Interactive SQL において、実行済み SQL 文内の警告をエラーとして処理し、表示させるには、**-we** オプションを指定します。**-q** オプションが指定されると、**-we** オプションもまた指定されている場合を除き、警告は非表示になります。

## Mobile Link の変更

### Mobile Link クライアントの新しい allow\_expired\_certs ネットワークプロトコルオプション

allow\_expired\_certs プロトコルオプションを使用して、期限切れになった、またはまだ有効でないサーバ証明書を受け入れて、同期を続行します。

### 大規模な同期システムの開発に役立つ、新しい Mobile Link ユーティリティ

Mobile Link テンプレートユーティリティ (mltemplate) は、高度な Mobile Link 開発者のための生産性ツールで、Mobile Link 同期スクリプトやそれらに付随する他のオブジェクトの作成に役立ちます。

このユーティリティとともに、MLTEMPLATE\_JARS 環境変数を使用します。

## Mobile Link の動作の変更

### buffer\_size Mobile Link クライアントネットワークプロトコルオプションに対するデフォルト値の変更

buffer\_size クライアントネットワークプロトコルオプションのデフォルト値が変更されました。Windows および Unix に対するデフォルト値は 64 KB から 256 KB に、Android、iOS、Linux ARM、Windows Phone/Store デフォルト値は 16 KB から 64 KB に増やされました。Windows Mobile デフォルト値は 16 KB のままです。

### Oracle 統合データベース用の新しい Fetch array size オプション

Fetch array size オプションを使用すると、Oracle 統合データベースからフェッチするローの数を指定できます。ローの数を増やすと、データをフェッチするための往復数が減り、パフォーマンスが向上します。

### Mobile Link サーバ実行ファイルの変更:mlserv17.dll (Microsoft Windows) および libmlserv17\_r.so (Unix)

Mobile Link サーバは次の 2 つの実行ファイルに分割されています。

表 1:

Microsoft Windows	mlsrv17.exe mlserv17.dll
UNIX	mlserv17.dll libmlserv17_r.so

## Ultra Light の変更

### Ultra Light の同時実行性

Ultra Light アプリケーションで複数のデータベースを開くとき、別のデータベースに関するリクエストを同時に実行できるようになりました。

### 新しい UploadUnknown データベースプロパティ

このプロパティは、最近の同期の失敗が、アップロードの送信後かつ同期サーバからのアップロード確認の受信前に発生したかどうかを示す値を返します。

### Ultra Light C++ API に新しいメソッドを追加

Ultra Light C++ API に次の新しい 3 つのメソッドが追加され、byte 配列を通じてカラムやパラメータ値を取得または設定できるようになりました。ULResultSet::GetBytes、ULResultSet::SetBytes、ULPreparedStatement::SetParameterBytes。

### WinRT API への変更

Ultra Light for WinRT API が、Microsoft Windows 10 でサポートされるようになりました。

### Ultra Light J への変更

Ultra Light J が Android x86 でサポートされるようになりました。

Ultra Light J for Android では、TLS の使用を含め、TCP/IP を介したデータ同期が可能になりました。この機能は、StreamTCPIPParams と StreamTLSParams という新しい 2 つのクラスによってサポートされています。これらのクラスは、それぞれ TCP/IP および TLS 同期に対するストリームパラメータを設定します。クラス SyncParams には、新たに

TCPIP\_STREAMとTLS\_STREAMという2つの定数が追加されています。これらの定数は、Connection.createSyncParms メソッドにより、SyncParms オブジェクトを作成します。

## マニュアルの変更

新しいチュートリアル LDAP ユーザ認証環境の作成のためのチュートリアルが新たに追加されました。

API マニュアルへのアクセス

### i 注記

API リファレンスマニュアルをお探しですか。マニュアルをローカルにインストールした場合は、Windows のスタートメニューを使用してアクセスするか (Microsoft Windows)、C:\Program Files\SQL Anywhere 17\Documentation にナビゲートします。

また、DocCommentXchange の Web で、SAP SQL Anywhere API リファレンスマニュアルにアクセスすることもできます。<http://dcx.sap.com>

上記の場所では、次のマニュアルが提供されています。

- SQL Anywhere .NET API
- SQL Anywhere C API
- SQL Anywhere JavaScript API
- SQL Anywhere Node.js API
- Dbmlsync .NET API
- Dbmlsync C++ API
- Mobile Link サーバ .NET API
- Mobile Link サーバ Java API
- Ultra Light C/C++ API
- Ultra Light Embedded SQL API
- Ultra Light for WinRT API
- Ultra Light .NET API
- Ultra Light J API

## 1.2 SQL Anywhere バージョン 17.0 の新機能 (製品全体)

SQL Anywhere 17.0 には、コンポーネントレベルの変更に加えて、知っておくべき製品全体の新機能および変更された機能がいくつか含まれています。

廃止された機能は、今後のバージョンで削除される予定です。アプリケーションを変更して、廃止された機能の代わりに推奨されている代替機能をご使用ください。

## i 注記

マニュアル完成後に追加された機能情報を含む、ソフトウェアに関する追加情報については、`readme.txt` ファイルを参照してください。

## 製品全体の新機能

### 新しい HTML-5、データベースサーバモニタツール:SQL Anywhere コックピット

コックピットはデータベースサーバの可用性、容量、およびパフォーマンスについて最新のビューを提供するデータベースモニタツールです。ブラウザで実行し、SAP Control Center に関する情報を提供することができます。

コックピットを使用して、データベースサーバにログオンした人を追跡したり、データベースサーバとクライアント統計の両方を表示したり、ユーザをデータベースから切断したりします。

コックピットはモニタする同一のデータベースサーバ上で実行します。軽量かつバックグラウンドで連続的に実行するように設計されているため、常に使用可能です。しかしながら、問題を調査するために簡単に開始や停止をすることもできます。

コックピットでは、バージョン 17 のデータベースサーバ上で動作するバージョン 17 およびバージョン 16 のデータベースがサポートされています。以下で説明するように、データベースユーザはコックピットに接続するために `COCKPIT_ROLE` ユーザ定義ロールが必要です。

SQL Anywhere コックピットをサポートするために、次のような変更が加えられています。

データベースサーバで **SQL Anywhere コックピット** を起動するための新しいデータベースサーバオプション `データベースサーバ` を起動するときは、`-cdb` データベースサーバオプションを指定して SQL Anywhere コックピット を起動します。

実行中のデータベースサーバ上で **SQL Anywhere コックピット** を制御するための `sa_server_option` の新しいオプション

実行中のデータベースサーバ上でコックピットを起動したり停止したりするには、`sa_server_option` システムプロシージャと `Cockpit` パラメータを使用します。

コックピットを使用するために必要な新しい `COCKPIT_ROLE` ユーザ定義ロール

コックピットに接続するには、ユーザに `COCKPIT_ROLE` ユーザ定義ロールの行使権限が必要です。新規作成されたデータベースにはユーザ定義のロールが含まれていますが、デフォルトではこのロールの行使権限がユーザに付与されていません。バージョン 17 のデータベースサーバ上で実行されているバージョン 16 のデータベースでコックピットを使用するには、`COCKPIT_ROLE` をデータベースに追加し、このロールの行使権限をユーザに付与します。

サンプルデータベースでの便宜上、`DBA` ユーザにはこのロールの行使権限が付与されています。

デフォルトでは、`COCKPIT_ROLE` ユーザ定義ロールにはコックピット内でアクションを実行するために必要なシステム権限がすべて含まれています。

コックピットを管理するためのセキュリティ機能

`-sf` サーバオプションを使用して、ユーザがコックピットの停止や起動ができるかどうかを制御します。この機能は、ユーザが `CockpitDB` オプションを使用して `sa_server_option` システムプロシージャを実行できるかどうかを制御します。

コックピットをサポートする新しいデータベースサーバプロパティ

CockpitURL データベースサーバプロパティは、コックピットにアクセスするための URL を返します。CockpitDB データベースサーバプロパティは、データベースサーバで現在使用されている一連のコックピットオプションを返します。

#### dbconsole の削除

コックピットが、ソフトウェアから削除されたグラフィカル管理ツール dbconsole の代わりとなります。

#### 新しいデータベース開発およびトラブルシューティングツールの SQL Anywhere プロファイラ

プロファイラは、データベースアプリケーションをチューニングおよびトラブルシューティングするための開発ツールです。これを使用して、データベースで現在実行中の SQL 文の迅速な表示、ストアドプロシージャの実行時間の比較、および、データベース内でオブジェクト同士がどのように作用しているかについての分析を行います。プロファイラでは、バージョン 17 のデータベースがサポートされています。

プロファイラは、SQL Central のほとんどのアプリケーションプロファイリング管理ツールの代わりとなります。

プロファイラをサポートするために、次のような変更が加えられています。

- SQL Central のアプリケーションプロファイラおよび SQL Anywhere パフォーマンスモニタがソフトウェアから削除され、プロファイラに置き換えられました。プロファイラを開き、バージョン 17 以前のデータベースに接続した場合、アプリケーションプロファイリングツールは使用できません。同様に、データベースサーバを開いた場合、**パフォーマンスモニタ**タブは削除されています。バージョン 17 以前のデータベースでこれらの機能を使用する場合は、データベースのバージョンに一致するバージョンの SQL Central を使用してください。
- データベーストレーシング機能は廃止予定です。診断トレーシングレベルの optimization\_logging および optimization\_logging\_with\_plans はソフトウェアで無視されます。SQL Central 用の SQL Anywhere プラグインの**データベーストレーシングウィザード**は削除されました。新しいプロファイラには、データベース操作を記録および表示する機能が含まれています。
- SQL Central 用の SQL Anywhere プラグインの**デッドロック**タブおよび**監査**タブが削除されました。新しいプロファイラには、ブロッキング情報とデッドロック情報の両方が表示される**ブロッキング**タブが含まれています。監査情報を生成し、CREATE TEMPORARY TRACE EVENT SESSION 文を使用して ETD ファイルを生成し、dbmanageetd ユーティリティを使用してファイルを表示することもできます。
- プロファイラに接続するには、新しい SYS\_RUN\_PROFILER\_ROLE システムロールが必要です。新規作成したデータベースには、ユーザ定義の SYS\_RUN\_PROFILER\_ROLE システムロールが含まれます。このロールは、プロファイラに接続するために必要です。デフォルトでは、ユーザにこのロールの行使権限は付与されません。サンプルデータベースでの便宜上、DBA ユーザにはこのロールに対する行使権限が付与されています。を参照してください。

## 製品全体の動作の変更と廃止予定機能または削除された機能

#### デフォルトの DBA/sql ユーザの削除およびデフォルトの最小パスワード長の変更

以前は、新規データベースの作成時にデータベースの DBA ユーザを指定しなかった場合、ユーザ ID DBA およびパスワード sql でユーザが自動的に作成されていましたが、データベースサーバでユーザ ID およびパスワードの指定が求められるようになりました。さらに、ユーザの最小パスワード長が 3 から 6 に変更されましたが、データベースの作成時にこの長さを無効にできます。

これらの変更は、新しいデータベースでのセキュリティの向上を目的としています。ユーザ ID が DBA/sql のデータベースがある場合、そのユーザ ID は悪意のあるユーザがデータベースにアクセスしようとするときに試行する可能性のあるユーザログインであるため、変更してください。

## CREATE DATABASE 文

データベースの作成時に、データベースの初期 DBA ユーザのユーザ ID およびパスワードの指定に使用される、DBA USER および DBA PASSWORD 句が必要になりました。また、新しいオプション句 MINIMUM PASSWORD LENGTH が追加され、(新しい) デフォルトの最小パスワード長の 6 を無効にできるようになりました。

### 初期化ユーティリティ (dbinit)

データベースの作成時に、データベースの初期 DBA のユーザ ID およびパスワードの指定に使用される、-dba オプションが必要になりました。また、新しいオプション -mpl が追加され、オプションで (新しい) デフォルトの最小パスワード長の 6 を無効にできるようになりました。たとえば、-mpl 3 は最小パスワード長を 3 に設定します。

### min\_password\_length データベースオプション

min\_password\_length データベースオプションのデフォルトの長さが 3 ではなく 6 になりました。

### -su データベースサーバオプションとユーティリティデータベース (utility\_db)

ユーティリティデータベース (utility\_db) のパスワードの設定に使用される -su データベースサーバオプションが、password; user-ID, password; を指定するオプション、または none を受け入れるようになりました。パスワードは 6 文字以上である必要があります。この制限は、パーソナルデータベースサーバ (dbeng17) の使用時にも該当します。

### util\_db.ini ファイル

バージョン 10 で廃止予定となった util\_db.ini ファイルがサポートされなくなりました。ユーティリティデータベース (utility\_db) のパスワードを設定するには、-su オプションを使用してください。

### サンプルデータベース

このマニュアルの例およびチュートリアル全体を通して広範囲にわたって使用されるサンプルデータベース (demo.db) は、多くのソフトウェアサンプルと同様に、互換性が持続するように引き続き DBA/sql ユーザが使用されます。これは、newdemo.bat スクリプト (Linux では newdemo.sh) を使用して作成されたデータベースにも該当します。同様に、newdemo.bat を使用して作成されたデータベースの最小パスワード長は 6 ではなく 3 です。

### SQL Anywhere 17 Demo データソース

以前は、SQL Anywhere Demo ODBC データソースを使用して、管理ツール、ユーティリティ、SQL 文の demo.db データベースに接続するときは、パスワードが ODBC データソースに設定として格納されているため、パスワード (sql) を指定する必要がありませんでした。SQL Anywhere 17 Demo ODBC データソースの使用時に、データベースでパスワード sql が必要になりました。

### SQL Anywhere 17 CustDB データソース

以前は、SQL Anywhere CustDB ODBC データソースを使用して、管理ツール、ユーティリティ、SQL 文の demo.db データベースに接続するときは、ユーザ ID とパスワードが ODBC データソースに設定として格納されているため、Mobile Link ユーザ ID (ml\_server) やパスワード (sql) を指定する必要がありませんでした。SQL Anywhere 17 CustDB ODBC データソースの使用時に、データベースでユーザ ID ml\_server とパスワード sql が必要になりました。

### マニュアルの例およびチュートリアル

サンプルデータベース以外の SQL Anywhere データベースへの接続またはデータベースの作成を示す、マニュアル内の SQL、コマンドライン、またはプログラミングの例では、DBA ユーザのユーザ ID およびパスワードが指定されます。以前は、こういった例で、DBA ユーザが指定されたり、ユーザ ID およびパスワードに対して DBA/sql が指定されたりすることはありませんでした。後者の場合、この例は、新しいデフォルトのパスワード長要件の 6 を満たすために、DBA ユーザ ID およびパスワードに対して DBA/passwd を使用するように変更されました。



サンプルデータベース、`newdemo` スクリプトを使用して作成されたデータベースに接続する他のすべてのマニュアル例は、変わりません。これらのデータベースでは、引き続き DBA/sql ユーザ ID およびパスワードが使用されるからです。

これには、SQL Anywhere データベースへの接続またはデータベースの作成が行われる Mobile Link や Ultra Light のような製品におけるチュートリアルや例も含まれます。

#### Mobile Link、Ultra Light、Relay Server、SQL Remote 製品

デフォルトのパスワード長の変更や、データベース初期化時のデフォルトの DBA/sql ユーザ ID およびパスワードの省略の結果による、これらの製品に対する変更はありません。

SQL Anywhere および Mobile Link のほとんどのサンプルおよびソフトウェアに含まれるサンプルデータベースでの埋込パスワードの不使用

ソフトウェアに含まれるサンプルは、可能な限り、データベースパスワードの処理に関するベストプラクティスを示すように設計されています。セキュリティ上の理由から、アプリケーションコード内にパスワードを含めないようにしてください。それにより、ほとんどのサンプルで埋込パスワードが使用されなくなります。サンプルの構築または実行にパスワードが必要な場合は、データベースパスワードを入力する必要があります。

たとえば、多くのサンプルで、サンプルディレクトリのルートに含まれるデモデータベース (`demo.db`) が使用されます。このデータベースでは、データベース管理ユーザ ID は `DBA` で、パスワードは `sql` です (パスワードでは大文字と小文字が区別されます)。

Microsoft Visual Studio の一部のサンプルプロジェクトでは、データベース信用証明書が引き続き設定ファイルに含まれており、バックグラウンドでサンプルデータベースに接続できる必要がある、Microsoft Visual Studio のデザイン時の機能がサポートされています。この機密上重要な設定情報はソフトウェア開発の完了時にアプリケーションプロジェクトから削除するように指示することをおすすめします。

#### SSL バージョン 3.0 のサポート終了

SSL バージョン 3.0 がサポートされなくなりました。代わりに TLS バージョン 1.0 以降を使用してください。

#### OpenSSL を使用した強力な暗号化の実現

##### i 注記

これらの変更は、バージョン 12 およびバージョン 16 のサポートパッケージでリリースされました。

SQL Anywhere バージョン 16.0 (ビルド 1695 以下) およびバージョン 12.0.1 (ビルド 3986 以下) に、ソフトウェア全体を通して使用される強力な暗号化を提供する Certicom 暗号化モジュールが含まれていましたが、SQL Anywhere に、強力な暗号化のための OpenSSL 暗号化モジュールが含まれるようになりました。Certicom 暗号化モジュールは、削除されました。

次の説明を読んで、この変更によってどのような影響を受けるかを確認してください。

#### FIPS 認定暗号化で、AES を使用して暗号化される ID ファイルのプライベートキーが必要

- OpenSSL では、ID ファイルのプライベートキーに対して FIPS 認定の AES 暗号化がサポートされています。プライベートキーが 3DES で暗号化されている ID ファイルを使用しているときには、FIPS 認定の OpenSSL 暗号化モジュールを使用している新規サーバは起動しません。ID ファイルは、AES を使用して再暗号化する必要があります。これを行うには、アップグレードされた `viewcert` ユーティリティを使用して、次のようなコマンドを実行してください。

```
viewcert -p -o new-file-name -op new-password -ip old-password old-file-name
```

新旧パスワードを同じにすることもできます。

- サンプルのサーバ ID ファイル (`rsaserver.id`) およびクライアント ID ファイル (`rsaclient.id`) が、プライベートキーは 3DES でなく AES を使用して暗号化されるように変更されました。
- `trusted_certificates` を使用して指定された信頼できるルート証明書ファイルは、変更する必要がありません。
- プライベートキーが AES で暗号化されている ID ファイルを使用しているときには、Certicom 暗号化モジュールを使用しているサーバのバージョンは起動しません。

#### TLS ハンドシェイク

Certicom では、クライアントは署名チェーンの任意の証明書を信頼することができ、TLS ハンドシェイクが成功します。OpenSSL では、クライアントは署名チェーンのルート証明書を信頼する必要があります。

自己署名証明書に **Certificate Signing** 属性が設定されていることが必要

ID 暗号化オプション (`x mlsrv` オプションや `-xs dbsrv` オプションなど) を使用しているときには、自己署名証明書に Certificate Signing 属性が設定されていることが必要になりました。証明書に Certificate Signing 属性が設定されているかどうかを判別するには、`viewcert` ユーティリティを使用して、出力結果の Key Usage 部分で Certificate Signing 属性を探します。自己署名証明書に Certificate Signing 属性が設定されていない場合は、証明書を再生成してください。

証明書作成ユーティリティ (`createcert`) で 3DES ではなく AES 暗号化を使用

証明書作成ユーティリティ (`createcert`) で、サーバ ID ファイルでのプライベートキーの暗号化に、3DES 暗号化ではなく AES が使用されるようになりました。

証明書作成ユーティリティに新しいオプション `-3des` が追加されました。このオプションにより、以前にリリースされたデータベースサーバで使用可能な、3DES で暗号化されたサーバ ID ファイルが作成されます。FIPS 認定暗号化を使用する新しいサーバは、3DES で暗号化された証明書を使用して起動することができません。ただし、FIPS 認定暗号化を使用しない場合は、3DES で暗号化された証明書を使用することができます。

証明書表示ユーティリティ (`viewcert`) で 3DES ではなく AES 暗号化を使用

証明書表示ユーティリティでは、`-p` オプションを指定して出力を PEM でエンコードする場合や、`-op` オプションを指定してパスワードを設定する場合に、3DES 暗号化でなく AES が使用されるようになりました。

証明書表示ユーティリティでは、AES ではなく 3DES を使用して出力やパスワードを暗号化する新しいオプション `-3des` がサポートされます。

**32 ビット Windows の起動時にデータベースサーバで FIPS 認定暗号化ライブラリをロード**

以前は、Windows の 32 ビットのデータベースサーバには必要な場合にのみ FIPS 認定暗号化ライブラリがロードされていましたが、32 ビット Windows で起動時に常にこれらのライブラリのロードが試行されるようになりました。ロードに失敗すると、FIPS 認定暗号化の使用が試行されない限り、エラーが返されなくなりました。64 ビット Windows の 32 ビットデータベースサーバでは、FIPS 認定暗号化はサポートされなくなりました。代わりに 64 ビットデータベースサーバを使用してください。FIPS 認定暗号化は別途ライセンスが必要です。

**FIPS 認定暗号化の配備**

FIPS 認定暗号化を配備する場合、配備する新しい共有ライブラリがあります。次のファイルがソフトウェアに含まれています。フォーマットファイルの `sbgse2.dll` と `libsbgse2.so` は、ソフトウェアによってインストールされなくなりました。配備する新しいファイルは次のとおりです。

- 64 ビット Windows: `libeay32.dll`、`ssleay32.dll`、`msvcr100.dll`
- 32 ビット Windows: `libeay32.dll`、`ssleay32.dll`、`msvcr90.dll`
- Linux: `libcrypto.so` および `libssl.so`

## i 注記

Windows では、暗号化のための 32 ビットおよび 64 ビットの FIPS 認定の OpenSSL ライブラリが提供されていますが、64 ビットシステムでは 64 ビットライブラリを使用してください。

### Mobile Link 関連の変更および情報

クライアント側証明書を使用した Mobile Link サーバへの接続で、Digital Signature 証明書属性が設定されていることが必要

クライアント側証明書を使用した Mobile Link サーバへの TLS 接続で、クライアント側証明書に Digital Signature 証明書属性が設定されていることが必要になりました。属性が設定されていない場合、接続は失敗します。

証明書に Digital Signature 属性が設定されているかどうかを判別するには、証明書表示ユーティリティ (viewcert) を使用して、出力結果の Key Usage 部分で Digital Signature 属性を探します。クライアント側証明書に Digital Signature 属性が設定されていない場合は、証明書を再生成する必要があります。

FIPS 認定のエンドツーエンド暗号化で、AES を使用して暗号化されるプライベートキーが必要

-x コマンドラインオプションの e2ee\_private\_key ファイルオプションによって Mobile Link サーバに提供されているプライベートキーファイルが 3DES を使用して暗号化され、サーバで FIPS 認定暗号化が使用されている場合、AES を使用して暗号化されたプライベートキーでプライベートキーファイルを再生成する必要があります。

FIPS 認定でない TLS (HTTPS を含む) およびクライアント側証明書を使用する Mobile Link 配備を更新する方法

1. クライアント側 ID 証明書に Digital Signature 属性が設定されておらず、クライアントが Mobile Link サーバに直接接続されている場合は、Digital Signature 属性が設定されたクライアント側証明書を再生成し、配備する必要があります。
2. サーバ側バイナリを更新します。
3. クライアント側バイナリを更新します。

FIPS 認定 TLS (HTTPS を含む) およびクライアント側証明書を使用する Mobile Link 配備を更新する方法

Digital Signature 属性がクライアント側 ID 証明書にない場合は、次の手順でクライアント ID 証明書が 2 回更新されます。この手順では、クライアント側の更新とサーバ側の更新が同時に実行されるように調整することなく同期を続行できるため、混乱が少なくなります。

1. 現在のクライアント側 ID 証明書に Digital Signature 属性が設定されておらず、クライアントが Mobile Link サーバに直接接続されている場合は、Digital Signature 属性が設定されたクライアント側証明書を再生成し、配備する必要があります。
2. サーバ側バイナリを更新し (新しい FIPS 認定暗号化ライブラリを含める)、AES で暗号化されたプライベートキー付のサーバ ID 証明書を配備します。
3. クライアント側バイナリを更新し (新しい FIPS 認定暗号化ライブラリを含める)、AES で暗号化されたプライベートキー付のサーバ ID 証明書を配備します。

FIPS 認定のエンドツーエンド暗号化を使用する Mobile Link 配備を更新する方法

1. e2ee\_private\_key 暗号化オプションによって参照されるプライマリキーファイルを再生成します。
2. Mobile Link サーバを停止します。
3. Mobile Link サーババイナリを更新します。新しい必要な FIPS 認定暗号化ライブラリを含めてください。
4. 新しいプライベートキーファイルを指す (または古いファイルを置き換える) ように e2ee\_private\_key オプションを変更し、必要に応じて e2ee\_private\_key\_password を更新します。
5. Mobile Link サーバを再起動します。

## Ultra Light の変更

Windows Mobile 上の Ultra Light で、FIPS 認定暗号化がサポートされなくなりました。

### Windows レジストリエントリ内の Sybase の SAP への置換え

以前は、レジストリエントリは Software¥Sybase の下に記録されていましたが、それらのレジストリエントリが Software ¥SAP の下に表示されるようになりました。

たとえば、HKEY\_LOCAL\_MACHINE¥Software¥Sybase¥SQL Anywhere¥16.0 が HKEY\_LOCAL\_MACHINE ¥Software¥SAP¥SQL Anywhere¥17.0 になります。

64 ビット Windows の場合、Software¥Wow6432Node¥Sybase が Software¥Wow6432Node¥SAP になります。ソフトウェア内の Oracle JRE の SAP JRE への置換え

#### i 注記

この変更は、バージョン 16 のサポートパッケージでリリースされました。

以前は、ソフトウェアにクライアントが使用する Oracle JRE が含まれていましたが、ソフトウェアで SAP JRE が使用されるようになりました。

SAP JRE は Oracle JRE と実行方法が異なる場合があります。java\_vm\_options オプション (SQL Anywhere) や -sl java オプション (Mobile Link) を使用して、Java VM 起動設定を最適化します。

### JDBC アプリケーション開発に対する JDK 8 の推奨

以前は、Java Development Kit (JDK 7) のバージョン 1.7.0 が JDBC アプリケーションでの使用に推奨されていましたが、JDK 8 が推奨されるようになりました。

### Windows XP のサポート終了

Windows XP で SQL Anywhere がサポートされなくなりました。

### Windows Vista のサポート終了

Windows Vista で SQL Anywhere がサポートされなくなりました。

## 1.3 SQL Anywhere サーバ 17.0 の新機能

SQL Anywhere 17.0 では、多くの新機能が導入され、プログラミングインタフェース、セキュリティ、パフォーマンス、SQL サポート、カタログ変更、管理ツールの領域のソフトウェア全体にわたって変更が加えられています。

#### i 注記

マニュアル完成後に追加された機能情報を含む、ソフトウェアに関する追加情報については、`readme.txt` ファイルを参照してください。

すべての SQL Anywhere サポートパッケージとパッチレベルアップグレードに関する `readme.txt` ファイルは、<http://sqlsupport.sap.com/readme/index.html> にあります。

## 上位機能

### OData のサポート (データベースのアップグレードが必要)

SQL Anywhere データベースサーバが OData サーバとして動作可能

この機能は OData サーバユーティリティに代わるものです。

OData プロデューサ情報がデータベースに格納され、OData サーバ情報がネットワークプロトコルオプションによって指定されます。

この機能をサポートするために、データベースサーバに次のような変更が加えられています。

#### 新しいシステム権限

- MANAGE ODATA システム権限
- VERIFY ODATA システム権限

#### 新しい SQL 文と強化された SQL 文

- CREATE ODATA PRODUCER 文。
- ALTER ODATA PRODUCER 文。
- COMMENT 文。
- DROP ODATA PRODUCER 文。

#### 新しいシステムビュー

- SYSODATAPRODUCER システムビュー。

#### 新しいデータベースサーバプロパティ

- ODataAddresses
- ODataSecureAddresses

#### -xs データベースサーバオプションの強化

-xs データベースサーバオプションで ODATA プロトコルがサポートされるようになりました。

#### 新しい OData プロトコルオプション

- ExitOnError (EXIT) プロトコルオプション。
- HttpMyIP (ME) プロトコルオプション。
- SecureMyIP プロトコルオプション。
- LogFile (LOG) プロトコルオプション。
- LogVerbosity プロトコルオプション。
- QuietConsole (QUIET) プロトコルオプション。
- SecureServerPort (HTTPSPORT) プロトコルオプション。
- ServerPort (PORT) プロトコルオプション。
- SSLKeyStore (KEYSTORE) プロトコルオプション。
- SSLKeyStorePassword (KEYSTOREPASSWORD) プロトコルオプション。

### OData プロデューサのサービス操作のデータベースカラム名

ServiceOperationsColumnNames オプションでは、ReturnType で使用される ComplexType のプロパティ名を指定するときに、サービス操作がデータベースの結果セットカラム名を使用することができます。デフォルトの動作では、サービス操作によって返された複合型のプロパティの名称が生成されます。

### CSRF トークン

新しい CSRFTokenTimeout オプションでは、この機能が有効になり、CSRF トークンが有効な秒数が表されます。CSRF トークン機能により、OData プロデューサがクロスサイトリクエストフォージェリ攻撃から保護されます。

#### 繰返可能要求

繰返可能要求機能を使用すると、クライアントが OData プロデューサとの信頼性の低い HTTP 通信を処理することができます。OData プロデューサがデータ変更要求に対する応答の受信に失敗した場合に、クライアントはデータベース破損のリスクを冒さずに要求を繰り返すことができます。

#### OData プロデューサのサービスモデルに対する OSDL のサポートの拡張

新しい OData Service Definition Language (OSDL) 構文では、次のタスクの実行機能が追加されています。

- OData プロデューサを通じて公開されているテーブルの名称を明示的に設定する
- カラムを明示的に含めたり除外したりする
- キーが生成されたエンティティセットを定義する
- 基盤となる関係テーブルを使用する複雑な関係を含む、エンティティ間の関係を定義する
- ナビゲーションプロパティを定義する
- オプティミスティック同時実行制御のエンティティで同時実行トークンを定義する
- OData プロデューサを通じて公開するサービス操作を定義する

#### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### 複数の OData プロデューサのサポート

OData サーバで、複数のデータベース接続を確立できる、カスタマイズ可能な複数の OData プロデューサがサポートされるようになりました。OData サーバ設定で新しい Embedded HTTP サーバオプション Producers を使用して、OData プロデューサを作成します。

#### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### オプティミスティック同時実行制御 (ETags)

OData プロデューサで、OData 仕様のバージョン 2.0 で定義されたオプティミスティック同時実行制御がサポートされるようになりました。ENTITY OSDL 文の concurrencytoken 句は、インスタンスの要求と同時にエンティティインスタンスの状態を識別する ETags を生成するために使用されます。

ETag の生成に使用される SQL は、SHA256 ハッシュ関数を使用し、同時実行トークンに含まれているプロパティ (カラム) の型とプロパティの数を考えると複雑になる場合があります。

#### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### サービス操作と有効な OData 識別子

OData プロデューサは、@ 文字で始まるパラメータを備えたほとんどのデータベースプロシージャおよび関数に対処できます。

サービス操作名とパラメータ名は、@name パラメータを除いて、有効な OData 識別子である必要があります。

#### ディープ挿入

OData プロデューサーで、ディープ挿入要求がサポートされるようになりました。ディープ挿入はエンティティを挿入する要求で、一部のリンクが既存のエンティティへの参照ではなく、挿入も行われる新しいインラインエンティティを含みます。ディープ挿入では、10 重にネストさせることが可能です。

ポイントインタイムリカバリ (PITR) に対するサポートの追加 (データベースのアップグレードが必要)

指定したタイムスタンプやトランザクションログのオフセットにデータベースをリストアします。この機能を使用できるようにするために、次のような変更が加えられています。

-ad データベースオプションの強化

データベースのリストア時にトランザクションログの場所に対して複数のディレクトリを指定します。これを行うには、ディレクトリ名の間にはデリミタとしてセミコロンを使用します。

新しい -ruo データベースオプション

このオプションを使用して、トランザクションログの指定したオフセットに復元します。

新しい -ru データベースオプション

このオプションを使用して、指定したタイムスタンプに復元します。

いくつかのコマンドラインユーティリティでの時間関連のレポート情報の増加

トランザクションログユーティリティ (dblog)、ログトランザクションユーティリティ (dbtran)、情報ユーティリティ (dbinfo) で、出力に含まれる時間関連の情報が増えました。

いくつかの新しいデータベースプロパティ データベースプロパティ CurrentTimelineID、CurrentTimelineSignature、PreviousTimelineID、TimelineBranchOffset が追加されました。ただし、内部使用専用です。

ログ変換ユーティリティ (dblog) の新しい -ft オプション

ログ変換ユーティリティ (dblog) に、SQL Remote 統合データベースを再ロードするときにタイムラインを指定するための新しいオプション -ft オプションが追加されました。

文のパラメータ化の改善 (データベースのアップグレードが必要)

次の強化により、アプリケーションが実行時に SQL 文の識別子およびパラメータを解決できるようになります。これらの強化により、EXECUTE IMMEDIATE 文を使用する必要性が減ることと、SQL の挿入からの保護を強化することによって、セキュリティが強化されます。また、これらの強化により、オブジェクト名または文オプションの値が実行時まで分からないときの、アプリケーションに対する柔軟性も増します。

間接識別子に対するサポートの追加

識別子構文 ('[@variable-name]') の強化を使用することによって、文内でユーザ名やオブジェクト名の変数を指定します。この方法で間接的に参照可能なサポート対象オブジェクトとして、所有者、テーブル、カラム、新しいビュー、セマフォがあります。

テーブル参照変数に対するサポートの追加 (新しい TABLE REF データ型)

ベーステーブル、テンポラリテーブル、ビューを参照する TABLE REF データ型の変数を作成して、DML 文で関数およびプロシージャのパラメータとしてオブジェクト名を直接指定する代わりに、テーブル参照変数を参照します。

一部の文の変数に対するサポートの追加

次の文でそれらの構文のパラメータの変数がサポートされるようになりました。文の実行直前に変数が解決されます。

- ALTER DATABASE 文 (データベースのアップグレードが必要)
- CREATE MIRROR SERVER 文 (データベースのアップグレードが必要)
- ALTER MIRROR SERVER 文 (データベースのアップグレードが必要)

- CREATE SUBSCRIPTION 文 (データベースのアップグレードが必要)
- CREATE PROCEDURE 文 [Web サービス] (データベースのアップグレードが必要)
- ALTER PROCEDURE 文 (データベースのアップグレードが必要)
- CREATE FUNCTION 文 [Web サービス] (データベースのアップグレードが必要)
- ALTER FUNCTION 文 (データベースのアップグレードが必要)
- CREATE EVENT 文 (データベースのアップグレードが必要)
- ALTER EVENT 文 (データベースのアップグレードが必要)
- CREATE EXTERNLOGIN 文
- DROP EXTERNLOGIN 文
- CREATE SERVER 文
- ALTER SERVER 文
- CREATE SEMAPHORE 文
- DROP SEMAPHORE 文
- NOTIFY SEMAPHORE 文
- WAITFOR SEMAPHORE 文
- CREATE MUTEX 文
- DROP MUTEX 文
- LOCK MUTEX 文
- RELEASE MUTEX 文

ユーザロック (ミューテックスおよびセマフォ) に対するサポートの追加 (データベースのアップグレードが必要)

ユーザ定義のミューテックスおよびセマフォをアプリケーションロジックに構築して、ロック動作を実現、リソースの可用性を制御および伝達します。

この機能をサポートするために、いくつかの変更が加えられています。

- 新しいミューテックス関連の文:CREATE MUTEX、LOCK MUTEX、RELEASE MUTEX、DROP MUTEX。
- 新しいセマフォ関連の文:CREATE SEMAPHORE、WAITFOR SEMAPHORE、NOTIFY SEMAPHORE、DROP SEMAPHORE。
- すべての永続ミューテックスおよびセマフォをデータベースに格納するための新しい ISYSMUTEXSEMAPHORE システムテーブル。その内容にアクセスするには、新しい SYSMUTEXSEMAPHORE システムビューを使用します。
- データベース内のすべてのミューテックスおよびセマフォの一覧を返すための新しい sp\_list\_mutexes\_semaphores システムプロシージャ。
- 新しい 2 つのカラム LockObject および LockObjectType を追加して、ロックに関連付けられているオブジェクトの名称と型を格納するための、sa\_conn\_info システムプロシージャの強化。
- ミューテックスでのロックに関する情報を含めるための、sa\_locks システムプロシージャの強化。
- ミューテックスおよびセマフォの管理と使用をサポートする新しいシステム権限:CREATE ANY MUTEX SEMAPHORE、UPDATE ANY MUTEX SEMAPHORE、DROP ANY MUTEX SEMAPHORE。
- 2 つの新しい接続プロパティ LockObjectOID および LockObjectType を使用すると、接続がブロックされているオブジェクトのオブジェクト ID および型に対してクエリを実行することができます。

シミュレートされたタイムゾーンでのデータベースの実行に対するサポートの追加 (データベースのアップグレードが必要)

データベースがデータベースサーバのシステムタイムゾーン以外のタイムゾーンで実行されているかのように動作させたい場合に、シミュレートされたタイムゾーンを作成し、そのタイムゾーンを使用するようにデータベースを設定することができますようになりました。



この機能をサポートするために、いくつかの強化が加えられています。

- 新しい文:CREATE TIME ZONE、ALTER TIME ZONE、DROP TIME ZONE。
- COMMENT ON 文の強化された構文を使用すると、システムテーブル内にタイムゾーンオブジェクトのコメントを格納することができます。
- データベースでタイムゾーンの計算に使用されるタイムゾーンを指定する新しい time\_zone データベースオプション。
- シミュレーションされたタイムゾーンオブジェクトの定義をデータベースに格納するための新しい ISYSTIMEZONE システムテーブル。その内容にアクセスするには、新しい SYSTIMEZONE システムビューを使用します。
- 新しいデータベースプロパティ:Timezone および CurrentTimezoneOffset。
- 新しい time\_zone 接続プロパティ。
- 新しい特別値:CURRENT SERVER TIME、CURRENT SERVER DATE、CURRENT SERVER TIMESTAMP。

## Node.js

Node.js が Node Packaged Modules の Web サイトおよび GitHub からダウンロードできるようになりました。Node.js JavaScript API は、SQL Anywhere データベースへの接続、SQL クエリの発行、結果セットの取得に使用できます。

### JavaScript 外部環境に対するサポートの追加

JavaScript のスタッドプロシージャおよび関数をユーザ定義の SQL スタッドプロシージャおよび関数と同様にデータベースから呼び出すことができます。

JavaScript ドライバは SQL Anywhere に含まれるようになりました。XS JavaScript API は、データベースへの接続、SQL クエリの発行、結果セットの取得に使用できます。

### JavaScript

JavaScript ドライバは SQL Anywhere に含まれるようになりました。XS JavaScript API は、データベースへの接続、SQL クエリの発行、結果セットの取得に使用できます。

### CREATE TABLE 文の強化 (データベースのアップグレードが必要)

#### OR REPLACE 句の使用

CREATE TABLE 文で、OR REPLACE 句の使用がサポートされるようになりました。

#### 別のテーブルのスキーマに基づく新しいテーブルの作成

別のテーブルの定義に直接基づいてテーブルを作成したり、カラム、制約、および LIKE 句を追加してテーブルを複製したり、SELECT 文に基づいてテーブルを作成したりすることができるようになりました。

変数の作成または宣言時、値のキャストまたは変換時、テーブルの作成または変更時、プロシージャの作成時の、%TYPE 属性および %ROWTYPE 属性に対するサポートの追加

%TYPE 属性と %ROWTYPE 属性を使用して、他のオブジェクトのデータ型に基づいてデータ型を定義します。カラムなどのスキーマオブジェクトの作成時に、%TYPE 属性を使用して、作成または変更するオブジェクトのデータ型をテーブルまたはビュー内のカラムのデータ型に設定します。%ROWTYPE 属性を使用して、データ型をテーブルまたはビュー内のローの複合データ型に設定します。変数の作成時、%TYPE 属性と %ROWTYPE 属性を使用して、データ型を変数やカーソルなどの一時オブジェクトのデータ型に設定することもできます。

次の文および関数がこの機能によって強化されます。

- CREATE TABLE 文 (%TYPE 属性のみ)
- ALTER TABLE 文 (%TYPE 属性のみ)
- CREATE PROCEDURE 文
- ALTER PROCEDURE 文

- CREATE FUNCTION 文
- ALTER FUNCTION 文
- CREATE DOMAIN 文 (%TYPE 属性のみ)
- CREATE VARIABLE 文
- DECLARE 文
- CAST 関数 (%TYPE 属性のみ)
- CONVERT 関数 (%TYPE 属性のみ)

#### SAP HANA のサポートの強化 (データベースのアップグレードまたは再構築が必要)

SAP HANA リモートサーバの作成、変更、削除に対する次の強化が追加されました。

- CREATE SERVER 文、ALTER SERVER 文、DROP SERVER 文の代わりに、CREATE REMOTE SOURCE 文、ALTER REMOTE SOURCE 文、DROP REMOTE SOURCE 文を使用して、リモートサーバを作成、変更、削除するための SAP HANA 構文のサポート。
- サーバクラス名でアンダースコアを指定するための SAP HANA 構文のサポート。たとえば、ANAODBC または HANA\_ODBC を指定できます。
- CREATE VIRTUAL TABLE を使用してプロキシテーブルを作成するための SAP HANA 構文のサポート。

このセクションの内容:

#### [パフォーマンスの強化 \[27 ページ\]](#)

SQL Anywhere 17.0 では、パフォーマンスがさまざまな形で強化されています。

#### [SQL 文、関数、プロシージャ、データ型の変更 \[28 ページ\]](#)

SQL Anywhere 17.0 では、SQL サポートに対する多数の新機能と強化が導入されています。

#### [データ管理の変更 \[32 ページ\]](#)

SQL Anywhere 17.0 は、管理に対する数多くの新機能が導入され、変更が加えられています。

#### [カタログの変更 \[36 ページ\]](#)

SQL Anywhere 17.0 では、システムオブジェクトに変更が加えられています。すべてのカタログ変更で、データベースのアップグレードが必要です。

#### [プログラミングインタフェースへの変更 \[37 ページ\]](#)

SQL Anywhere 17.0 は、プログラミングインタフェースの分野で新しい機能の導入と変更を行いました。

#### [管理ツールの変更 \[40 ページ\]](#)

SQL Anywhere 17.0 では、いくつかの新しいグラフィカル管理ツールおよび変更されたグラフィカル管理ツールが導入されています。

#### [OData サポートへの変更 \[43 ページ\]](#)

SQL Anywhere 17.0 では OData サポートに対する強化が導入されています。

#### [セキュリティへの変更 \[45 ページ\]](#)

SQL Anywhere 17.0 は、セキュリティに対する数多くの新機能が導入され、変更が加えられています。

#### [マニュアルの強化 \[48 ページ\]](#)

SQL Anywhere 17.0 は、マニュアルも強化されています。

#### [その他の変更点と強化点 \[48 ページ\]](#)

SQL Anywhere 17.0 一般的な強化がさまざまに施されています。

#### [動作変更、廃止機能、およびサポートを終了した機能 \[52 ページ\]](#)

SQL Anywhere 17.0 は動作に変更が施されています。一部には廃止された機能やサポートを終了した機能もあります。

## 1.3.1 パフォーマンスの強化

SQL Anywhere 17.0 では、パフォーマンスがさまざまな形で強化されています。

### リカバリ中のパフォーマンスの向上

データベースのリカバリ中のパフォーマンスを向上させるために、いくつかの強化が加えられています。リカバリの高速化は、次の要因によって異なります。

負荷の同時実行レベル 負荷の同時実行レベルが大きいほど、リカバリ時間が短縮されます。

リカバリを実行するマシンで使用可能なハードウェアリソース CPU コアおよびディスク I/O の使用可能なスピンドルが多いほど、リカバリに必要な時間が短くなります。

トランザクションログでの DDL 操作と DML 操作の組合せ DML 操作は、シリアル化が必要でデータベースのリカバリにかかる時間を増大させる DDL 操作よりも、並列リカバリによる利点を得ることができます。

プライマリキーの存在 プライマリキーのあるテーブルでの操作では、リカバリプロセスが大幅に高速化します。

これらのデータベースリカバリの改善は、次の機能に影響を及ぼします。

データベースミラーリング ミラーサーバは、ログ操作が順次ではなく並行して適用されるため、プライマリサーバで発生した変更に基づいて適応できるようになりました。

通常のデータベースリカバリ 通常のデータベースリカバリが改善されました。さらに、サーバスイッチ -a を使用したログ操作の適用も改善されました。

SQL Anywhere On-Demand Edition SQL Anywhere On-Demand Edition では、データベースミラーリングを使用して、クラウドインフラストラクチャでの変更を送信します。これらの変更によってプライマリサーバでの待機時間が短縮され、クラウドインフラストラクチャ全体での応答時間も短縮されます。

ミラーログを実行するデータベースの起動速度の短縮 ミラーログを実行するデータベースの起動時間を短縮するために、いくつかの内部的な改善が行われました。

### クエリ内並列処理の強化

クエリ内並列処理に内部的な強化が加えられています。この機能を使用できるようにするために、max\_query\_tasks オプションを 1 よりいっくらか高い値に設定することを検討してください。このオプションを 0 に設定して、データベースサーバがクエリ単位での数を決定できるようにすることもできます。

### サーバおよびクライアントのプランのキャッシュの改善

以前は、プランのキャッシュは、プロシージャ内の文、および単純なバイパス文のためだけに限定されていましたが、すべての文が (潜在的に) プランのキャッシュの条件を満たし、定数値が含まれるクライアントでの文をそれぞれのプランの再利用の可能性を増やすために自動的にパラメータ化できるようになりました。プランのキャッシュ機能の堅牢性を大幅に向上させるための強化も加えられています。

### 新しい sp\_plancache\_contents システムプロシージャ

キャッシュ内のプランの実行にかかった時間についての統計など、接続のプランのキャッシュの内容を返します。

### 新しい MANAGE CACHED PLANS システム権限

ユーザが任意の接続についてのキャッシュされたプランに関する統計にアクセスできます。

### 新しい parameterization\_level データベースオプション

クライアントでの文のパラメータ化のレベルを設定します。SELECT 文で新しいクエリヒント `parameterization_level` を使用することによって、個々のクエリに対するこのオプションの設定を無効にすることができます。このオプションを追加することにより、クエリを実行する対応する接続プロパティが追加され、現在のパラメータ化設定が返されます。  
**INSERT 文、UPDATE 文、DELETE 文での新しい `parameterization_level` クエリヒント**

文内で新しい `parameterization_level` クエリヒントオプションを使用することによって、INSERT 文、UPDATE 文、DELETE 文のパラメータ化レベルを設定できるようになりました。

**新しい `ParameterizationPrepareCount` サーバ、データベース、接続プロパティ**

自動的にパラメータ化された文に対して行われた準備要求の数を反映したカウンタ値を返します。  
**グラフィカルなプランの変更**

文がデータベースサーバによってパラメータ化されている場合、グラフィカルなプランには、元の SQL 文ではなく、パラメータ化されたバージョンの SQL 文が表示されます。

**クエリの最適化の改善**

プリフェッチに対する強化が加えられ、インデックススキャン時のクエリパフォーマンスが向上しました。  
**トポロジを考慮したスケジューリング**

トポロジを考慮したスケジューリングでは、同じソケットの別のコアの使用しようとする前に、ソケットごとに1つのコアを使用するようにタスクがスケジュールされます。`sa_server_option` システムプロシージャを使用して、トポロジを考慮したスケジューリングをオン（デフォルト設定）またはオフに設定します。トポロジを考慮したスケジューリングでは、一定の負荷に対するパフォーマンスが向上する可能性があります。負荷にデータ競合がある場合、トポロジを考慮したスケジューリングによってパフォーマンスが向上する可能性はないため、トポロジを考慮したスケジューリングをオフにすることができます。

**テンポラリテーブル ID の使用における改善**

以前は、トランザクション中に使用されるテンポラリテーブルはテーブル ID に割り当てられていましたが、データベースサーバではテンポラリテーブル ID の数が制限され、トランザクション中にテンポラリテーブル ID が不足する可能性があります。データベースサーバが、正確さやパフォーマンスに影響を与えずに、テンポラリテーブル ID を再利用できる（再利用する価値がある場合）ようになったため、ID が不足する可能性が低くなりました。

**キャッシュマネージャの改善**

以前は、最大キャッシュサイズが大きい場合は初期キャッシュサイズも大きくなっていましたが、初期キャッシュサイズが最大キャッシュサイズの設定の影響を受けなくなりました。

## 1.3.2 SQL 文、関数、プロシージャ、データ型の変更

SQL Anywhere 17.0 では、SQL サポートに対する多数の新機能と強化が導入されています。

廃止機能は、今後のバージョンでサポートされなくなる予定です。アプリケーションを変更して、廃止機能の代わりに推奨されている代替機能をご使用ください。

### SQL 文

これらの構文の強化のいずれかを使用するには、データベースをアップグレードするか、バージョン 17 のデータベースサーバでデータベースを初期化する必要があります。

## LEAVE 文と CONTINUE 文の構文レベルが任意に (データベースのアップグレードが必要)

SQL プロシージャ、トリガ、バッチで LEAVE 文と CONTINUE 文を実行するときに、文レベルを指定する必要がなくなりました。LEAVE 文の場合、文レベルを指定する必要がないと、最も内側のループが終了します。

## SQL プロシージャ、トリガ、バッチでの GOTO 文のサポート (データベースのアップグレードが必要)

以前は、GOTO 文がサポートされているのは Transact-SQL プロシージャ、トリガ、バッチのみでしたが、どのプロシージャ、トリガ、バッチでも GOTO 文を使用できるようになりました。を参照してください。

## CALL 文の強化

CALL 文で AS USER...IDENTIFIED BY 句の使用がサポートされるようになりました。AS USER...IDENTIFIED BY 句を使用すると、別のユーザとしてストアプロシージャを実行できます。を参照してください。

## CREATE DOMAIN 文および ALTER DOMAIN 文の変更による ROW ドメインまたは ARRAY ドメインの作成のサポート

以前は、SQL 文で ROW および ARRAY を指定する場合は、ROW および ARRAY を使用するたびに ROW または ARRAY の完全な定義をインラインで繰り返す必要がありましたが、ドメインを作成して、ROW または ARRAY の定義を保持し、SQL 文でドメインを指定できるようになりました。CREATE DOMAIN 文と ALTER DOMAIN 文が拡張され、定義内のデータ型として ROW または ARRAY の指定がサポートされるようになりました。例:

```
CREATE DOMAIN MyRow ROW( a INT, b INT );
```

## SELECT 文および FETCH 文の変更によるロー変数の使用

SELECT 文に、ロー変数の指定をサポートするための INTO VARIABLE 句と、新しいテーブルの明示的な作成をサポートするための INTO TABLE 句が含まれるようになりました。FETCH 文を使用して、INTO 句のロー変数を指定できるようになりました。

## Transact-SQL プロシージャおよびバッチでのデリミタとしてのセミコロンをサポートの追加 (アップグレードが必要)

Transact-SQL のプロシージャ、ユーザ定義の関数、文のバッチで、文のデリミタとしてセミコロンがサポートされるようになりました。

この強化を使用するには、以前のデータベースをアップグレードする必要があります。

## ロールの行使権限を取り消すためのサポートの追加 (REVOKE ROLE 文) (アップグレードまたは再構築が必要)

以前は、ユーザまたはロールからロールを取り消す場合は、行使権限と管理権限の両方が必要でしたが、REVOKE ROLE 文の新しい { ADMIN | EXERCISE } OPTION FOR 句を使用することによって、行使権限、管理権限、またはその両方 (デフォルト) のどれを取り消すかを指定できるようになりました。

## SET MIRROR OPTION 文の強化

この文で新しい max\_logfile\_size オプションがサポートされるようになりました。

## CREATE EVENT 文の強化

### OR REPLACE 句の使用

CCREATE EVENT 文で、OR REPLACE 句の使用がサポートされるようになりました。

## EXTERNAL NAME 句の強化

EXTERNAL NAME 句で、汎用オペレーティングシステム、特定のオペレーティングシステム、プロセッサのアーキテクチャなど、ライブラリのより細密な指定がサポートされるようになりました。EXTERNAL NAME 句は、ALTER TEXT CONFIGURATIONC 文、CREATE FUNCTION 文、CREATE PROCEDURE 文の一部です。

## UNLOAD 文の強化

アンロード中の @@rowcount グローバル変数の増化

UNLOAD 文で、アンロードされるローの数が @@rowcount 変数に設定されるようになりました。を参照してください。

テーブルのアンロード

UNLOAD TABLE 文が拡張され、QUOTES 句、新しい WITH COLUMN NAMES 句で ALL の使用がサポートされるようになりました。を参照してください。

## システムプロシージャ

システムプロシージャとユーザ定義のプロシージャで、実行中に、変更、置換、削除が可能に

以前は、実行中のプロシージャを変更、置換、削除しようとする、エラーが発生しましたが、変更、置換、削除が成功するようになりました。現在の実行では、プロシージャが実行を開始したときからプロシージャの定義が使用されます。変更、置換、削除後のプロシージャに対する新しい呼出では、新しい定義が使用されます。単一のトランザクションまたは接続でプロシージャが呼び出され、そのプロシージャが実行中に変更されると、同じトランザクションまたは接続で再度同じプロシージャが呼び出された場合に、プロシージャは変更された定義を使用して実行されます。

**sa\_db\_option** システムプロシージャの強化 (データベースのアップグレードが必要) sa\_db\_option システムプロシージャの val パラメータで、CHAR (128) ではなく LONG VARCHAR データ型が受け入れられるようになりました。

新しい **sp\_disk\_info** システムプロシージャ (データベースのアップグレードが必要)

sp\_disk\_info システムプロシージャは、特定のファイルパスまたはディレクトリパスのディスクドライブ情報を返します。sp\_disk\_info システムプロシージャを実行するには、新しい ACCESS DISK INFO システム権限が必要です。ユーザに sp\_disk\_info システム権限を実行させないようにするために、local\_io セキュリティ機能セットの一部である新しい sp\_disk\_info セキュリティ機能が追加されました。

**sa\_get\_dtt** システムプロシージャの強化 (データベースのアップグレードが必要)

sa\_get\_dtt システムプロシージャが、ランダムアクセスが行われるディスクの I/O キューの未処理の I/O の数を返すようになりました。これにより、ハードウェアによってはパフォーマンスが大幅に向上する可能性があります。この強化の利点を得るには、データベースを再調整してください。

新しい **sa\_list\_statements** システムプロシージャ (データベースのアップグレードが必要)

sa\_list\_statements システムプロシージャは、現在の接続で使用されている文の一覧を返します。

新しい **sp\_read\_etd** システムプロシージャ (データベースのアップグレードが必要)

sp\_read\_etd システムプロシージャは、指定されたイベントトレースデータ (ETD) ファイルを読み込み、ファイルの内容をローのセットとして返します。

**sa\_server\_option** システムプロシージャの強化 (データベースのアップグレードが必要)

sa\_server\_option システムプロシージャが、val パラメータの LONG VARCHAR 値を受け入れるようになりました。トポロジを考慮したスケジューリングもサポートされるようになりました。トポロジを考慮したスケジューリングでは、同じソケットの別のコアの使用しようとする前に、ソケットごとに 1 つのコアを使用するようにタスクがスケジュールされます。

**xp\_startsmtp** システムプロシージャの強化 (データベースのアップグレードが必要)

xp\_startsmtp システムプロシージャが、SMTPS をサポートするデータベースサーバへの SMTPS 接続を行うようになりました。

**xp\_getenv** システムプロシージャの強化 (データベースのアップグレードが必要)

xp\_getenv システムプロシージャが変更され、呼出側/定義者設定に関係なく呼出側の権限で実行されるプロシージャになりました。このプロシージャは LONG NVARCHAR 値を返します。以前は、LONG BINARY 値を返していました。

### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

**sa\_cpu\_topology** システムプロシージャの強化 (データベースのアップグレードまたは再構築が必要)

sa\_cpu\_topology システムプロシージャが変更され、-gta データベースサーバオプションまたは ProcessorAffinity サーバプロパティを使用して指定されたユーザ選択の物理プロセッサに関する情報が含まれるようになりました。この機能により、実行時にデータベースサーバが新たに追加されたプロセッサを使用できます (ホットアドおよびホットリムーブとも呼ばれます)。

-gt、-gta、-gtc データベースサーバオプションに対する制限は維持されます。

### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

**sp\_list\_directory** システムプロシージャの強化 (データベースのアップグレードが必要)

sp\_list\_directory システムプロシージャが、指定した場所のすべてのファイルおよびディレクトリのパス、ファイルタイプ、ファイル名が返されることに加えて、ファイルの作成日付、最終変更日付、最終アクセス日付のほか、所有者およびそのファイルやディレクトリに必要なアクセス権も返されるようになりました。

## 関数

関数の実行中に、変更、置換、削除が可能に

以前は、実行中の関数を変更、置換、削除しようとする、エラーが発生しましたが、変更、置換、削除が成功するようになりました。現在の実行では、関数が実行を開始したときから関数の定義が使用されます。変更、置換、削除後の関数に対する新しい呼出では、新しい定義が使用されます。単一のトランザクションまたは接続で関数が呼び出され、その関数が実行中に変更されると、同じトランザクションまたは接続で再度同じ関数が呼び出された場合に、関数は変更された定義を使用して実行されます。

新しい **READ\_SERVER\_FILE** 関数

READ\_SERVER\_FILE 関数を使用して、サーバ上の指定したファイルからデータを読み込み、ファイルの完全な内容または一部の内容を LONG BINARY 値として返します。

**EXTENDED\_PROPERTY** 関数についての新しい文書

EXTENDED\_PROPERTY 関数では、特定のデータベースサーバプロパティの値を返し、プロパティ固有の文字列パラメータをオプションで指定することができます。

以前は、この関数は内部使用のみが可能であり、マニュアルには記載されていませんでした。

## データ型

**BYTE** 文字列のサポートの強化

BYTE 文字列のサポートを強化するために、次のような機能強化と変更が加えられています。

#### 新しい BYTE\_STUFF 関数

1つの文字列から複数のバイトを削除して、別のバイトに置き換えます。

#### 新しい BYTE\_INSERTSTR 関数

別の文字列のバイト単位で指定された位置に文字列を挿入します。

#### 新しい BYTE\_LOCATE 関数

BYTE 文字列を検索して別の BYTE 文字列で置換し、新しい結果を返します。

#### 新しい BYTE\_REPLACE 関数

異なる文字列内のある BYTE 文字列の位置を返します。

#### BYTE\_SUBSTR 関数の変更

以前は、この関数は、入力文字列式の型に応じて、BINARY、LONG BINARY、VARCHAR、LONG VARCHAR、NVARCHAR、LONG NVARCHAR を返していましたが、BINARY または LONG BINARY のみを返すようになりました。

#### REPLACE 関数の変更

この関数は LONG BINARY を返すことができるようになりました。このプロシージャにバイナリ文字列を入力した場合、この関数は新しい BYTE\_REPLACE 関数と同様に動作します。

#### LOCATE 関数の変更

このプロシージャにバイナリ文字列を入力した場合、この関数は新しい BYTE\_LOCATE 関数と同様に動作します。

#### STUFF 関数が LONG BINARY を返すように変更

以前は、STUFF 関数と INSERTSTR 関数は入力式が BINARY の場合でも LONG NVARCHAR または LONG VARCHAR を返していましたが、入力式が BINARY の場合は LONG BINARY を返すようになりました。

#### INSERTSTR 関数が LONG BINARY を返すように変更

以前は、INSERTSTR 関数は入力式が BINARY の場合でも LONG NVARCHAR または LONG VARCHAR を返していましたが、入力式が BINARY の場合は LONG BINARY を返すようになりました。

## 1.3.3 データ管理の変更

SQL Anywhere 17.0 は、管理に対する数多くの新機能が導入され、変更が加えられています。

#### ユーザ定義のアップグレードに対するサポートの追加 (データベースのアップグレードが必要)

SQL Anywhere データベースに対する独自のアップグレード SQL スクリプトを作成および実行して、データベースをアップグレードします。この機能は、ALTER DATABASE...UPGRADE 文、および SQL Central の [データベースアップグレードウィザード](#) に用意されている新しい構文 (SCRIPT FILE) を使用することによってサポートされます。

#### 最短ダウン時間での運用データベースの再構築に対するサポートの追加

データベースの動作中に運用データベースを再構築できるようになり、ダウン時間が短縮されました。最短ダウン時間での再構築をサポートするために、いくつかの新しい項目が追加されました。

アンロードユーティリティ (`dbunload`) の新しいオプション (データベースの再構築が必要) アンロードユーティリティ (`dbunload`) で、`-ao`、`-aob`、`-aot`、`-dt` オプションがサポートされるようになりました。

バックアップユーティリティ (`dbbackup`) の新しいオプション バックアップユーティリティ (`dbbackup`) で、`-wa` オプションがサポートされるようになりました。



**db\_backup** 関数の新しい情報タイプ db\_backup 関数で新しい DB\_BACKUP\_INFO\_WAIT\_AFTER\_END 情報タイプがサポートされるようになりました。

**DBChangeLogName** メソッドでの **a\_change\_log** 値の変更 パラメータ (const a\_change\_log \*pcl) が (a\_change\_log \*pcl) に変更されました。

**a\_change\_log** 構造体での 3 つの新しいメンバーデータ項目のサポート 新しいメンバーデータ項目は、starting\_offset、current\_relative\_offset、end\_offset です。zap\_current\_offset、zap\_starting\_offset、encryption\_key メンバーは、const char\* 型になりました。

**an\_unload\_db** 構造体での新しいメンバーのサポート 新しいメンバーは、online\_rebuild、online\_rebuild\_from\_backup、shouldstoprtn、online\_rebuild\_max\_apply\_sec、temporary\_directory です。新しい処理停止コールバックルーチン 処理停止コールバックルーチンは、ツールが処理を続行すべきか停止すべきかを決定する必要がある場合に呼び出されます。

#### 検証の強化

検証ユーティリティ (**dbvalid**) でのインメモリバックアップ検証のサポート dbvalid を使用してデータベースサーバを自動起動する場合に、インメモリバックアップ検証モードを使用してバックアップ検証が実行されるように指定することができます。バックアップデータベースまたはそのトランザクションログファイルを変更せずにバックアップデータベースを検証するには、このモードを使用します。

#### インメモリデータベース検証の強化

-im データベースサーバオプションが、データベースを変更せずにデータベースのインメモリバックアップ検証を行えるように拡張されました (-im v/-im valid)。

検証の強化 VALIDATE 文、Validation ユーティリティ (dbvalid)、sa\_validate システムプロシージャが、データベース破損の誤報の可能性をなくすオプションによって強化されました。さらに、sa\_validate は、検証の成功または失敗を示すコードを、失敗した場合は無効なテーブルの名称とともに、返すようになりました。

#### VALIDATE 文での 2 つの新しい句のサポート

##### WITH DATA LOCK

指定したテーブルに排他データロックを適用することで、トランザクションによってテーブルスキーマまたはデータが修正されることを回避します。トランザクションは、テーブルデータまたはスキーマを読み込むことができますが、修正することはできません。

FOREIGN KEY 句を指定すると、排他データロックは対応するプライマリーキーを含むテーブルにも適用されます。

##### WITH SNAPSHOT

スナップショットアイソレーションを適用して、コミットされたデータのみがチェックされるようにします。トランザクションは、データを読み込むことも修正することもできます。この句を使用するには、データベースで (allow\_snapshot\_isolation データベースオプションを使用して) スナップショットアイソレーションが有効になっている必要があります。この句ではスナップショットアイソレーションを使用するため、パフォーマンスに影響することがよくあります。

#### 検証ユーティリティでの 2 つの新しいオプションのサポート

-wl 他のスレッドで検証対象のテーブル内のデータが変更されないようにしながら、テーブルに適用されている排他データロックを使用して検証を実行します。

-ws テーブルに適用されているスナップショットアイソレーションを使用して検証を実行します。

**sa\_validate** システムプロシージャの強化: sa\_validate で、新しいパラメータ **isolation\_type** がサポートされます。このパラメータを使用して、検証中にデータロックまたはスナップショットをテーブルに適用する必要があるかどうかを指定できます。

sa\_validate システムプロシージャで、検証が成功したか失敗したかの判定に使用可能なコードが返されるようになりました。また、テーブルの検証が失敗すると、sa\_validate が失敗の原因となったテーブルの名称が返されるようになりました。

#### データベースサーバプロパティ値の追跡に対するサポートの追加

データベースサーバのプロパティ値をポーリング、分析、格納することなく、指定した時間または指定した量のメモリに達するまで数値を返すデータベースサーバプロパティを追跡するように、データベースまたはデータベースサーバを設定できるようになりました。

この機能をサポートするために、次のような変更が加えられています。

#### 新しい sp\_property\_history システムプロシージャ (データベースのアップグレードが必要)

データベースによって追跡されたすべてのデータベースサーバプロパティの値を返します。

#### 新しい PROPERTY\_IS\_TRACKABLE 関数

追跡された値を格納することによって指定したデータベースサーバプロパティの履歴データを更新できるかどうかを返します。

#### sa\_server\_option システムプロシージャの新しいデータベースサーバオプション (データベースのアップグレードが必要)

- PropertyHistoryList
- PropertyHistorySize

#### sa\_db\_option システムプロシージャの新しいデータベースオプション (データベースのアップグレードが必要)

- PropertyHistoryList

#### 新しいデータベースサーバオプション

- -phi データベースサーバオプション。
- -phs データベースサーバオプション。

#### 新しいデータベースサーバプロパティ

- CacheMisses
- CompletedReq
- ConnCount
- CurrRead
- CurrWrite
- DiskWrite
- PropertyHistoryList
- PropertyHistoryListActual
- PropertyHistorySize
- PropertyHistorySizeBytes

#### 新しいデータベースプロパティ

- PropertyHistoryList

#### 新しいシステム権限

- MANAGE ANY PROPERTY HISTORY システム権限 (データベースのアップグレードが必要)

#### -sf データベースサーバオプションの強化

MANAGE\_SERVER 機能セットに MANAGE\_PROPERTY\_HISTORY 機能が追加されました。  
データベースサーバの動作中の接続リスナの開始および停止に対するサポートの追加 (データベースのアップグレードまたは再構築が必要)

データベースサーバを再起動せずに、HTTP、HTTPS、TCP/IP、共有メモリ接続リスナーを開始および停止できるようになりました。

この機能をサポートするために、いくつかの強化が加えられています。

#### システムプロシージャ

sp\_start\_listener、sp\_stop\_listener、sp\_http\_listeners の 3 つの新しいシステムプロシージャが追加されました。

#### データベースサーバプロパティ

次のデータベースサーバプロパティが追加されました。HttpListeners、HttpsListeners、TcpiListeners。

#### 新しい MANAGE LISTENERS システム権限

バージョン 16.0 以降を使用して新しいデータベースを作成した場合や、16.0 より前のデータベースをアップグレードした場合、この権限は SYS\_AUTH\_SSO\_ROLE システムロールと SYS\_RUN\_REPLICATION\_ROLE システムロールにあります。

バージョン 16.0 のデータベースをアップグレードした場合、この権限は UPGRADE ROLE システム権限に付与されます。

新しい MANAGE\_LISTENERS 機能 MANAGE\_SERVER 機能セットに MANAGE\_LISTENERS 機能が追加されました。

イベントトレースの強化 (データベースのアップグレードが必要) スケジュールされたイベント内で実行される各文のログを記録するための新しいトレースイベントが追加されました。

#### SQL Anywhere 監査の強化 (データベースのアップグレードまたは再構築が必要)

SQL Anywhere 監査機能により、ユーザが、追加イベントを追跡し、監査可能なイベントを監査ログファイルまたはシステムイベントログに記録できるようになりました。監査イベントは、システムトレースイベントのサブセットです。

強化された監査をサポートするために、次のデータベースオブジェクトが変更されています。

**sa\_enable\_auditing\_type** システムプロシージャ sa\_enable\_auditing\_type システムプロシージャで、xp\_cmdshell のログ記録の呼び出しがサポートされるようになりました。

**sa\_disable\_auditing\_type** システムプロシージャ sa\_disable\_auditing\_type システムプロシージャで、xp\_cmdshell のログ記録の呼び出しがサポートされるようになりました。

**audit\_log option** 監査ログのタイプと場所を指定できるように、audit\_log オプションが追加されました。

次のシステムプロシージャが MANAGE AUDITING システム権限を必要とし、include\_audit\_events パラメータが含まれるように

- sp\_trace\_events システムプロシージャ。
- sp\_trace\_event\_fields システムプロシージャ。
- sp\_trace\_event\_session\_events システムプロシージャ。
- sp\_trace\_event\_session\_targets システムプロシージャ。
- sp\_trace\_event\_session\_target\_options システムプロシージャ。

## i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

現在実行されている文の特定に対する強化 (sa\_stack\_track システムプロシージャではデータベースのアップグレードが必要)

以前は、データベースサーバによって最後に実行された文を返すために、LastStatement プロパティおよび関連する RememberLastStatement オプションおよびプロパティのみ使用できました。ただし、最後のクライアントでの文で別の文が実行された場合 (たとえば、クライアントでの文が別の SQL 文を実行するスタアドプロシージャを呼び出した場合など) は、結果に実行した別の文が含まれませんでした。

STACK\_TRACE 関数および sa\_stack\_trace システムプロシージャを使用して、データベースサーバで現在実行されている文に関する完全な情報を 3 つのパラメータ stack\_frames、detail\_level、connection\_id を使用して返すことができるようになりました。

データベースの自動起動の強化 (-ufd データベースサーバオプション)

致命的なエラーまたはアサーションの失敗がデータベースで発生したときにデータベースサーバが実行するアクションに対する制御が強化されました。-ufd データベースサーバオプションが、restart\_escalate および restart\_abort の 2 つの新しい値を使用する 2 つの新しい自動起動動作がサポートされるように拡張されました。既存の再起動オプションも引き続きサポートされますが、廃止予定です。代わりに同等の restart\_escalate 値を使用してください。

## 1.3.4 カタログの変更

SQL Anywhere 17.0 では、システムオブジェクトに変更が加えられています。すべてのカタログ変更で、データベースのアップグレードが必要です。

### カタログの新機能

**SYSROCPARM** システムビューがチェックポイントごとに更新されるように

以前は、SYSROCPARM システムビューは、プロシージャまたは関数の作成または変更時、あるいは ALTER PROCEDURE RECOMPILE 文の実行時にのみ更新されていました。

SYSROCPARM のプロシージャまたは関数の値がテーブル、ビュー、プロシージャなどの別のオブジェクトに依存していて、そのオブジェクトが変更された場合、それらの値は以前の値のままです。

古いプロシージャまたは関数が次の条件を満たす場合は、チェックポイントが実行されるたびに SYSROCPARM が更新されるようになりました。

- プロシージャやファンクションが変更されて以降、参照されています。
- プロシージャが RESULT 句を持つ、またはプロシージャが、RESULT 句を持たない他のプロシージャに対する呼び出しが 10 回ネストされた再帰プロシージャではない。

新しいカタログオブジェクト

このカタログには、次のオブジェクトが追加されています。

- ISYSDATABASEVARIABLE と SYSDATABASEVARIABLE

- ISYSMUTEXSEMAPHORE と SYSMUTEXSEMAPHORE
- ISYSODATAPRODUCER と SYSODATAPRODUCER
- ISYSTIMEZONE と SYSTIMEZONE

## 1.3.5 プログラミングインタフェースへの変更

SQL Anywhere 17.0 は、プログラミングインタフェースの分野で新しい機能の導入と変更を行いました。

**データベースサーバごとに1つの CLR 外部環境を使用するためのサポートの追加**

以前は、データベースサーバで、データベースサーバ上で動作しているデータベースごとに1つの CLR 外部環境が使用されていましたが、データベースサーバ上で動作しているすべてのデータベースに対して1つの CLR 外部環境が使用されるように、データベースサーバを設定できるようになりました。この機能の一部として、次の新しいプロパティとデータベースサーバオプションが追加されました。

- -sclr データベースサーバオプション
- SingleCLR データベースサーバプロパティ
- SingleCLRInstanceVersion データベースサーバオプション

**CLR 外部環境での .NET バージョンの使用に対するサポートの追加**

CLR 外部環境では、任意のバージョンの .NET を使用できます。使用する .NET のバージョンを特定するには、ALTER EXTERNAL ENVIRONMENT 文を使用します。

**データベースサーバごとに1つの Java VM を使用するためのサポートの追加**

以前は、データベースサーバで、データベースサーバ上で動作しているデータベースごとに1つの Java VM が使用されていましたが、データベースサーバ上で動作しているすべてのデータベースに対して1つの Java VM が使用されるように、データベースサーバを設定できるようになりました。この機能の一部として、次の新しいプロパティとオプションが追加されました。

- JavaVM データベースプロパティ
- JavaVM データベースサーバプロパティ
- SingleJVM データベースサーバプロパティ
- -sjvm データベースサーバオプション
- SingleJVMLocation データベースサーバオプション
- UseSingleJVMInstance データベースサーバオプション

**Embedded SQL:ワイド更新およびワイド削除に対するサポートの追加**

Embedded SQL で、EXECUTE 文 [ESQL] の ARRAY 句を使用して、ワイド更新とワイド削除がサポートされるようになりました。

**ODBC:ワイド更新およびワイド削除に対するサポートの追加**

ODBC では、SQLSetStmtAttr(SQL\_ATTR\_PARAMSET\_SIZE) と SQLBindParameter を使用したワイド更新とワイド削除がサポートされるようになりました。

**JDBC:バッチ更新およびバッチ削除に対するサポートの追加**

JDBC では、PreparedStatement.addBatch() および PreparedStatement.executeBatch() を使用したワイドマージと更新またはバッチマージと更新がサポートされるようになりました。

**SQL Anywhere C API:ワイド挿入に対するサポートの追加**

SQL Anywhere C API のバージョン 4 でワイド挿入がサポートされます。\_SACAPI\_VERSION を 4 として定義したときに、次のメソッドが使用可能になりました。

```
sqlany_set_batch_size( a_sqlany_stmt stmt, size_t num_rows )
```

このメソッドは、指定した文のバッチでのローの数を設定します。ワイド挿入の場合、これは挿入されるローの数です。デフォルト値は 1 で、単一のローが実行されます。バッチサイズを 0 に設定すると、エラーが発生します。

```
sqlany_get_batch_size( a_sqlany_stmt stmt )
```

このメソッドは、指定した文のバッチでのローの数を取得します。

```
sqlany_set_param_bind_type( a_sqlany_stmt stmt, row_size )
```

このメソッドは、ローワイズバインドの使用時のローのバイトサイズを設定します。この値により、次の境界値を検出するためにスキップするバイトの数が決まります。row\_size のデフォルト値は 0 で、カラムワイズバインドを示します。

#### SQL Anywhere C API:ワイドフェッチに対するサポートの追加

SQL Anywhere C API のバージョン 4 にワイドフェッチに対するサポートが追加されました。\_SACAPI\_VERSION を 4 として定義したときに、次のメソッドが使用可能になりました。

- sqlany\_set\_rowset\_size メソッド
- sqlany\_set\_rowset\_pos メソッド
- sqlany\_get\_rowset\_size メソッド
- sqlany\_set\_column\_bind\_type メソッド
- sqlany\_bind\_column メソッド
- sqlany\_fetched\_rows メソッド

#### SQL Anywhere C API:各国の文字型に対するサポートの追加

SQL Anywhere C API のバージョン 4 に追加された各国の文字 (NCHAR、NVARCHAR、LONG NVARCHAR、NTEXT) 型に対するサポート。この機能は \_SACAPI\_VERSION を 4 として定義したときに使用可能です。

a\_sqlany\_column\_info 構造体の native\_type 項目に指定できる値は、DT\_NVARCHAR、DT\_NFIXCHAR、DT\_NSTRING、DT\_LONGNVARCHAR です。

#### SQL Anywhere C API:コールバックに対するサポートの追加

SQL Anywhere C API のバージョン 3 にコールバックに対するサポートが追加されました。sqlany\_register\_callback 関数は \_SACAPI\_VERSION を 3 として定義したときに使用可能です。この関数はコールバック関数の登録に使用できません。

#### i 注記

この強化は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### JDBC ドライバ

エスケープシーケンスに対する JDBC ドライバのサポートが、TIMESTAMPADD 関数および TIMESTAMPDIFF 関数を含むように強化されました。

#### i 注記

この強化は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### ODBC ドライバ

エスケープシーケンスに対する ODBC ドライバのサポートが、TIMESTAMPADD 関数および TIMESTAMPDIFF 関数を含むように強化されました。

以前は、ODBC SQLGetInfo 関数を呼び出して ODBC ドライバのバージョン (SQL\_DRIVER\_VER) を取得すると、ドライバのビルド番号が含まれない文字列が返されましたが、文字列が xx.yy.zzzz の形式で返されるようになりました。xx は 2 桁のメジャーバージョン番号、yy は 2 桁のマイナーバージョン番号、zzzz はビルド番号を表します。たとえば、16.00.1234 のようになります。

#### **i** 注記

この強化は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### **.NET 2.0 のデータプロバイダの削除**

旧バージョンの SQL Anywhere には、.NET バージョン 2.0 のデータプロバイダが含まれていました。このプロバイダは削除されました。代わりに .NET バージョン 3.5 のプロバイダを使用できます。.NET バージョン 3.5、4.0、4.5 がサポートされています。

#### **空間 API:新しい ST\_Buffer メソッド**

ST\_Buffer メソッドは、geometry-expression の指定距離内にあるすべてのポイントを表す ST\_Geometry 値を返します。

#### **PHP サポート**

##### **廃止予定の PHP 関数の削除 (sqlanywhere\_\* family)**

廃止予定の PHP 関数 sqlanywhere\_commit、sqlanywhere\_connect などが削除されました。これらの関数は廃止予定で、バージョン 11 で sasql\_commit、sasql\_connect などの関数ファミリーに置き換えられました。

#### **Python サポート**

##### **Django ドライバ (sqlany-django)**

Django は、Web サイトを作成するための Python ベースのフレームワークです。新しい sqlany-django ドライバを使用すると、SQL Anywhere データベースサーバを Django ベース Web サイトのバックエンドとして使用することができます。

SQL Anywhere Django ドライバの最新のソフトウェアおよびマニュアルは、PyPI (Python Package Index) Web サイト (<https://pypi.python.org/pypi>) で入手することができます。sqlany-django を検索します。

##### **SQLAlchemy ドライバ (sqlalchemy-sqlany)**

SQLAlchemy は、Python ベースのツールキットで、オブジェクトリレーショナルマッパーです。新しい sqlalchemy-sqlany ダイアレクトを使用すると、SQL Anywhere データベースサーバとの通信が可能な SQLAlchemy アプリケーションを登録することができます。

SQL Anywhere SQLAlchemy ダイアレクトの最新のソフトウェアおよびマニュアルは、PyPI (Python Package Index) Web サイト (<https://pypi.python.org/pypi>) で入手することができます。sqlalchemy-sqlany を検索します。

## 1.3.6 管理ツールの変更

SQL Anywhere 17.0 では、いくつかの新しいグラフィカル管理ツールおよび変更されたグラフィカル管理ツールが導入されています。

廃止された機能は、今後のバージョンで削除される予定です。アプリケーションを変更して、廃止された機能の代わりに推奨されている代替機能をご使用ください。

**モニタの強化** モニタは、SQL Anywhere データベース、Mobile Link サーバ、Mobile Link サーバファームの正常性や可用性に関する情報を示します。Web サーバ、プロキシサーバ、およびホストコンピュータの可用性についての情報も提供することができます。

### Web サーバ、プロキシサーバ、ホストコンピュータのモニタ

SQL Anywhere 以外のデータベースサーバ、プロキシサーバ、ホストコンピュータを含む、Web サーバの可用性をモニタします。たとえば、ネットワーク上のプリンタの可用性をモニタするリソースを作成することができます。

データベースリソースのカスタムメトリックおよび警告の作成

関心のあるデータをモニタするために、メトリックおよび警告の独自のしきい値を作成できるようになりました。ユーザー定義のメトリックから収集されたデータを表示するウィジェットを作成することもできます。

### 警告の整理

以前のモニタでは、警告は、重大度別、時間別に表示されていました。警告を時系列および警告タイプ別にまとめることができるようになりました。警告をタイプ別にまとめると、**警告タイプの詳細**ウィンドウで、そのタイプのすべての警告が1つのカラムに表示され、警告の詳細が別のカラムに表示されます。特定のタイプのすべての警告を解決したり、アクティブとしてマーク付けしたり、このウィンドウから削除したりすることができます。

解決済の警告の非表示化

**警告リスト**ウィジェットで、解決済の警告がデフォルトで非表示になりました。

アクセシビリティ機能の強化

モニタのフォントサイズとカラーテーマを変更できるようになりました。キーボードナビゲーションも改善されました。

**SQL Anywhere モニタでの Relay Server のサポート終了**

モニタで Relay Server がサポートされなくなりました。

### Interactive SQL の強化

新しいタブ付きインターフェースを備えた **Interactive SQL** の1つのインスタンス内での複数の接続の管理

タブ付きの **Interactive SQL** の1つのインスタンス内で複数の接続を管理します。新しいタブを開くには、**ウィンドウ** ▶ **新規タブ** ▶ をクリックします。

改善された履歴タブにメッセージタブの内容が含まれるように

**Interactive SQL** の**結果**ウィンドウ枠の**履歴**タブに、履歴および実行メッセージ、警告、エラーが表示されるようになりました。**メッセージ**タブが削除されました。階層ファイルの形式が変更され、旧バージョンのソフトウェアとの互換性がなくなりました。

データベースロックが存在する場合の新しいビジュアルなインジケータ

**Interactive SQL** ウィンドウの下部にあるステータスバーに、保持されているデータベースロックの数のインジケータ、および他の接続がブロックされている場合のインジケータが表示されるようになりました。新しい**ロックビューア**ウィンドウにも、この情報が詳細に表示されます。

強化された **Microsoft Excel** との相互運用性



INPUT 文や OUTPUT 文、またはインポートウィザードやエクスポートウィザードを使用して、Microsoft Excel からのデータのインポートおよび Microsoft Excel へのデータのエクスポートをより簡単に行えるようになりました。すべての文のすべての結果を表示 以前は、Interactive SQL でデフォルトでは最後の文の最初の結果のみが表示されていましたが、実行されたすべての文の結果が結果ウィンドウ枠に表示されるようになりました。

#### グラフィカルなプランの比較

Interactive SQL のプラン比較ウィンドウを使用して、2 つのグラフィカルなプランを比較できるようになりました。Interactive SQL のプランビューアを使用して、グラフィカルなプランを作成および保存します。次に、プラン比較ツールで 2 つのプランを開き、比較します。

#### Mac OSX での Interactive SQL のショートカットの改善

Mac OSX で使用する Interactive SQL のキーボードショートカットが、Mac の手法を利用できるように更新されました。

Mac OSX のショートカット	アクション
Command+Q	Interactive SQL を閉じます。
Command+R	SQL 文ウィンドウ枠内のすべてのテキストを実行します。以前、これは F5 を使用して行っていました。
Command+Option+R	SQL 文ウィンドウ枠で選択されているテキストを実行します。
Control+R	選択された SQL 文を実行し、次の文を選択します。このショートカットを使用すると、一連の SQL 文をステップスルーできます。

Mac OSX で OMMIT 文および ROLLBACK を実行するショートカットが削除されました。

#### 実行時間の精度の改善

SQL 文の実行時間が、実行された文からアプリケーションが結果をフェッチするのにかかる時間が含まれるように改善されました。

#### テキスト補完の改善

テキスト補完機能が改善されました。たとえば、自分が所有していないテーブル名を補完するときには所有者が自動的に追加され、ブランクが含まれる識別子を補完するときには識別子が補完の一部として引用符で囲まれます。

#### 数字と日付のローカライズ

数字と日付がコンピュータの地域固有の形式を使用して表示されるようになりました。以前は、デフォルトの動作は、コンピュータのロケールにかかわらず、同じ形式でした。

たとえば、カンマが小数点として使用されるロケールで Interactive SQL が実行されている場合、テキストとしてインポートおよびエクスポートされたデータにおいてセミコロンがデフォルトの項目デリミタとして使用されます。ピリオドが小数点として使用されるロケールでは、デフォルトの項目デリミタはカンマです。

デフォルトの動作を変更するには、**ツール > オプション > インポート/エクスポート** をクリックし、テキストオプションを調整します。

#### SQL Central の強化

Sybase Central が SQL Central と呼ばれるようにグラフィカル管理ツール Sybase Central の名称が SQL Central に変更されました。

## データベースオブジェクトのフィルタリング

右ウィンドウ枠の右上隅にある**フィルタ**項目を使用して、右ウィンドウ枠に表示されるデータベースオブジェクトの一覧をフィルタリングします。**フィルタ**項目では、特定のテキストを含む項目をフィルタリングします。

## フォルダの作成によるデータベースオブジェクトの整理

左ウィンドウ枠のシステム定義のデータベースオブジェクトフォルダに加えて、独自のフォルダを作成することができます。フォルダを作成してデータベースオブジェクトをマニュアルで追加するか、作成する式に基づいて動的に移植および更新されるスマートフォルダを作成することができます。すべてのフォルダにデータベースオブジェクトが1タイプずつ保存されます。

## データベーススキーマの比較ツールの新しいオプション

以前は、データベーススキーマの比較時にパスワード項目の内容が常に含まれていましたが、データベースの比較時にパスワードを比較するかどうかを指定できるようになりました。

## SQL Anywhere のレジストリキーを使用した表示言語の上書

SQL Central では、SQL Anywhere のレジストリキーの SQL Anywhere 実行プログラム...¥Software¥SAP ¥SQL Anywhere¥17.0 および SALANG 環境変数が優先されます。以前は、SQL Central でレジストリキー ... ¥Software¥SAP¥SQL Central¥17.0.0 の言語レジストリ値が使用されていました。このレジストリキーは廃止されました。

## Relay Server プラグインのサポート終了

SQL Central の Relay Server プラグインがサポートされなくなりました。

## Windows 用 Deployment ウィザードの強化

### Windows 用 Deployment ウィザード

Windows 用 Deployment ウィザードで、アップグレードインストールイメージを作成できるようになりました。このタイプのインストールは、ソフトウェアの既存のインストールを更新することを目的としています。

## その他の廃止機能

### SAP JRE のバージョン 7 からバージョン 8 への更新

以前は、Java Run-time Edition (JRE 7) のバージョン 1.7.0 がソフトウェアに含まれていましたが、管理ツールで、SAP の Java Run-time Edition 1.8.0 (JRE 8) が使用されるようになりました。

**SQL Central** および **Interactive SQL** の高速ランチャオプションの削除 これらのグラフィカル管理ツールの起動時間を短縮していた SQL Central または Interactive SQL の高速ランチャオプションが削除されました。最近のコンピュータではこの機能の意味がなくなっています。

### [接続] ウィンドウの変更

接続ウィンドウで指定した情報に基づいて ODBC データソースを作成していた **ODBC データソースとして保存** オプションが削除されました。

グラフィカル管理ツールのアップグレードにより、**Java Runtime Edition 1.8.0 (JRE 8)** を使用 Java が使用されるグラフィカル管理ツールで、Java Runtime Edition 1.8.0 (JRE 8) が使用されるようになりました。

## 1.3.7 OData サポートへの変更

SQL Anywhere 17.0 では OData サポートに対する強化が導入されています。

### OData の強化

#### ログ生成の強化

生成されたログで、OData プロデューサと、各ログイベントに関連付けられた要求が識別されるようになりました。

#### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### ConnectionPoolMaximum

OData プロシージャの ConnectionPoolMaximum のデフォルト設定オプションが、データベースサーバの最大接続の半分に設定されます。

#### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### 関連付けの改善

関連付けがメタデータの参照整合性制約として注釈が付けられるようになり、参照整合性制約の関連付けプロパティがデフォルトで表示されるようになりました。

関連付けにはメタデータの OnDelete 属性を指定することができ、これにより、関連付けられているプリンシパルエンティティインスタンスが削除されたときに、依存するエンティティインスタンスがどのように影響を受けるかが説明されます。

#### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### メタデータキャッシュの改善

OData プロデューサでは、プロデューサの潜在ユーザから同一のアクセスがあるユーザのメタデータをキャッシュするときは、使用するオブジェクトが少なくなります。

#### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### OSDL ファイルの強化

OSDL ファイルで、引用符で囲まれた文字列に追加されたエスケープシーケンスがサポートされます。

#### i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### モデルパスでのサーブレットコンテキストの使用

OData プロデューサオプションのパスとして**モデル**を指定すると、パスの基準がサーブレットのコンテキストまたはサーバの現在のディレクトリになります。パスおよびファイル名は、`${variable-name}` 形式の環境変数参照用に処理されません。

#### **i** 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### プロキシテーブルの挿入

SQL Anywhere データベースのプロキシテーブルであるエンティティセットでエンティティを作成するときは、すべてのキープロパティを明示的に指定する必要があります。

#### **i** 注記

この制限は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### substringof 戻り値

substringof(s1, s2) フィルタは、s1 文字列が s2 のサブ文字列であるかどうかを返します。

#### **i** 注記

この変更は、バージョン 16 のサポートパッケージで最初にリリースされました。

## OData の動作の変更

#### 検索文字列の 254 バイトへの制限

startswith、substringof、indexof などの OData フィルタ関数が含まれる長い検索文字列を使用すると、検索は最初の 254 バイトに対してのみ実行されます。

#### **i** 注記

この制限は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### OData サーバユーティリティおよび OData サーバ停止ユーティリティのサポート終了

データベースサーバを OData サーバとして起動できます。OData プロデューサ情報は、データベースに格納されます。

Windows 用サービスユーティリティ (dbsvc) で、-t OData オプションがサポートされなくなりました。

#### OData サーブレットのネームスペースの変更

OData サーブレットのネームスペースが com.sap.odata.producer.servlets.ODataServlet に変更されました。メタデータのデフォルトのネームスペースが SAPSQLData に変更されました。

#### StartsWith() 文字列フィルタでの 254 文字の制限の緩和

検索文字列がリテラルであるか、長さが 126 未満の括弧で囲まれたリテラルである場合に、StartsWith が LIKE を使用するために最適化されました。

## i 注記

この機能は、バージョン 16 のサポートパッケージで最初にリリースされました。

## 1.3.8 セキュリティへの変更

SQL Anywhere 17.0 は、セキュリティに対する数多くの新機能が導入され、変更が加えられています。

### データベース内のパスワード情報の保護の大幅な強化

このリリースでは、カタログ内およびデータベース操作中のパスワード情報を保護するために、いくつかの変更が加えられています。次に新機能の一覧を示します。追加情報がカタログ変更セクションと動作変更セクションに一覧表示されています。

#### 新しい ACCESS USER PASSWORD システム権限

このシステム権限により、ユーザが、パスワードハッシュを表示し、データベースのアンロード、抽出、比較など、パスワードへのアクセスを伴う操作を実行するビューにアクセスできるようになります。

バージョン 16.0 以降を使用して新しいデータベースを作成した場合や、16.0 より前のデータベースをアップグレードした場合、この権限は SYS\_AUTH\_SSO\_ROLE システムロールと SYS\_RUN\_REPLICATION\_ROLE システムロールに付与されます。

バージョン 16.0 のデータベースをアップグレードした場合、この権限は UPGRADE ROLE システム権限に付与されます。

#### アンロードユーティリティ (dbunload) および抽出ユーティリティ (dbextract) の新しい -up オプション

このオプションでは、パスワードをアンロードできます。このオプションの使用に関連する動作の変更があります。

#### SQL Central におけるパスワード情報の表示の変更 (データベースのアップグレードまたは再構築が必要)

ユーザに ACCESS USER PASSWORD システム権限がない場合、データベースサーバでは、次の場合に、ユーザにパスワード指定を求めるプロンプトが表示されます。

- コピー/ペーストまたはドラッグアンドドロップを使用して LDAP サーバを複製する場合
- プロパティシートで LDAP サーバの定義が変更されたがデータベースに保存されていないときや、アクセスパスワードが変更されていないときに、LDAP サーバへの接続をテストする場合
- プロパティシートでリモートサーバの定義が変更されたがデータベースに保存されていないときや、リモートサーバに現在のユーザの外部ログインがあるときに、リモートサーバへの接続をテストする場合

以前は、既存の外部ログインまたは LDAP サーバをコピーアンドペーストすることによって、新しい外部ログインまたは LDAP サーバを SQL Central に作成するとき、パスワードもコピーされていましたが、パスワードがコピーされるのは、ACCESS USER PASSWORD システム権限がある場合のみになりました。この権限がない場合は、新しい外部ログインまたは LDAP サーバのパスワードを入力するようにプロンプトが表示されます。

以前は、既存のユーザをコピーアンドペーストすることによって、新しいユーザを SQL Central に作成するとき、パスワードもコピーされていましたが、新しいユーザのパスワードを指定するようにプロンプトが表示されるようになりました (ACCESS USER PASSWORD システム権限がある場合でも)。

#### クライアントコンピュータにパスワードを格納するための新しいパスワード難読化オプション

EncryptedPassword 接続パラメータの名称が EncodedPassword (省略形 ENP) に変更され、ODBC データソースに格納されているパスワードの暗号化および復号化に使用可能なオプションにより適した名称になりました。新し

いオプションには、特定のコンピュータおよびユーザ/コンピュータの組合せに対するパスワードの暗号化/復号化が含まれます。Windows では、Microsoft Cryptography API を使用して、データソースが作成されたコンピュータまたはユーザ/コンピュータによる使用のみを目的としたパスワードが暗号化されるようになりました。EncryptedPassword 接続パラメータは、下位互換性を確保するために引き続きサポートされます。データソースに、EncryptedPassword パスワードの代わりに省略形 ENP が含まれるようになります。

Windows で、*ODBC データソースアドミニストレータの SQL Anywhere の ODBC 設定* ウィンドウに、パスワードの暗号化をサポートするための新しいオプションが用意されました。

データソースユーティリティ (dbdsn) に、特定のコンピュータおよびユーザ/コンピュータの組合せに対するパスワードの暗号化をサポートするための新しいオプションが用意されました。

#### 設定ファイルの新しい難読化オプション

ファイル非表示ユーティリティ (dbfhide) で、Linux、MAC OS X、その他の UNIX システム上での特定のコンピュータおよびユーザ/コンピュータの組合せに対するファイルの暗号化のための新しいオプションがサポートされます。

**RSA キーペアを使用したメッセージの暗号化、署名、検証に対するサポートの追加 (データベースのアップグレードが必要)**

RSA キーペアを使用した、データの小さなブロックの暗号化、メッセージへの署名、メッセージの検証に対するサポートが追加されました。

この機能をサポートするために、いくつかの変更が加えられています。

- RSA キーペアを作成するための新しい sp\_generate\_key\_pair システムプロシージャ (データベースのアップグレードまたは新しいデータベースの作成が必要)。
- メッセージにデジタル署名するための新しい SECURE\_SIGN\_MESSAGE 関数。
- デジタル署名されたメッセージを検証するための新しい SECURE\_VERIFY\_MESSAGE 関数。
- RSA 暗号化アルゴリズムと OAEP および PKCS1 埋込タイプをサポートするための ENCRYPT 関数の強化。
- RSA 暗号化アルゴリズムと OAEP および PKCS1 埋込タイプをサポートするための DECRYPT 関数の強化。

#### データベース独立性に対するサポートの追加

データベースサーバのデータベース独立性を有効にするためのサポートが追加されました。データベース独立性をオンにすると、各データベースがデータベースサーバで動作している唯一のデータベースであるかのように動作します。

次のように変更されています。

- 新しい -edi データベースサーバオプションにより、データベースのアイソレーションを有効化または無効化できます。
- SERVER\_SECURITY 機能セットに DATABASE\_ISOLATION 機能が追加されました。この機能を使用することで、現在の接続のデータベースアイソレーションを一時的に無効化できます。

#### プロシージャ実行時のユーザコンテキストを決定するための新しい特別値

データベースサーバで、プロシージャ実行時に、ログインユーザ、呼び出し側のユーザ、有効なユーザ、プロシージャ所有者を決定できる 4 つの特別値がサポートされるようになりました。これらの特別値は、ネストされたプロシージャ、SQL SECURITY INVOKER および SQL SECURITY DEFINER で定義されているプロシージャがある場合と、実行しているユーザがすぐに分からない場合に特に役立ちます。

- SESSION USER 特別値。
- INVOKING USER 特別値。
- EXECUTING USER 特別値。

- PROCEDURE OWNER 特別値。

この機能の一部として、次の予約語がソフトウェアに追加されました。

- 呼び出し
- 実行
- session\_user
- invoking\_user
- executing\_user
- procedure\_owner

呼び出し側セキュリティモデルを使用して初期化またはアップグレードされたデータベースのプロシージャパーミッションの変更

従来の定義者セキュリティモデルを使用してデータベースを初期化またはアップグレードする場合、現在のログインユーザにプロシージャを実行するパーミッションがあることを確認するために、次のプロシージャが続きます。より安全な呼び出し側モデルを使用して初期化またはアップグレードしたデータベースの場合、次のストアプロシージャで現在の有効なユーザにプロシージャを実行するための該当するパーミッションがあることが確認されるようになりました。

対象のプロシージャには、次のものがあります。

- sa\_server\_option
- sa\_db\_option
- sp\_create\_secure\_feature\_key
- sp\_alter\_secure\_feature\_key
- sp\_drop\_secure\_feature\_key

セキュア接続に使用できるようになったオペレーティングシステムの証明書ストア

TLS 接続 (HTTPS Web プロシージャ、LDAP ユーザ認証、セキュア TLS 接続を含む) で、オペレーティングシステム証明書ストアを使用して、セキュア接続用に信頼できる証明書を取得できるようになりました。サーバ証明書がストア内の任意の証明書によって署名され、接続が成功します。この機能のサポートは、SQL Central の SQL Anywhere プラグインに含まれています。

trusted\_certificates プロトコルオプションの使用時に、証明書ファイル名として \* を指定できるようになりました。この値により、クライアントソフトウェアがオペレーティングシステム証明書ストアで該当するルート証明書を検索します。

CREATE PROCEDURE または CREATE FUNCTION 文の CERTIFICATE 句の file オプションを使用するとき、証明書ファイル名として \* を指定できるようになりました。この値により、データベースサーバがオペレーティングシステム証明書ストアで該当するルート証明書を検索します。

TLS または HTTPS 接続を行うときに、certificate\_name プロトコル、certificate\_company プロトコル、certificate\_unit プロトコルのいずれも設定されていない場合は、データベースサーバのホスト名がデータベースサーバ証明書のホスト名と照合されます。このホスト名チェックが実行されないようにするために、次のいずれかを選択します。

#### TLS 接続を行うとき

skip\_certificate\_name\_check プロトコルオプションを ON に設定します。

**HTTP** サーバへの安全な要求を行う Web クライアントプロシージャを作成または変更するとき

skip\_certificate\_name\_check プロトコルオプションを ON に設定します。

**SMTP** で電子メールセッションを開始するとき (データベースのアップグレードが必要) xp\_startsmtp システムプロシージャの skip\_certificate\_name\_check パラメータを 1 に設定します。

データベースの暗号化でのセキュリティの強化 (データベースのアップグレードまたは再構築が必要)

初期化ベクトルの改善により、データベースの暗号化でのセキュリティが強化されました。

UNIX および Linux システムに対する PAM ユーザ認証のサポート (データベースのアップグレードまたは再構築が必要)

サポート対象のすべての UNIX および Linux プラットフォームで、PAM ユーザ認証 (PAMUA) が使用可能です。

LDAP ユーザ認証

LDAP ユーザ認証 (LDAPUA) ライブラリが再実装され、ファイル名が新しくなりました。2 つのライブラリがあり、1 つは FIPS 認定の暗号化用で、もう 1 つは FIPS 認定でない暗号化用です。

ARM プロセッサで動作している Linux に対するサポートの追加

Linux ベースの ARM プロセッサ (たとえば、Raspberry Pi など) で、SQL Anywhere がサポートされるようになりました。

## 1.3.9 マニュアルの強化

SQL Anywhere 17.0 は、マニュアルも強化されています。

バックアップおよびリカバリマニュアルの改善

データベースのバックアップとリストアについてのトピックが再構成され、移動しやすくなり、使用可能なバックアップタイプに関する情報が充実し、用語が統一されました。さらに、タスクがバックアップツールごとに再構成されたため、使用しているツールの機能に関する情報が見つかりやすくなりました。

## 1.3.10 その他の変更点と強化点

SQL Anywhere 17.0 一般的な強化がさまざまに施されています。

### その他の新機能

データベーススコープ変数に対するサポートの追加 (データベースのアップグレードが必要)

以前は、接続スコープ変数しか作成できませんでしたが、データベーススコープ変数を作成できるようになりました。データベーススコープ変数は、接続間で値を共有し、データベースの再起動後に宣言時の接続初期値を保持する優れた方法です。使用目的は、変更頻度が低い小さな共有値を格納することです。変更頻度が高い値を格納することは、アプリケーションのパフォーマンスに影響を与える可能性があるため、おすすめしません。

データベーススコープ変数をサポートするために、ソフトウェアにいくつかの新しい機能が追加されています。

**CREATE VARIABLE 文の変更**

この構文に、データベーススコープ変数を作成するためのオプションの DATABASE キーワードが含まれるようになりました。

**DROP VARIABLE 文の変更**

データベーススコープ変数を削除するための新しい構文が追加されました。



## 新しいシステム権限

4 つの新しいシステム権限が追加されました。

- CREATE DATABASE VARIABLE システム権限
- MANAGE ANY DATABASE VARIABLE システム権限
- SELECT PUBLIC DATABASE VARIABLE システム権限
- UPDATE PUBLIC DATABASE VARIABLE システム権限

## 新しい YSDATABASEVARIABLE システムビュー

SYSDATABASEVARIABLE システムビューと ISYSDATABASEVARIABLE システムテーブルに、すべてのデータベーススコープ変数に関する情報が含まれます。

## VAREXISTS 関数の変更

VAREXISTS 関数で、データベーススコープ変数の所有者を指定するための新しいパラメータがサポートされます。

## DROP VARIABLE 文、SET 文、UPDATE 文の変更

DROP VARIABLE 文、SET 文、UPDATE 文で、データベーススコープ変数の所有者の指定がサポートされるようになりました。

## 変数の文書の強化

製品における変数サポートの文書が平易になり、改善されました。

## リモートサーバの強化 (データベースのアップグレードが必要)

- ALTER SERVER 文の新しい CONNECTION CLOSE 句を使用したリモートサーバの読込専用への変更に対するサポート。  
CREATE SERVER 文も、以前のバージョンのソフトウェアとの下位互換性も維持しながら、ALTER SERVER 文の READ ONLY 句と同じ構文をサポートするように拡張されました。
- リモートサーバ作成時のデフォルトログインの指定に対するサポート (新しい DEFAULT LOGIN 句)。ユーザに外部ログインがない場合は、代わりに接続にデフォルトログイン情報が使用されます。
- CREATE SERVER 文、ALTER SERVER 文、DROP SERVER 文での、オプションの REMOTE キーワードに対するサポート (たとえば、CREATE REMOTE SERVER など)。

## データベースサーバでのオートコミットに対するサポートの追加

データベースサーバでオートコミットがサポートされるようになりました。旧バージョンでは、オートコミットは有効になっている場合にクライアントソフトウェアによって実行されていましたが、サーバで要求が自動的にコミットされるようにクライアントが要求できるようになりました。この結果、パフォーマンスが向上します。

## テーブル式の行列変換および列行変換に対するサポートの追加

クエリの FROM 句で PIVOT と UNPIVOT の 2 つの句を使用してテーブルデータを行列変換および列行変換して、行列変換された派生テーブルまたは列行変換された派生テーブルを作成できるようになりました。行列変換では、カラムデータがローに回転され、ビジネスニーズにとって有意義な方法でデータが集計されます。列行変換は同様の操作ですが、ローデータがカラムに回転されることはありません。列行変換は、他のデータとジョインする必要がある類似データのいくつかの列など、正規化されていないデータを正規化する際に便利です。

## ディレクトリアクセスサーバの作成と変更の強化 (データベースのアップグレードが必要)

以前は、ディレクトリアクセスサーバを作成または変更してディレクトリアクセスを読込専用にするかどうかを制御するときは、USING 文字列で READ ONLY 句を指定していましたが、USING 句とは関係なく使用するために READ ONLY 句を指定するようになりました。以前の READ ONLY 構文は引き続きサポートされますが、廃止予定であり、今後のリリースでサポートされなくなる可能性があります。さらに、CREATE SERVER 文または ALTER SERVER 文で新しい ALLOW

{ SPECIFIC | ALL } 句を使用してディレクトリアクセスサーバにアクセスするために外部ログイン (externlogin) が必要であるかを制御できるようになりました。

#### 同名の複数の HTTP ヘッダの読み込みおよび設定に対するサポートの追加

読み込みおよび設定対象の HTTP ヘッダのインスタンスを追加できるようになりました。この機能をサポートするために、次のような変更が加えられています。

- HTTP\_HEADER 関数で、instance パラメータがサポートされるようになりました。
- HTTP\_RESPONSE\_HEADER 関数で、instance パラメータがサポートされるようになりました。
- sa\_set\_http\_header システムプロシージャで、instance パラメータがサポートされるようになりました (データベースのアップグレードが必要)。
- 要求に同名の HTTP ヘッダが複数含まれている場合、sa\_http\_header\_info システムプロシージャが同名のローを複数返すことがあります (データベースのアップグレードが必要)。

HTTP\_HEADER 関数と HTTP\_RESPONSE\_HEADER 関数の新しい機能にアクセスするには、SQL Anywhere パージョン 17 のデータベースサーバを使用してください。sa\_set\_http\_header システムプロシージャと sa\_http\_header\_info システムプロシージャの新しい機能にアクセスするには、データベースのアップグレードまたは再構築が必要です。

#### HTTP OPTIONS メソッドに対するサポートの追加

HTTP Web サービス、ストアプロシージャ、関数で、OPTIONS メソッドがサポートされるようになりました。何より、この機能を使用すると、Web サービスでサポートされるクロスオリジンリソース共有 (CORS) が可能になります。

#### 安定した状態に対するキャッシュウォーミングのサポートの追加 (データベースのアップグレードが必要)

パフォーマンスを向上させるために、安定期にキャッシュコンテンツを記録し、必要に応じてこの情報をリストアすることができます。

この機能をサポートするために、次のような変更が加えられています。

##### 新しい SQL 文

- ALTER DATABASE SAVE CACHE 文
- ALTER DATABASE RESTORE CACHE 文
- ALTER DATABASE DROP CACHE 文

##### 新しいシステムプロシージャ

- sp\_db\_cache\_contents システムプロシージャ。
- sp\_read\_db システムプロシージャ。

#### CONTAINS 探索条件に対するフラット化できないクエリ式の許可

ビューや派生テーブルなどのフラット化できないクエリ式に対して、クエリの WHERE 句で CONTAINS 探索条件が許可されるようになりました。

#### Native Component Supportability (NCS) ライブラリに対するサポートの追加

Microsoft Windows および 64ビット Linux で、データベースサーバが NCS (Native Component Supportability) ライブラリを使用して、ステータスイベントを SAP Solution Manager に送信できるようになりました。

#### 変数の初期値の強化

以前は、変数の作成または宣言時に、定数式、特別値、定数式の組込関数を使用してのみ、変数を初期化することができましたが、他の変数、ユーザ定義の関数、サブクエリなど、任意の式を指定できるようになりました。

#### ORDER が指定されていない場合のトリガの実行順序の変更

カラムで複数のトリガを定義する場合、複数のトリガで同じイベントタイプが指定され、重複するイベントタイプが呼び出された場合、データベースサーバでトリガが実行される一意の順序をトリガごとに指定する必要があります。たとえば、イベントタイプが UPDATE および INSERT のトリガと、イベントタイプが UPDATE OF のトリガは、イベントタイプが重複すると見なされるため、トリガごとに一意の実行順序を指定する必要があります。

このような場合に順序を指定しなかった場合、データベースサーバでは、必ずしも予期できず、リリース間で変更される可能性がある実装固有の方法でトリガが実行されます。今回のリリースでは、実装固有の順序の変更が導入されています。

このような変更が導入される可能性があるため、イベントタイプが重複するトリガの順序を指定してください。このようにすることにより、今後のリリースで、イベントタイプが重複するトリガの予測可能な実行順序が保証されます。

イベントタイプが UPDATE の 2 つのトリガと同様に、イベントタイプが UPDATE および UPDATE OF のトリガでは一意の順序指定が必要です。

#### UNIX システムで POSIX セマフォを使用

以前は、UNIX システムでは SYSV セマフォが使用されており、データベースサーバへの接続が制限されていましたが、Mac OS X と Solaris を除くすべての UNIX システムで、POSIX セマフォが使用されるようになりました。

Mac OS X では POSIX セマフォがサポートされておらず、Solaris では POSIX セマフォと同様に機能する独自のセマフォが使用されています。

#### UNIX 共有メモリのセキュリティの強化

セキュリティを強化するために、UNIX がディスク上のファイルをメモリにマップするとき、ファイルが作成直後にファイルシステムから削除されるようになりました。以前は、これらのファイルには SATMP の場所にアクセスするパーミッションがあればアクセスできました。

#### CPU トポロジの検出の改善

データベースサーバが、特に仮想マシンで動作しているときに、Linux 上の CPU トポロジをより正確に検出できるようになりました。

サーバライセンス取得ユーティリティ (dblic) でコアベースライセンスがサポートされるようにサーバライセンス取得ユーティリティ (dblic) が、パーシートライセンスに加えて、コアベースライセンスをサポートするようになりました。プロセスベースのライセンスは廃止予定です。

Linux 用サービスユーティリティ (dbsvc) での systemd サービスのサポート Linux 用サービスユーティリティ (dbsvc) で systemd サービスインタフェースがサポートされるようになりました。この新機能のサポートにおいて、i、-li、-lt オプションがユーティリティに追加されました。このユーティリティでは、サービス名、サービス引数、SQL Anywhere のインストールパスにおいて、スペースおよびその他の特殊文字がサポートされるようになりました。

## データベース、接続、データベースサーバのオプション、パラメータ、プロパティ

データベース、接続、データベースサーバのオプション、パラメータ、プロパティが、次のように強化されています。

#### -ufd データベースサーバオプションの改善

データベースアサーション障害を処理するより柔軟な手段を提供するために、-ufd dbsrv17 データベースサーバオプションが 2 つの新しいオプションを使用できるように拡張されています。

- restart\_escalate
- restart\_abort

#### 新しい auto\_commit オプション

auto\_commit オプションを ON に設定すると、データベースサーバはすべての要求の後に自動的にコミットします。

以前にオートコミットがサポートされていなかった Embedded SQL などの API に対して、これらのアプリケーションが auto\_commit オプションを使用して、実行ごとにデータベースサーバを自動的にコミットさせることができるようになりました。

#### 新しいデータベースオプション

- max\_connections データベースオプション
- connection\_type データベースオプション
- time\_zone データベースオプション

#### 新しいデータベースサーバオプション

- -uq データベースサーバオプション

#### 新しい接続パラメータ

- CommBufferSize (CBSIZE) 接続パラメータの強化

#### 新しいデータベースプロパティ

- TimeWithoutClientConnection データベースプロパティ
- BackupInProgress データベースプロパティ
- MaxConnections データベースプロパティ
- Timezone データベースプロパティ
- CurrentTimezoneOffset データベースプロパティ

#### 新しいデータベースサーバプロパティ

- HasSecureFeatureKey データベースサーバプロパティ
- HasSecuredFeature データベースサーバプロパティ

#### 新しい接続プロパティ

- HasSecuredFeature 接続プロパティ
- connection\_type 接続プロパティ
- max\_connections 接続プロパティ
- time\_zone 接続プロパティ

#### 新しいネットワークプロトコルオプション

- LogRename (LRENAME) プロトコルオプション

## 1.3.11 動作変更、廃止機能、およびサポートを終了した機能

SQL Anywhere 17.0 は動作に変更が施されています。一部には廃止された機能やサポートを終了した機能もあります。

### 動作の変更

**SAP HANA** テーブルでインデックスがマイグレート対象外に SAP HANA テーブルを SQL Anywhere に移行する場合、インデックスはテーブルとともに移行されないため、移行後に手動で作成する必要があります。

システムビューでのパスワードハッシュの非表示化

以前は、システムビューでユーザがパスワードハッシュを表示することができましたが、パスワードハッシュへのアクセスが、次の方法を使用して制御されるようになりました。

- 一部のビューへのアクセスに必要な新しい ACCESS USER PASSWORD システム権限
- 制限付の新しいシステムビュー
- パスワード関連のカラムでの値の表示方法の変更

空間 API: ST\_TransformGeom メソッドで、アップグレードされた PROJ.4 ライブラリをサポート

アップグレードされたライブラリを使用すると、特定の射影の結果に以前の PROJ.4 ライブラリを使用して計算した射影の結果の値よりも正確な新しい値が含まれるようになります。

以前は、PROJ.4 ライブラリで変換パラメータとして no\_defs が受け入れられていましたが、パラメータを +no\_defs とし指定しなければならなくなりました。すべてのコードまたは SRS 定義が更新されるようになります。

アップデートされた PROJ.4 ライブラリでは、垂直データがサポートされます。+to\_meter を使用した射影では、指定した位取り係数を使用して Z 座標がメートルに変換されます。新しい +vto\_meter 設定を使用することによって、Z 座標に別の換算係数を指定することができます。垂直データが変換されないようにするには、+vto\_meter=1.0 と設定します。

**UPPER、UCASE、LOWER、LCASE 関数が LONG (N)VARCHAR を返す (アップグレードまたは再構築が必要)**

以前は、UCA 照合を使用するデータベース内の NCHAR データまたは CHAR データで UPPER、UCASE、LOWER、LCASE 関数が呼び出されたとき、データは入力と同じ長さを持つものとして記述されていましたが、これらの関数を呼び出すと、UCA 照合を使用するデータベース内で、NCHAR データに対しては LONG NVARCHAR が返され、CHAR データには LONG VARCHAR が返されるようになりました。

これらの関数を呼び出すマテリアライズドビューを再作成および再コンパイルし、これらの関数を呼び出すすべてのビューとプロシージャを再コンパイルしてください。

データベースサーバをシャットダウンするかどうかを決定するときに、セッション ID が関連付けられていない HTTP 接続と HTTPS 接続が無視される

以前は、データベースサーバをシャットダウンするための条件が満たされているかどうかをどうかを判定するときに、すべての HTTP 接続と HTTPS 接続が考慮されていましたが、データベースサーバをシャットダウンするかどうかを決定するときに、セッション ID が関連付けられている HTTP 接続と HTTPS 接続のみが考慮されるようになりました。セッション ID のない HTTP 接続と HTTPS 接続は無視されます。この動作の変更には、次のような影響があります。

- データベースサーバのメッセージウィンドウでシャットダウンをクリックすると、セッション ID が関連付けられていない HTTP 接続および HTTPS 接続のみがある場合に、警告ウィンドウが表示されなくなります。
- セッション ID が関連付けられていない HTTP 接続と HTTPS 接続のみがある場合は、dbstop ユーティリティを使用してデータベースサーバを条件付きで停止すると、うまくいくようになりました。
- セッション ID が関連付けられていない HTTP 接続と HTTPS 接続のみがある場合は、STOP SERVER 文を使用してデータベースサーバを条件付きで停止すると、うまくいくようになりました。

**GET\_IDENTITY 関数で権限が必要**

以前は、GET\_IDENTITY 関数を使用して ID 値を返したり予約したりするためにテーブル上での権限は不要でしたが、この関数を使用するためにテーブル上で INSERT 権限が必要になりました。

**UtilCmdsPermitted 接続プロパティの変更** UtilCmdsPermitted プロパティが、現在の接続を除くすべての接続に対して空の文字列を返すようになりました。

**アンロードユーティリティ (dbunload) の変更**

以前は、アンロードユーティリティを使用して再ロード操作 (ac, -an, -ar オプション) でアンロードを実行するには、ELECT ANY TABLE 権限と SERVER OPERATOR 権限が必要でしたが、これらのシステム権限に加えて、このリリースで導入された新しい ACCESS USER PASSWORD システム権限が必要になりました。

以前は、再ロードなしで (すなわち、-ac、-an、-ar を指定しないで) アンロードを実行するときは、デフォルトでパスワードがアンロードされていましたが、パスワードをアンロードするには -up オプションを指定する必要があり、そのために ACCESS USER PASSWORD システム権限が必要になりました。

#### カスタマイズされた max\_connections データベースオプションの変更

バージョン 17 では、新しい max\_connections データベースオプションが導入されました。ユーザ定義の max\_connections データベースオプションがすでにある場合は、ユーザ定義の動作が引き続き使用されます。新しい max\_connections データベースオプションの動作を使用するには、データベースをバージョン 17.0 にアップグレードする必要があります。アップグレード中に max\_connections データベースオプションを設定すると、アップグレードが失敗します。現在のデータベースサーバ接続の制限は維持されます。

#### システムビューでのパスワード情報の保護に関するカタログ変更 - アップグレードが必要

パスワードハッシュなどの機密情報を非表示にするための改善に対応するために、次のビューが新しくなったか、変更されました。

ビュー名	変更点
SYSSYNC	ビューは変更されていませんが、このシステムビューから選択するための SELECT ANY TABLE システム権限とともに、新しい ACCESS USER PASSWORD システム権限も必要になりました。
SYSSYNC2	以前は、ユーザに SELECT ANY TABLE システム権限がある場合、このオプションおよび server_connect カラムにはパスワードハッシュなどの機密データが表示されていましたが、値が格納されている場合はこれらのカラムに 3 つのアスタリスク (***) が表示され、値が格納されていない場合は NULL が表示されるようになりました。さらに、このビューには SYSSYNC システムビューにある server_protocol カラムが表示されるようになりました。
SYSSYNCS	以前は、このビューは ISYSSYNC システムテーブルと ISYSPUBLICATION システムテーブルから選択していましたが、パスワードハッシュなどの機密データが表示されない SYSSYNC2 システムビューと SYSPUBLICATION システムビューから選択するようになりました。
SYSSYNCSUSERS	以前は、このビューは ISYSSYNC システムテーブルから選択していましたが、パスワードハッシュなどの機密データが表示されない SYSSYNC2 システムビューから選択するようになりました。
SYSSYNCPUBLICATIONDEFAULTS	以前は、このビューは ISYSSYNC システムテーブルと ISYSPUBLICATION システムテーブルから選択していましたが、パスワードハッシュなどの機密データが表示されない SYSSYNC2 システムビューと SYSPUBLICATION システムビューから選択するようになりました。

ビュー名	変更点
SYSSYNCSUBSCRIPTIONS	以前は、このビューは ISYSSYNC システムテーブル、ISYSUSER システムテーブル、ISYSPUBLICATION システムテーブルから選択していましたが、パスワードハッシュなどの機密データが表示されない SYSSYNC2 システムビュー、SYSSYNCSUSERS システムビュー、SYSPUBLICATION システムビューから選択するようになりました。
SYSUSERPERM	パスワードカラムは以前は BINARY(128) ででしたが CHAR(3) になりました。また、パスワードカラムには以前はパスワードハッシュが表示されていましたが、パスワードが格納されている場合を示すためにアスタリスクが表示され、パスワードが格納されていない場合は NULL が表示されるようになりました。パスワードハッシュにアクセスするには、SYSUSERPASSWORD システムビューを使用します。
SYSUSERAUTH	このビューは SYSUSERPERM システムビューから選択し、システムビューの変更によって変更されました (パスワードハッシュが表示されなくなります)。
SYSUSER	パスワードカラムと dual_password カラムは以前は BINARY(128) ででしたが CHAR(3) になりました。また、これらのカラムには以前はパスワードハッシュが表示されていましたが、パスワードが格納されている場合を示すためにアスタリスクが表示され、パスワードが格納されていない場合は NULL が表示されるようになりました。パスワードハッシュにアクセスするには、SYSUSERPASSWORD システムビューを使用します。
SYSUSERPASSWORD	新規。パスワードハッシュが SYSUSER システムビューに表示されなくなったため、このビューを使用して、格納されているログイン ID のパスワードハッシュにアクセスします。
SYSEXTERNLOGIN	remote_password カラムは以前は VARBINARY(1024) ででしたが CHAR(3) になりました。また、このカラムには以前はパスワードハッシュが表示されていましたが、パスワードが格納されている場合を示すためにアスタリスクが表示され、パスワードが格納されていない場合は NULL が表示されるようになりました。
SYSEXTERNLOGINPASSWORD	新規。パスワードハッシュが SYSEXTERNLOGIN システムビューに表示されなくなったため、このビューを使用して、格納されているリモートログインのパスワードハッシュにアクセスします。
SYSLDAPSERVER	ldsrv_access_dn_pwd カラムは以前は VARBINARY(1024) ででしたが CHAR(3) になりました。また、このカラムには以前はパスワードハッシュが表示されていましたが、パスワードが格納されている場合を示すためにアスタリスクが表示され、パスワードが格納されていない場合は NULL が表示されるようになりました。

ビュー名	変更点
SYSLDAPSERVERPASSWORD	新規。パスワードハッシュが SYSLDAPSERVER システムビューに表示されなくなったため、このビューを使用して、格納されている LDAP サーバのパスワードハッシュにアクセスします。
SYSSYNCPROFILE	このシステムビューに対してクエリを実行するために、SELECT ANY TABLE システム権限と ACCESS USER PASSWORD システム権限が必要になりました
SYSSYNCPROFILE2	新規。このビューは、機密パスワード情報を含めることができる profile_defn カラムがないことを除いて、SYSSYNCPROFILE システムビューと同じです。このシステムビューにアクセスするために必要な権限はありません。

## 廃止された機能

廃止された機能は、今後のバージョンで削除される予定です。アプリケーションを変更して、廃止された機能の代わりに推奨されている代替機能をご使用ください。

### 2048 (2K) バイトのデータベースページサイズのサポートの廃止

データベースページサイズの 2048 バイトの値は廃止予定であり、推奨されなくなりました。2048 バイトのページサイズは引き続きサポートされますが、今後のリリースでサポートされなくなる可能性があります。

**sa\_make\_object** システムプロシージャが廃止予定に

以前は、スクリプトで sa\_make\_object システムプロシージャを使用して、データベース内のオブジェクトを作成したり変更したりしていました。この方法は、代わりに OR REPLACE 句が使用される CREATE 文の使用を優先して、廃止予定になりました。CREATE OR REPLACE を使用したほうが効率的であり、すでに存在するオブジェクトを作成しようとするときの動作が正しくなります。

## 削除された機能

### Windows Mobile での SQL Anywhere サーバのサポートの終了

Windows Mobile で SQL Anywhere サーバがサポートされなくなりました。ただし、Ultra Light は Windows Mobile で引き続きサポートされ、正常に機能し続けます。

### TLS 接続および HTTPS 接続での RC4-MD5 暗号化のサポート終了

TLS 接続および HTTPS 接続で、RC4-MD5 暗号化がサポートされなくなりました。

サポート終了:**CONNECTION CLOSE** 句、**ALTER SERVER** 文

ALTER SERVER の CONNECTION CLOSE 句がサポートされなくなりました。代わりに DROP REMOTE CONNECTION 文を使用してください。

**-sm** データベースオプションのサポート終了

旧バージョンで廃止予定だった -sm データベースオプションがサポートされなくなりました。



ON を除くすべての -xp データベースオプション値のサポート終了

旧バージョンで廃止予定だった -xp データベースオプションの次の値がサポートされなくなりました。

- partner-conn
- auth-str
- arbiter-conn
- mode
- autofailover
- pagetimeout
- preferred

## 1.4 Mobile Link バージョン 17.0 の新機能

Mobile Link バージョン 17.0 のいくつかの新機能、変更、廃止、削除された機能についてご紹介します。

廃止された機能は、今後のバージョンで削除される予定です。アプリケーションを変更して、廃止された機能の代わりに推奨されている代替機能をご使用ください。

### i 注記

マニュアル完成後に追加された機能情報を含む、ソフトウェアに関する追加情報については、`readme.txt` ファイルを参照してください。

## Mobile Link サーバ

### Native Component Supportability (NCS) ライブラリに対するサポートの追加

Mobile Link サーバで、`ncs.conf` ファイルを使用して SAP Diagnostics Agent に接続できるようになりました。

### SAP パスポートのサポート

Mobile Link サーバで、トレースのために SAP パスポートを使用できるようになりました。を参照してください。

### SAP IQ のサポート

Mobile Link サーバにより、AP IQ 16.0 以降のサーバ上で動作する統合データベースがサポートされるようになりました。

### i 注記

この強化は、バージョン 16 のサポートパッケージで最初にリリースされました。

### Oracle 12.1 統合データベースのサポート

Mobile Link サーバにより、Oracle 12.1 統合データベースがサポートされるようになりました。SQL Anywhere17 で Oracle 12.1 の機能を利用するには、SQL Anywhere - Oracle ODBC ドライバがビルド番号 1733 以降である必要があります。Oracle 12.1 のインストールイメージから Oracle OCI ライブラリをインストールする必要があります。

SQL Anywhere 17 で Oracle 12.1 の機能を利用するには、SQL Anywhere - Oracle ODBC ドライバがビルド番号 2087 以降である必要があります。Oracle 12.1 のインストールイメージから Oracle OCI ライブラリをインストールする必要があります。

#### SAP ASE 16.0 統合データベースのサポート

Mobile Link サーバにより、SAP ASE 16.0 統合データベースがサポートされるようになりました。

#### Microsoft SQL Server 2014 統合データベースのサポート

Mobile Link サーバにより、Microsoft SQL Server 2014 統合データベースがサポートされるようになりました。

#### MySQL 5.6.20 統合データベースのサポート

Mobile Link サーバにより、MySQL 5.6.20 統合データベースがサポートされるようになりました。

#### IBM DB2 10.5 統合データベースのサポート

Mobile Link サーバにより、IBM DB2 10.5 統合データベースがサポートされるようになりました。

#### Mobile Link による LDAP 認証のサポートの変更

Mobile Link による LDAP ユーザ認証のサポート方法が、次のように変更されています。

- ml\_add\_user\_auth\_policy システムプロシージャの ldap\_failover\_to\_std パラメータの設定が 0、1、または 2 に拡張されました。
- -zup Mobile Link サーバオプションを使用して、LDAP サーバに対してユーザ認証のデフォルトの認証ポリシー名を指定することができます。
- ldap\_failover\_to\_std パラメータを値 1 または 2 で設定すると、Mobile Link ユーザパスワードがハッシュのみされ、統合データベースの ml\_user テーブルに格納されます。このパラメータを 0 に設定すると、パスワードは保存されません。

## Mobile Link クライアント

### 再起動可能なダウンロードの新しいオプション

失敗したダウンロードを再起動できるようにするために必要な情報を保存するには、-kpd dbmsync オプションまたは KeepPartialDownload 同期プロファイルオプションを使用します。以前のリリースでは、ダウンロードが失敗すると dbmsync は常にダウンロードの再起動に必要な情報の保存を試行していましたが、-kpd dbmsync オプションまたは KeepPartialDownload 同期プロファイルオプションを指定した場合にのみ、ダウンロードが失敗すると dbmsync がダウンロードの再起動に必要な情報の保存を試行するようになりました。

dbmsync がデータがまったく受信されていない場合にダウンロードを再起動する機能を提供 dbmsync ユーティリティを使用して、ダウンロードがまったく受信されていない場合でも失敗したダウンロードを再起動できるようになりました。以前は、少なくとも 1 バイトが受信された場合にのみダウンロードを再起動できました。

#### i 注記

この強化は、バージョン 16 のサポートパッケージで最初にリリースされました。

### dbmsync オフラインランザクションログの取得の拡張

dbmsync ユーティリティを使用して、SQL Anywhere データベースサーバに直接アクセスするのではなく、SQL Anywhere データベースサーバからオフラインランザクションログを取得できるようになりました。オフラインランザクションログが必要だが、特定のオフラインランザクションログディレクトリを直接開くことができない場合や、オフラインラン

ザクシヨソログファイルが含まれていない場合に、dbmsync はデータベースサーバを通じてオフライントランザクシヨソログを取得します。次の制限事項があります。

- 同期データベースに接続するために dbmsync で使用されるユーザ ID には READ FILE 権限と WRITE FILE 権限が必要で、すべてのオフライントランザクシヨソログファイルがオンライントランザクシヨソログディレクトリにある必要があります。
- SQL Anywhere データベースサーバには、この機能をサポートするビルド番号 1961 以降のサポートパッケージが必要です。

この機能を使用すると、データベースサーバがページを取得するために処理が増えるため、パフォーマンスが若干低下します。パフォーマンスが重要な場合は、dbmsync OfflineDirectory 拡張オプションの使用が配備に最適であることがあります。

### i 注記

この強化は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### 認証の改善

TLS または HTTPS 接続を行うときに、certificate\_name プロトコル、certificate\_company プロトコル、certificate\_unit プロトコルのいずれも設定されていない場合は、データベースサーバのホスト名がデータベースサーバ証明書のホスト名と照合されます。このホスト名がチェックされないようにするには、TLS または HTTPS への接続時に新しい skip\_certificate\_name\_check プロトコルオプションを ON に設定します。を参照してください。

#### 新しい skip\_certificate\_name\_check プロトコルオプション

クライアントライブラリがデータベースサーバの証明書ホスト名に対するサーバホスト名のチェックを省略するかどうかを制御するには、skip\_certificate\_name\_check プロトコルオプションを使用します。を参照してください。

#### Mobile Link クライアントの新しい allow\_expired\_certs ネットワークプロトコルオプション

allow\_expired\_certs プロトコルオプションを使用して、期限切れになった、またはまだ有効でないサーバ証明書を受け入れて、同期を続行します。

## SQL Central の Mobile Link プラグイン

ウィンドウ設定のテストウィンドウが設定可能に

同期モデルのテスト、同期ロジックのより詳細なテストの有効化、運用環境に近い環境の作成を行うために使用するいくつかのオプションを設定するには、[ウィンドウ設定のテストウィンドウ](#)を使用します。

複数のユーザへのユーザ認証ポリシーの割当

同時に複数のユーザにユーザ認証ポリシーを割り当てることができるようになりました。

同期モデルが複製可能 Mobile Link プラグインの同期モデルメニューに、同期モデルを複製するための新しいオプションが追加されました。

### i 注記

この強化は、バージョン 16 のサポートパッケージで最初にリリースされました。

#### Mobile Link プラグインによる LDAP 認証のサポートの変更

Mobile Link プラグインによる LDAP ユーザ認証のサポート方法が、次のように変更されています。

- Mobile Link プラグインで、LDAP 認証のための標準 Mobile Link 認証スクリプトの呼出がサポートされるようになりました。新規ユーザ認証ポリシーウィザードの *LDAP サーバ* ページおよび認証ポリシーのプロパティウィンドウの一般タブで、呼び出さない、常に呼び出す、LDAP サーバが利用できない場合を指定することができます。
- Mobile Link サーバオプション `-zup` がサポートされます。`-zup` オプションを使用して、LDAP サーバに対してユーザ認証のデフォルトの認証ポリシー名を指定することができます。このオプションには、サーバのコマンドラインのプロパティウィンドウの詳細タブからアクセスできます。

#### Oracle サポートへの強化点

`dbmlsrv17 -dt` オプションにより、Oracle 統合データベースがサポートされるようになりました。

## Mobile Link 動作変更、廃止機能、およびサポートを終了した機能

### Mobile Link プラグインでの `-cmax` オプション、`-cmin` オプション、`-vk` オプションのサポート終了

サーバのコマンドラインのプロパティ行で、Mobile Link サーバオプション `cmax`、`-cmin`、`-vk` がサポートされなくなりました。これらのオプションのいずれかを指定して既存のコマンドラインをロードした場合、これらのオプションは削除されません。

### SAP HANA 統合データベースのサポート

Mobile Link の以前のリリースは、SAP HANA 統合データベースをサポートしていました。現在は、SQL Anywhere クライアントを SAP HANA 統合データベースと同期するには、SAP HANA リモートデータ同期を使用します。サーバ起動同期および監視サーバを除くすべての機能がサポートされています (HANA プラットフォームには独自の高可用性ツールが搭載されています)。

### Mobile Link サーバの API パッケージ名の変更

Mobile Link サーバの Java API のパッケージ名が変更されました。

```
com.sap.ml.script
```

Mobile Link サーバの .NET API のパッケージ名が変更されました。

```
Sap.MobiLink.Script
```

### `mlrsa_tls17.dll` および `mlrsa_tls_fips17.dll` の置換え

Mobile Link サーバで、`mlrsa_tls17.dll` や `mlrsa_tls_fips17.dll` が使用されなくなりました。代わりに `dbrsa17.dll` または `dbfips17.dll` が使用されます。

### Mobile Link サーバオプション `-cmax`、`-cmin`、`-cm`、`-vk` のサポート終了

Mobile Link サーバで、`cmax` オプション、`-cmin` オプション、`-vk` オプションがサポートされなくなりました。指定しても無視されます。

### Mobile Link および Ultra Light での Microsoft ActiveSync プロバイダのサポート終了

Mobile Link および Ultra Light で、Microsoft ActiveSync ユーティリティがサポートされなくなりました。

### `-oe` オプションの削除

`mlsrv17 -x` オプションの `-oe` オプションがサポートされなくなりました。

統合化 **Outbound Enabler** のサポート終了 Mobile Link の統合化 **Outbound Enabler** がサポートされなくなりました。`mlsrv17` で `-x oe` オプションは使用できなくなりました。代わりに標準のスタンドアロン **Relay Server Outbound Enabler** を使用してください。

## POP3 および MAPI のユーザ認証クラスの削除

POP3 および MAPI を介した Mobile Link ユーザ認証の次の便利なクラスが削除されました。

- `ianywhere.ml.authentication.IMAP`  
`ianywhere.ml.authentication.POP3`

これらのクラスは、SQL Central の Mobile Link プラグインを使用して生成された `authenticate_user` イベントスクリプトに出現していました。そのようにして生成された `authenticate_user` イベントスクリプトが Mobile Link モデルに含まれている場合、それをマニュアルで作成した `authenticate_user` イベントスクリプトで置き換える必要があります。生成された `authenticate_user` イベントスクリプトは、SQL Central で Mobile Link モデルを開くと、自動的に削除されます。

## 関連情報

[プラットフォーム別 SAP SQL Anywhere 16.0 コンポーネント](#)

[Mobile Link の推奨 ODBC ドライバ](#)

## 1.5 Ultra Light バージョン 17.0 の新機能

Ultra Light バージョン 17.0 の新機能、また、変更、廃止、削除された機能についてご紹介します。

廃止された機能は、今後のバージョンで削除される予定です。アプリケーションを変更して、廃止された機能の代わりに推奨されている代替機能をご使用ください。

### 全体的な変更

エンコードに PKCS #12 を使用した ID サポート

ulinit および ulload ユーティリティ向けの *identity-file* オプションで、PEM および PKCS12 ID をサポートします。

**observer** の同期ステータスの追加

UL\_SYNC\_STATE\_RECEIVING\_UPLOAD\_ACK ステータスが削除され、新規のステータスが複数追加されました。

### Ultra Light.NET

メモリ管理の改善のため、`ULIndexSchema.Close()` を追加

この方法を使用して、Ultra Light.NET アプリケーションの `ULIndexSchema` インスタンスを閉じます。

## Ultra Light J

### Android Studio IDE 向け CustDB サンプルプロジェクト

Android Studio は Android デバイス用の公式 IDE です。CustDB サンプルに Android Studio プロジェクトが追加されました。

### Java パッケージ名の変更

Java クラスのパッケージ名は、com.sap.ultralitejni17 となりました。CustDB のパッケージ名は、com.sap.custdb となりました。

## プラットフォームとデバイス

### Windows Phone 8.1 および Windows 8.1 ストアアプリのサポート

Ultra Light for WinRT API により、Windows 8.1 ストアアプリおよび Windows Phone 8.1 アプリケーションがサポートされます。WinRT 向けの Ultra Light CustDB サンプルにより、Microsoft Visual Studio 2013 および Windows Phone 8.1 がサポートされます。

### x86 Linux のサポート

Ultra Light は x86 Linux プラットフォームをサポートしています。

### ARM64 で動作する Android のサポート

Ultra Light により、64 ビット ARM プロセッサ上で動作する Android プラットフォームがサポートされます。

## 動作変更、廃止機能、およびサポートを終了した機能

**Ultra Light の BlackBerry サポートの終了** BlackBerry および J2SE インタフェースは削除されました。

### Microsoft ActiveSync プロバイダサポートの終了

Ultra Light の Microsoft ActiveSync プロバイダサポートは利用できなくなりました。

### パブリケーションアーティクルの最大サイズの増加

パブリケーションアーティクルは、2,048 バイトまでサポートされています。

新規のデータベースを作成するか、または既存のデータベースを再構築して、よりサイズの大きいパブリケーションアーティクルにアクセスします。古いデータベースでは、パブリケーションアーティクルの最大サイズは 256 バイトのままです。  
データベースファイルの最大サイズの増加

Ultra Light のデータベースファイルの最大サイズは、4 KB、8 KB、および 16 KB のページサイズについて 4 GB から 16 GB に増加されました。ページサイズが 2 KB のデータベースについては、最大サイズは 8 GB までしか増加されていません。ページサイズ 1 KB のデータベースについては、最大サイズは 1 GB に減少しました。

**Ultra Light Embedded SQL (ESQL) サンプルの削除** Embedded SQL、ESQLAuth、および ESQLSecurity サンプルの CustDB は削除されました。

**Ultra Light による 32 ビット Linux のサポート開始** Ultra Light で 32 ビット Linux プラットフォームをサポートできるようになりました。

Ultra Light の新しい `kdf_iterations` 作成オプション `kdf_iterations` は、DBKEY パラメータによって提供されるパスワードを実際の暗号化キーに変換するキー導出機能の繰り返し回数を 1000 単位で表します。この機能は、攻撃の試みを長引かせることで、暗号化されたデータベースへのアクセスをより困難にします。

## 1.6 Relay Server バージョン 17.0 の新機能

Relay Server バージョン 17.0 のいくつかの新機能、変更、廃止、削除された機能についてご紹介します。

廃止された機能は、今後のバージョンで削除される予定です。アプリケーションを変更して、廃止された機能の代わりに推奨されている代替機能をご使用ください。

### i 注記

マニュアル完成後に追加された機能情報を含む、ソフトウェアに関する追加情報については、`readme.txt` ファイルを参照してください。

## 新機能

### Relay Server のパフォーマンス向上

パフォーマンスを向上させるために、いくつかの強化が加えられています。

- Relay Server で、Apache および Microsoft IIS 8 の WebSocket プロトコルがサポートされるようになりました (旧バージョンの IIS では WebSocket トラフィックはサポートされません)。WebSocket プロトコルをサポートするアプリケーションでは、WebSocket プロトコルが自動的に使用されます。WebSocket プロトコルをサポートしないアプリケーションでは、HTTP が使用されます。
- Outbound Enabler で、Relay Server に要求を送信したり Relay Server から要求を送信したりする、専用のバックエンド接続のためのジャンクシヨンプールが使用されるようになりました。rsoc2 ユーティリティでは、ジャンクシヨンプールのサイズを制御するオプションとして、`-jsl`、`-jsh`、`-jl` がサポートされます。
- Relay Server を Microsoft Windows に配備するための新しい設定スクリプト (`iis7_plus_setup.bat`) が追加されました。

### Microsoft IIS のサポートの強化

- Microsoft IIS 8.5 で Relay Server を実行できるようになりました。
- Microsoft IIS 上の Relay Server のスレッドプールが自己制御されるようになりました。デフォルトの Web ガーデンサイズの 1 を使用することや、Relay Server 設定ファイルのオプションセクションで `min_thread` and `max_thread` プロパティを使用して Relay Server スレッドプールを制御することができます。

### Apache の Relay Server での二重の Apache サーバインスタンスのサポート

Relay Server で、2 つの Apache サーバインスタンスの設定および実行がサポートされます。Apache サーバの一方のインスタンスは、クライアントに対して表示され、クライアント要求のみ処理します。他方のインスタンスは、サーバに対して表示され、Outbound Enabler 要求のみ処理します。この設定で Apache Relay Server を実行すると、大規模な配備でのスケーラビリティが向上します。

### Relay Server のログファイルサイズの設定

Relay Server のログファイルのサイズを制限するには、Relay Server 設定ファイルのオプションセクションで log\_size\_limit オプションを使用します。

#### Microsoft IIS での設定ファイルのモニタ

Microsoft IIS で、Relay Server によって設定ファイルが継続的にモニタされ、ログファイルでステータスの変更がレポートされるようになりました。

#### バックエンドファームおよびバックエンドサーバの自動設定

新しい new auto\_config オプションが、Relay Server にバックエンドファームおよびバックエンドサーバを自動的に設定するよう指示します。この機能は Windows でのみ使用できます。

## 動作変更、廃止機能、およびサポートを終了した機能

### 旧バージョンとの互換性

バージョン 17 の Relay Server は、バージョン 16 以前の Outbound Enabler では動作しません。同様に、バージョン 17 の Outbound Enabler は、バージョン 16 以前の Relay Server では動作しません。

### rsoc ユーティリティの変更

- HTTPS 接続で、Outbound Enabler が正しいバックエンドサーバに接続されていることを確認するために、certificate\_name、certificate\_company、certificate\_unit のうちの少なくとも 1 つを指定しなければならなくなりました。証明書をチェックしないようにするには、skip\_certificate\_name\_check オプションを指定してください。
- -cs status\_url オプションが削除されました。
- ユーティリティの名称が rsoe2 に変更されました。
- TLS または HTTPS を使用して Outbound Enabler をバックエンドサーバに接続するときは、プロトコルオプション certificate\_name、certificate\_company、certificate\_unit のうちの少なくとも 1 つを指定してください。Outbound Enabler により、これらの値がバックエンドサーバの証明書と照合され、正しいバックエンドサーバに接続されていることが確認されます。この確認が実行されないようにするには、新しい skip\_certificate\_name\_check プロトコルオプションを true に設定してください。

### AdminChannel ツールの変更

ianywhere.ml.rs.AdminChannel の名称が com.sap.relayserver.AdminChannel に変更されました。

### SQL Anywhere サーバモニタのサポート

SQL Anywhere サーバモニタで Relay Server がサポートされなくなりました。ステータスページのみサポートされています。

### ファイル名の変更

以前のファイル名	新しいファイル名
rs-setup.bat	iis7_plus_setup.bat
rsoesuppXX	このファイルは削除されました
rsoe.exe	rsoe2.exe



以前のファイル名	新しいファイル名
ianywhere.ml.rs Java パッケージ修飾子	com.sap.relayserver
rs_admin.dll,rs_client.dll, rs_monitor.dll,rs_server.dll	rs.dll (すべての単一の DLL に結合)

### タイムアウト動作の変更

Relay Server および Outbound Enabler で、クライアントトラフィックが IAS-RS-App-Timeout-Minute ヘッダに従ってタイムアウトしなくなり、タイムアウト動作がクライアント、他の仲介者、バックエンドサーバ、Relay Server が含まれる Web サーバによって制御されるようになりました。IAS-RS-App-Timeout-Minute のデフォルトの 8 分のタイムアウトは適用されなくなりました。

### Relay Server Record の終わりの変更

Relay Server Record が次で終了しなくなりました。

```
<processId.threadId.F (BEFarmIdx) B (BEServerIdx) S (BESessionNum) R (RequestIdx)>
```

次で終了するようになりました。

```
<label: J{relayserver-host#backend-farm-name#backend-server-name#junction-index}  
R{request-number}>
```

### Microsoft IIS サポートの変更

- Relay Server は Microsoft IIS バージョン 7.0 以降で実行してください。バージョン 6.0 ではサポートされなくなりました。
- Relay Server は Web サービスの実行中に必要に応じて起動されるようになりました。Web サービスを停止すると、Relay Server も停止します。この変更の結果、Microsoft IIS 用の Relay Server は共有メモリ上で実行されなくなり、Web ガーデンサイズが 1 の単一のアプリケーションプールが使用されるようになりました。rshost サービスが Windows 上で Relay Server を管理するために使用できなくなり、dbsvc -t RSHOST オプションが削除されました。Windows 上で、rshost は内部状態管理のためだけに使用できます。Windows では、rstoool.jar の AdminChannel クラスを使用して Relay Server をローカルまたはリモートで管理します。Linux では、引き続き rshost を使用できます。

同様に、Relay Server は情報を次のロケーションに記録します。

ログファイル	ロケーション
設定ファイルのログ	Relay-Server-DLL-directory/rs.config
Relay Server のログ	Relay-Server-DLL-directory/Log/rs.log

### Relay Server プラグインの削除

SQL Central 用の Relay Server プラグインが削除されました。Relay Server の設定を変更するには、Relay Server 設定ファイルを編集します。

サポート対象外のプロパティ

プロパティタイプ	プロパティ	注記
オプションセクション	shared_mem	このプロパティは Microsoft IIS で無視されるようになりました。 Apache ではサポートされます。
	up_pad_size	Microsoft IIS および Apache で無視されます。
backend_farm section	client_security	この新しい動作は、以前のデフォルトと同じです (制限されない)。Web サーバの設定によって制限してください。
	backend_security	この新しい動作は、以前のデフォルトと同じです (制限されない)。Web サーバの設定によって制限してください。
	active_cookie	この新しい動作は、以前のデフォルトと同じです (標準 Cookie を注入する場合は "はい")。この Cookie は、サーバレベルのアフィニティのみとなりました。
	active_header	Relay Server でアフィニティ管理の独自のヘッダは注入されません。
	max_client_buffer	フローおよびリソース制御がクライアントごとの専用接続の自然な輻輳によって管理されるようになりました。
	renew_overlapped_cookie	ソケットレベルのアフィニティを使用できなくなったため、この設定は不要になりました。
	socket_level_affinity	ソケットレベルのアフィニティを使用できなくなりました。Relay Server は、バックエンドへの下り方向で、永続 HTTP を非永続 HTTP に変換します。遅延時間が長いネットワーク上の Relay Server クライアントの永続性は維持されます。標準 Cookie メカニズムによるサーバレベルのアフィニティのみがサポートされる動作です。

## 1.7 SQL Remote バージョン 17.0 の新機能

SQL Remote バージョン 17.0 では機能強化と動作の変更が行われています。

廃止された機能は、今後のバージョンで削除される予定です。アプリケーションを変更して、廃止された機能の代わりに推奨されている代替機能をご使用ください。

### i 注記

マニュアル完成後に追加された機能情報を含む、ソフトウェアに関する追加情報については、`readme.txt` ファイルを参照してください。

## 強化

### dbremote オフライントランザクションログの取得の強化

dbremote ユーティリティを使用して、SQL Anywhere データベースサーバに直接アクセスするのではなく、SQL Anywhere データベースサーバからオフライントランザクションログを取得できるようになりました。オフライントランザクションログが必要だが、特定のオフライントランザクションログディレクトリを直接開くことができない場合や、オフライントランザクションログファイルが含まれていない場合に、dbremote はデータベースサーバを通じてオフライントランザクションログを取得します。次の制限事項があります。

- レプリケーションデータベースに接続するために dbremote で使用されるユーザ ID には READ FILE 権限と WRITE FILE 権限が必要で、すべてのオフライントランザクションログファイルがオンライントランザクションログディレクトリにある必要があります。

この機能を使用すると、データベースサーバがトランザクションログを取得するために実行する処理が増えるため、パフォーマンスが若干低下します。

## 動作の変更

### SQL Remote 抽出ユーティリティ (dbxtract) およびデータベース抽出ウィザードの変更

以前は、SQL Central で抽出ユーティリティ (dbxtract) またはデータベース抽出ウィザードを使用してデータベースを抽出するときは、デフォルトでパスワードがアンロードされていましたが、アンロードされなくなりました。パスワードを抽出するかどうかを指定する必要があり (dbxtract の -up オプション、およびデータベース抽出ウィザードの新しい UI コントロール)、データベースを抽出するための ACCESS USER PASSWORD システム権限が必要になりました。

## 1.8 SQL Anywhere の以前のリリースの新機能

次のリンクを使用して、以前のリリースの SQL Anywhere のマニュアルにアクセスします。そこで、機能がいつどのように追加、強化、廃止、削除されたかを確認できます。

バージョン	新機能、動作変更、廃止機能へのリンク
16.0.0	<a href="http://dcx.sap.com/sa160/en/sachanges/sa-16-nagano.html">http://dcx.sap.com/sa160/en/sachanges/sa-16-nagano.html</a>
12.0.1	<a href="http://dcx.sap.com/1201/en/sachanges/new1201.html">http://dcx.sap.com/1201/en/sachanges/new1201.html</a>
12.0.0	<a href="http://dcx.sap.com/1200/en/sachanges/newinnsbruck.html">http://dcx.sap.com/1200/en/sachanges/newinnsbruck.html</a>
11.0.1	<a href="http://dcx.sap.com/1101/en/sachanges_en11/new1101.html">http://dcx.sap.com/1101/en/sachanges_en11/new1101.html</a>

バージョン	新機能、動作変更、廃止機能へのリンク
11.0.0	<a href="http://dcx.sap.com/1101/en/sachanges_en11/newpanorama.html">http://dcx.sap.com/1101/en/sachanges_en11/newpanorama.html</a> 
6.0.1 ~ 10.0.1	<a href="http://dcx.sap.com/1001/en/dbwnen10/wn-new1001.html">http://dcx.sap.com/1001/en/dbwnen10/wn-new1001.html</a> 

## 1.9 SQL Anywhere の最新バージョンへのアップグレード方法

メジャーリリース、サービスパック、パッチレベルの更新には、データベースをアップグレードまたは再構築しなければならない可能性のあるソフトウェアの機能強化が含まれています。

### メジャーリリース、サービスパック、パッチレベルの更新の違い

#### メジャーリリース

メジャーリリースは、前回のメジャーリリース以降のすべての新機能および修正から構成されます。メジャーリリースでは新しいメジャー番号が取得されます (たとえば、16.0 の代わりに 17.0 など)。


#### Service Pack (SP)

サービスパックは、前回のサービスパックまたはメジャーリリース以降のバグフィックスおよび新機能または変更された機能から構成されます。サービスパックでは新しいマイナー番号が取得されます (たとえば、17.1 の代わりに 17.2 など)。

#### パッチレベルの更新 (PL)

パッチレベルの更新は、バグフィックスから構成されます。

### 既存のデータベース、データベースサーバ、管理ツールとの互換性

詳細については、SQL Anywhere がサポートするプラットフォームおよびエンジニアリングサポート状況ページ (<http://scn.sap.com/docs/DOC-35654> ) を参照してください。

### 既存のソフトウェアとの互換性

バージョン 17 の SQL Anywhere データベースサーバは、バージョン 12 以降のソフトウェアを使用するクライアントアプリケーションからの接続をサポートしています。

バージョン 17 の SQL Anywhere クライアントは、バージョン 12 以降の SQL Anywhere データベースサーバに接続できません。

バージョン 17 の SQL Anywhere データベースサーバは、サポートされているすべての TDS クライアントバージョンからの接続をサポートしています (SAP Open Client バージョン 15、16、またはそれ以降、および jConnect JDBC ドライバのバージョン 7 以降)。

バージョン 17 の Mobile Link サーバは、バージョン 12 以降の同期クライアントからの接続をサポートしています (Ultra Light および Mobile Link SQL Anywhere クライアントユーティリティ (dbmlsync))。バージョン 17 の同期クライアントは、バージョン 16 以降の Mobile Link サーバに接続できます。

SQL Anywhere サポートパッケージとパッチレベルアップグレードすべてに関する `readme.txt` ファイルは、<http://sqlasupport.sap.com/readme/index.html> にあります。

このセクションの内容:

#### [SQL Anywhere のアップグレード \[70 ページ\]](#)

このバージョンのソフトウェアで既存のアプリケーションを使用する前に、動作の変更のリストを確認して、アプリケーションに影響がないかどうかを確認してください。

#### [Mobile Link のアップグレード \[103 ページ\]](#)

アップグレードを行う前に、影響がありそうな動作の変更を確認し、一般的なアップグレード前の対応策を実行してください。

#### [Ultra Light のアップグレード \[116 ページ\]](#)

このバージョンのソフトウェアで既存の Ultra Light アプリケーションを使用する前に、新機能と動作の変更のリストを確認して、アプリケーションに影響がないかどうかを確認してください。

#### [SQL Remote のアップグレード \[120 ページ\]](#)

SQL Remote のアップグレードは一度に 1 サイトずつ実行でき、アップグレード時には、最新バージョンでサポートされていない、旧バージョンの SQL Remote で使用可能だった機能に対処する必要があります。

#### [モニタのアップグレードおよびリソースの移行 \[121 ページ\]](#)

モニタをアップグレードするには、新しいソフトウェアをインストールし、リソースを移行します。メジャーリリース間でモニタをアップグレードするときは、ソフトウェアインストーラを使用し、リソース移行のプロンプトに従います。メジャーリリースへアップグレードするときは、ソフトウェアをインストールし、移行ユーティリティを実行してリソースを移行します。

#### [バージョン 16 のデータベースのアップグレードによる SQL Anywhere コックピットの使用 \[123 ページ\]](#)

ユーザがコックピットに接続するために必要な COCKPIT\_ROLE ユーザ定義ロールを作成し、ユーザに行使権限を付与します。

#### [バージョン 17 のデータベースのアップグレードによる SQL Anywhere コックピットの使用 \[124 ページ\]](#)

バージョン 17 のユーザに ACCESS DISK INFORMATION システム権限を付与して、バージョン 17 コックピット内のすべての機能を使用できるようにします。

#### [バージョン 16 の OData サーバの変換 \[125 ページ\]](#)

SQL 文を使用して ODATA PRODUCER 定義を作成し、`dbsrv17 -xs odata` プロトコルオプションを使用して、有効化および定義済の ODATA PRODUCER 文を開始します。

## 1.9.1 SQL Anywhere のアップグレード

このバージョンのソフトウェアで既存のアプリケーションを使用する前に、動作の変更のリストを確認して、アプリケーションに影響がないかどうかを確認してください。

### i 注記

マニュアル完成後に追加された機能情報を含む、ソフトウェアに関する追加情報については、`readme.txt` ファイルを参照してください。

このセクションの内容:

#### [バージョン 10 以降のデータベースのアップグレード \[71 ページ\]](#)

バージョン 10 以降からアップグレードするには、データベースをアップグレードするか再構築します。

#### [バージョン 9 以前のデータベースのアップグレード \[71 ページ\]](#)

バージョン 9 以前からバージョン 17 にアップグレードするには、データベースを再構築します。再構築するには、古いデータベースをアンロードしてから、バージョン 17 の新しいデータベースに再ロードします。

#### [ユーザ定義のアップグレード \[72 ページ\]](#)

システムアップグレードの実行に加えて、ユーザ定義のアップグレードの SQL 文を `.sql` スクリプトファイルに格納することによって、独自のアップグレードプロセスを定義することができます。

#### [アップグレードと再構築の注意事項 \[73 ページ\]](#)

SQL Anywhere はさまざまな設定で使用されるので、すべての場合に対応できるアップグレード手順があるわけではありません。ソフトウェアをアップグレードする前に、次の作業を行う必要があります。

#### [複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

SAP SQL Anywhere の複数のバージョンが同じ Windows コンピュータにインストールされている場合は、ユーティリティを使用するとき、システムのパスをチェックしてください。ソフトウェアのインストールでは、その実行プログラムのディレクトリがシステムパスの最後に追加されるため、最新バージョンをインストールしたにもかかわらず、誤って旧バージョンを実行してしまう場合があります。

#### [データベースの再構築 \(アンロード/再ロード\) \(コマンドラインの場合\) \[75 ページ\]](#)

以前のデータベースをソフトウェアの最新バージョンに再構築します。

#### [バージョン 10 以降のデータベースの再構築手順 \[76 ページ\]](#)

データベースを再構築するには、データベースをアンロードしてから再ロードすることでファイルフォーマットをアップグレードします。ファイルフォーマットをアップグレードすると、ディスク上でデータの保管とアクセスに使用されるフォーマットが変更され、ソフトウェアの最新バージョンの新機能とパフォーマンス向上をすべて利用できます。

#### [バージョン 9 以前のデータベースの再構築手順 \[80 ページ\]](#)

バージョン 9 以前のデータベースの再構築に固有の制限事項と考慮事項があります。データベースは、常にバックアップしてから再構築します。

#### [バージョン 10 以降のデータベースのアップグレード手順 \[87 ページ\]](#)

データベースをアップグレードすると、バージョン 17 の機能を有効にするために、システムテーブル、システムプロシージャ、データベースオプションが追加、変更されます。ディスク上でデータが保管、アクセスされるファイルフォーマットは変更されず、ソフトウェアの最新バージョンの新機能とパフォーマンス向上をすべて利用できるわけではありません。

#### [データベースミラーリングシステムでのアップグレードと再構築 \[92 ページ\]](#)

データベースミラーリングシステム内のすべてのデータベースサーバで、同じマイナーリリースを使用する必要があります。

[データベースアップグレードのトラブルシューティング: サポートパッケージの適用時は JDBC アプリケーションが動作していないことを確認してください \[99 ページ\]](#)

サポートパッケージの適用後、JDBC アプリケーションが動作を停止する場合があります。

[データベースアップグレードのトラブルシューティング: 集合関数と外部参照 \[101 ページ\]](#)

SQL Anywhere は、サブクエリでの集合関数の使用法を明確に規定した ISO SQL 標準に準拠しています。この変更は、旧バージョン用に記述された文の動作に影響を与えます。以前は有効だったクエリでエラーメッセージが生成され、結果セットが変わる可能性があります。

[SQL Anywhere のインストール \(UNIX\) \[102 ページ\]](#)

コマンドラインを使用して SQL Anywhere をインストールします。

[SQL Anywhere のインストール \(Windows\) \[103 ページ\]](#)

Windows で SQL Anywhere をインストールします。

## 1.9.1.1 バージョン 10 以降のデータベースのアップグレード

バージョン 10 以降からアップグレードするには、データベースをアップグレードするか再構築します。

バージョン 17 のソフトウェアでバージョン 10 以降のデータベースを使用できるので、アップグレードや再構築を実行するかどうかは任意です。ただし、バージョン 17 の新機能をすべて利用するには、データベースを再構築します。

### i 注記

データベースサーバをアップグレードした後、またはアップグレード後のデータベースサーバで使用できるようにデータベースを再構築またはアップグレードした後は、データベースのマテリアライズドビューを再表示してください。

## 関連情報

[バージョン 10 以降のデータベースのアップグレード手順 \[87 ページ\]](#)

[バージョン 10 以降のデータベースの再構築手順 \[76 ページ\]](#)

[複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

## 1.9.1.2 バージョン 9 以前のデータベースのアップグレード

バージョン 9 以前からバージョン 17 にアップグレードするには、データベースを再構築します。再構築するには、古いデータベースをアンロードしてから、バージョン 17 の新しいデータベースに再ロードします。

バージョン 9 以前のデータベースを開始しようとすると、データベースの開始時にエラーが発生します。既存のデータベースを再構築するには、いくつかの方法があります。

- -an (データベースの新規作成) オプションまたは -ar (古いデータベースの置き換え) オプションを指定して、バージョン 17 のアンロードユーティリティ (dbunload) を使用します。

#### i 注記

アンロード ユーティリティ (dbunload) には、SQL Anywhere の全バージョンで同じファイル名が使用されています。正しいバージョンを使用していることを確認してください。使用しているアンロードユーティリティのバージョンを確認するには、`dbunload -?` コマンドを実行します。

- バージョン 17 のアンロードユーティリティを使用してデータベースをアンロードしてから、バージョン 17 のデータベースサーバにある `reload.sql` ファイルを使用してデータベースを再ロードします。スキーマを変更する必要がある場合は、この方法を使用してアップグレードすることをお奨めします。スキーマの変更後は、データベースを作成して、再ロードスクリプトを適用できます。
- SQL Central の **データベースアンロードウィザード** を使用します。新しいデータベースを作成するか、既存のデータベースを新しいデータベースに置き換えるか、データベースをファイルにアンロードします。
- 旧バージョンの dbunload を使用してデータベースをアンロードし、`reload.sql` ファイルとバージョン 17 のデータベースサーバを使用してデータベースを再ロードします。廃止予定またはサポート対象外のデータベースオプション設定、オブジェクト、または SQL 構文が `reload.sql` ファイルにアンロードされる可能性があるため、この方法は他の方法が失敗する場合にのみ使用してください。再ロード中に問題が発生したら、ファイルを手動で編集してください。バージョン 17 では、内部再ロード機能により、このような問題の多くが解決されています。

#### i 注記

Mac OS X 用の SQL Anywhere 9.0.2 は PPC でサポートされ、SQL Anywhere 10 以降は Intel でサポートされていました。Mac OS X に 9.0.2 以前のバージョンのデータベースがある場合は、次の 2 つの方法でデータベースをアンロードできます。

- バージョン 9.0.2 のソフトウェアを使用してデータベースをアンロードします。
- SQL Anywhere17 がインストールされている別のプラットフォームにデータベースをコピーしてから、バージョン 17 のソフトウェアを使用してデータベースをアンロードします。

データベースをアンロードした後は、Mac OS X でバージョン 17 のソフトウェアを使用してデータベースを再ロードできます。

## 関連情報

[複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

### 1.9.1.3 ユーザ定義のアップグレード

システムアップグレードの実行に加えて、ユーザ定義のアップグレードの SQL 文を `.sql` スクリプトファイルに格納することによって、独自のアップグレードプロセスを定義することができます。

ユーザ定義のアップグレードを使用して、データベースのアトミックなアップグレードを実行することができます。データベースに加えられた変更だけが、スクリプトファイルに一覧表示されます。スクリプトの実行中にエラーが発生すると、データベース



は停止し (または再起動し)、アップグレードがロールバックされます。通常、スクリプトファイル内の SQL 文はすべて、スクリプトの一部として実行されますが、トランザクションログエントリは生成されません。

次の文の制限事項および考慮事項が、アップグレードスクリプトファイルの内容に適用されます。

- ALTER DATABASE 文と BACKUP DATABASE 文はスクリプトファイルで許可されません。スクリプトファイルにこれらの文のいずれかが含まれている場合、アップグレードは停止し、データベースはアップグレードの開始時に自動的にロールバックされたチェックポイントまでロールバックされます。
- CHECKPOINT 文は無視されます。
- アップグレードがロールバックされ、データベースサーバが停止されていない場合、CALL sa\_server\_option 文によって引き起こされたアクションはどれもロールバックされません。
- 結果セットを返す SELECT 文と CALL 文は呼出元に結果セットを返しません、実行はされます。この動作により、意図しない副次的影響が生じ、アップグレードのパフォーマンスに影響を与えます。対処方法として、結果のレポートが必要な場合に、スクリプト内でこれらの文をカーソルまたは FOR ループとともに使用できます。
- ミラーリング構成のアップグレードされたプライマリサーバは、アップグレードされていないミラーサーバやコピーノードで使用しないでください。ミラーおよびコピーノードデータベースは、開始前に、アップグレードされたデータベースファイルのコピーに置き換える必要があります。
- システムのアップグレードやアンロード/再ロードも実行されていない場合、アップグレードされたデータベースは以前のデータベースサーバで使用できます。

システムアップグレードとの類似点は次のとおりです。

- ミラーリングされたデータベースは、ミラーリングの設定時に実行されている間、アップグレードはできません。
- アップグレード時に実行された文は、トランザクションログに記録されません。
- アップグレード時にデータベースへのその他の接続がないようにする必要があります。
- アップグレードに成功すると、トランザクションログは名称変更されます。
- アップグレードが完了次第、データベースをバックアップします。アップグレードの一部として加えられた変更は含まれていないため、既存のバックアップは無効になります。

ユーザ定義のアップグレードを実行するには、ALTER DATABASE 文または SQL Central の **データベースアップグレードウィザード** を使用してください。アップグレードユーティリティ (dbupgrad) では、ユーザ定義のアップグレードはサポートされていません。

## 1.9.1.4 アップグレードと再構築の注意事項

SQL Anywhere はさまざまな設定で使用されるので、すべての場合に対応できるアップグレード手順があるわけではありません。ソフトウェアをアップグレードする前に、次の作業を行う必要があります。

### アップグレード前の JRE ディレクトリのバックアップ

アップグレードにより、JRE ディレクトリ (%SQLANY17%\binXX\jre180) およびそのサブディレクトリが上書きされる場合があります。この場合、証明書を使用していれば、証明書が含まれる証明書ストア (%SQLANY17%\binXX\jre180\lib\security\cacerts) が上書きされます。同様に、管理ツールで文字を表示するために %SQLANY17%\binXX\jre180\lib\fonts\fallback ディレクトリに追加したフォントが失われる場合があります。JRE の変更についてのアップグレードステップを最小限にするには、必要に応じてバックアップからファイル (cacert など) をリストアできるように、アップグレード前に JRE ディレクトリおよびそのすべてのサブディレクトリのバックアップコピーを作成します。設定をリストアするには、java\_vm\_options オプション (SQL Anywhere) や -sl java オプション (Mobile Link) を使用して、JVM 起動設定を最適化します。

### 動作の変更のチェック

記述されている動作変更が既存のアプリケーションに影響を与えないことを確認してください。影響する場合は、既存のアプリケーションを更新してください。

アプリケーションのテスト

事前に SQL Anywhere17 環境でアプリケーションのテストを徹底的に行ってから、実際の運用に使用するアプリケーションをアップグレードしてください。

正しいバージョンのユーティリティの使用

新しいデータベースには、正しいバージョンのデータベースユーティリティを使用してください。

データベースの検証とバックアップ

アップグレードを開始する前に、データベースを検証し、ソフトウェアとデータベースをバックアップします。後でリカバリできるように、アップグレードの終了時にデータベースをバックアップします。

アップグレードの前に同期

Ultra Light データベースや、Mobile Link インストール環境の SQL Anywhere リモートデータベースなど、同期に関連するデータベースの場合は、同期を正しく行ってからアップグレードしてください。

アップグレード手順のテスト

アップグレード手順は、十分にテストしてから運用システムで実行してください。

## 関連情報

[SQL Anywhere バージョン 17.0 の新機能 \(製品全体\) \[13 ページ\]](#)

[複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

### 1.9.1.5 複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法

SAP SQL Anywhere の複数のバージョンが同じ Windows コンピュータにインストールされている場合は、ユーティリティを使用するとき、システムのパスをチェックしてください。ソフトウェアのインストールでは、その実行プログラムのディレクトリがシステムパスの最後に追加されるため、最新バージョンをインストールしたにもかかわらず、誤って旧バージョンを実行してしまう場合があります。

たとえば、パス内でバージョン 12 の実行プログラムのディレクトリがバージョン 17 の実行プログラムのディレクトリよりも前にある場合に dbinit コマンドを使用すると、バージョン 12 のユーティリティを使用することになり、その結果、バージョン 12 のデータベースが作成されてしまいます。

バージョン 17 のユーティリティが確実に使用されるようにするには、いくつかの方法があります。

- バージョン 17 の実行プログラムディレクトリが、旧バージョンの実行プログラムディレクトリより前になるようにシステムパスを変更します。
- 実行するユーティリティの正確な場所を示すユーティリティ名への完全修飾パスを指定します。
- 正しいバージョンのユーティリティが使用されるように、環境を変更するスクリプトを作成します。
- 旧バージョンのソフトウェアをアンインストールします。

## 1.9.1.6 データベースの再構築 (アンロード/再ロード) (コマンドラインの場合)

以前のデータベースをソフトウェアの最新バージョンに再構築します。

### 前提条件

バージョン 17 のデータベースでは、次のシステム権限が必要です。

- BACKUP DATABASE
- VALIDATE ANY OBJECT
- SERVER OPERATOR
- SELECT ANY TABLE

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

データベースをバックアップします。次に例を示します。

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=passwd" old-db-backup-dir
```

### 手順

1. 断片化されているドライブでは、データベースのパフォーマンスが低下するので、可能な場合は、新しいデータベースを保存するドライブの断片化を解除してください。
2. バージョン 17 の dbunload ユーティリティは、旧バージョンのデータベースサーバで実行されているデータベースに対して使用できないので、すべての SQL Anywhere データベースサーバを停止します。次に例を示します。

```
dbstop -c "DBF=mydb.db;UID=DBA;PWD=passwd"
```

3. 既存のデータベースをアンロードして、新しいバージョン 17 のデータベースに再ロード (再構築) します。次に例を示します。

```
dbunload -c "DBF=mydb.db;UID=DBA;PWD=passwd" -an mydb17.db
```

4. 新しいデータベースは、バックアップしてから使用します。次に例を示します。

```
dbbackup -c "DBF=mydb17.db;UID=DBA;PWD=passwd" new-db-backup-dir
```

5. 新しいデータベースは、検証してから使用します。次に例を示します。

```
dbvalid -c "DBF=mydb17.db;UID=DBA;PWD=passwd"
```

### 結果

データベースが最新バージョンに再構築されます。デフォルトでは、データベースが停止されてから再起動されます。

## 次のステップ

アプリケーションを使用して、再構築されたデータベースをテストします。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

[アップロードユーティリティを使用したバージョン 9 以前のデータベースの再構築 \(dbunload\) \[85 ページ\]](#)

[バージョン 9 以前のデータベースの再構築 \(SQL Central\) \[84 ページ\]](#)

### 1.9.1.7 バージョン 10 以降のデータベースの再構築手順

データベースを再構築するには、データベースをアンロードしてから再ロードすることでファイルフォーマットをアップグレードします。ファイルフォーマットをアップグレードすると、ディスク上でデータの保管とアクセスに使用されるフォーマットが変更され、ソフトウェアの最新バージョンの新機能とパフォーマンス向上をすべて利用できます。

#### 警告

大規模なデータベースのアンロードや再ロードには時間がかかり、大容量のディスク領域が必要になることがあります。この処理は、アンロードされたデータと新しいデータベースファイルを保持するため、データベースの約 2 倍のディスク領域を要する場合があります。

SQL Anywhere のインデックスが変更された結果、データベースをアンロードしてから再ロードして再構築すると、再構築されたデータベースが元のデータベースよりも小さくなる場合があります。このデータベースのサイズの縮小は、問題や、データが失われたことを示すものではありません。

#### 注記

データベースは、バックアップしてから再構築します。

このセクションの内容:

#### [AUTOINCREMENT カラムのあるテーブルの再ロード \[77 ページ\]](#)

dbunload の `-I` オプションを指定すると、再構築されたデータベースの AUTOINCREMENT カラムに次に使用可能な値を保持できます。このオプションを指定すると、AUTOINCREMENT 値が含まれるテーブルごとに、生成される `reload.sql` スクリプトに `sa_reset_identity` システムプロシージャへの呼び出しが追加され、`SYSTABCOL.max_identity` の現在の値が維持されます。

#### [データベースの再構築 \(SQL Central\) \[77 ページ\]](#)

データベースアンロードウィザードを使用して、バージョン 10 以降の SQL Anywhere データベースを最新バージョンにアップグレードします。

#### [データベースの再構築 \(コマンドライン\) \[78 ページ\]](#)

dbunload ユーティリティを使用して、バージョン 10 以降の SQL Anywhere データベースを最新バージョンにアップグレードします。

## 1.9.1.7.1 AUTOINCREMENT カラムのあるテーブルの再ロード

dbunload の `-I` オプションを指定すると、再構築されたデータベースの AUTOINCREMENT カラムに次に使用可能な値を保持できます。このオプションを指定すると、AUTOINCREMENT 値が含まれるテーブルごとに、生成される `reload.sql` スクリプトに `sa_reset_identity` システムプロシージャへの呼び出しが追加され、`SYSTABCOL.max_identity` の現在の値が維持されます。

## 1.9.1.7.2 データベースの再構築 (SQL Central)

データベースアンロードウィザードを使用して、バージョン 10 以降の SQL Anywhere データベースを最新バージョンにアップグレードします。

### 前提条件

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

バージョン 10、11、12 のデータベースにアップグレードする場合、システム権限は不要です。

バージョン 16 以降のデータベースのアップグレードをするには、SELECT ANY TABLE および SERVER OPERATOR システム権限が必要です。

データベースのアンロードと再ロードを行う前に、データベースをバックアップしてください。例:

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=passwd" old-db-backup-dir
```

### 手順

1. **▶ スタート ▶ プログラム ▶ SQL Anywhere17 ▶ 管理ツール ▶ SQL Central ▶**の順にクリックします。
2. アップグレードするデータベースを実行するためにバージョン 17 のデータベースサーバを起動してから、SQL Central からデータベースに接続します。
3. **▶ ツール ▶ SQL Anywhere17 ▶ データベースのアンロード ▶**をクリックします。
4. データベースアンロードウィザードの最初のページの説明を読んで、**次へ**をクリックします。
5. **現在のバージョンのサーバで実行中のデータベースをアンロードする**をクリックし、リストからデータベースを選択します。**次へ**をクリックします。
6. **[新しいデータベースへのアンロードと再ロード]**を選択します。**次へ**をクリックします。
7. データベースの新しいファイル名を指定します。
8. 新しいデータベースのページサイズを指定することもできますが、指定するページサイズは、データベースサーバのページサイズ以下である必要があります。デフォルトのページサイズは 4096 バイトです。必要に応じてデータベースファイルを暗号化できます。強力な暗号化を選択した場合は、データベースを起動するたびに暗号化キーが必要です。**次へ**をクリックします。

9. 構造とデータをアンロードをクリックします。必要に応じて、他のオプションも選択します。次へをクリックします。
10. すべてのデータベースオブジェクトをアンロードをクリックします。次へをクリックします。
11. アンロード/再ロードが完了したときに新しいデータベースに接続するかどうかを指定します。
12. 完了をクリックすると、処理が開始します。

## 結果

データベースが最新バージョンにアップグレードされます。デフォルトでは、データベースが停止されてから再起動されます。

## 次のステップ

新しいデータベースを検査し、再構築が正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

### 1.9.1.7.3 データベースの再構築 (コマンドライン)

dbunload ユーティリティを使用して、バージョン 10 以降の SQL Anywhere データベースを最新バージョンにアップグレードします。

## 前提条件

`connection-string` で指定されたデータベースユーザは、SELECT ANY TABLE および SERVER OPERATOR システム権限を持っている必要があります。

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

アップグレードするデータベースに排他的にアクセスできること、またシステムパスで、バージョン 17 のユーティリティのパスがその他のユーティリティのパスより前に指定されていることを確認します。

データベースのアンロードと再ロードを行う前に、データベースをバックアップしてください。次に例を示します。

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=passwd" old-db-backup-dir
```

## コンテキスト

バージョン 10 以降のデータベースで dbunload を使用する場合は、使用する dbunload のバージョンが、データベースへのアクセスに使用するデータベースサーバのバージョンと一致する必要があります。dbunload のバージョンがデータベースサーバのバージョンより古いまたは新しい場合は、エラーが返されます。

dbunload ユーティリティを使用して、Mobile Link インストール環境のリモートデータベース、または SQL Remote レプリケーションに使用するデータベースを再構築する場合は、-ar オプションまたは -an オプションを必ず使用してください。これらのオプションによって、新しいデータベースのトランザクションログのオフセットが、古いデータベースと同じになるように設定されます。

## 手順

1. アンロードユーティリティ (dbunload) を実行し、-an オプションを使用してデータベースを作成します。

```
dbunload -c "connection-string" -an new-db-file
```

このコマンドにより、データベースが作成されます。既存のデータベースをアップグレード後のデータベースに置き換えるには、-an オプションの代わりに -ar オプションを指定します。-ar オプションを使用するには、パーソナルデータベースサーバに接続するか、アンロードユーティリティ (dbunload) と同じコンピュータ上にあるネットワークデータベースサーバに接続します。

2. 再ロードしたデータベースを使用する前に、データベースを停止し、トランザクションログを圧縮します。

アンロードと再ロードのときに (大文字と小文字を "区別する" から "区別しない" に変更するなど) データベースの特性を変更する場合、手順はもう少し複雑になります。

## 結果

データベースが最新バージョンにアップグレードされます。デフォルトでは、データベースが停止されてから再起動されます。

## 次のステップ

新しいデータベースを検査し、再構築が正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

[複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

## 1.9.1.8 バージョン 9 以前のデータベースの再構築手順

バージョン 9 以前のデータベースの再構築に固有の制限事項と考慮事項があります。データベースは、常にバックアップしてから再構築します。

### 警告

大規模なデータベースのアンロードや再ロードには時間がかかり、大容量のディスク領域が必要になることがあります。この処理では、アンロードされたデータと新しいデータベースファイルを保持するため、データベースの 2 倍のディスク領域にアクセスする必要があります。

考慮事項	詳細
Mac OS X のサポート	<p>Mac OS X 用の SQL Anywhere 9.0.2 は PPC でサポートされ、SQL Anywhere 10.0.0 以降は Intel でサポートされていました。Mac OS X に 9.0.2 以前のバージョンのデータベースがある場合は、次の 2 つの方法でデータベースをアンロードできます。</p> <ul style="list-style-type: none"><li>バージョン 9.0.2 のソフトウェアを使用してデータベースをアンロードします。</li><li>SQL Anywhere17 がインストールされている別のプラットフォームにデータベースをコピーしてから、バージョン 17 のソフトウェアを使用してデータベースをアンロードします。</li></ul> <p>データベースをアンロードした後は、Mac OS X でバージョン 17 のソフトウェアを使用してデータベースを再ロードできます。</p>



考慮事項	詳細
バージョン 17 のツールを使用してバージョン 9 以前のデータベースを再構築するときの、アップグレードの制限事項	<ul style="list-style-type: none"> <li>旧バージョンの全データベースサーバのデータベースを切断して、コンピュータで実行している旧バージョンのデータベースサーバを停止する必要があります。また、コンピュータで実行しているバージョン 17 のデータベースサーバも停止する必要があります。dbunload によってこれらの状態が検出され、処理を続行できない場合は、エラーが発生し、再構築に失敗します。</li> <li>再構築前のデータベースの dbunload 接続文字列に、ENG、START、または LINKS 接続パラメータを含めないでください (-c オプションで指定)。これらの接続パラメータを指定すると、無視され、警告が表示されます。SQL CentralSybase Central の接続ウィンドウのサーバ名フィールドまたは開始行フィールドには、値を入力しないでください。</li> <li>既存のデータベースへの直接ファイルシステムアクセスを使用して、コンピュータで dbunload を実行します (dbunload が、共有メモリを使用してデータベースに接続できる状態である必要があります)。</li> <li>再構築を実行するコンピュータで、dbunload_support_engine という名前のデータベースサーバを実行しないでください。</li> <li>バージョン 9 以前のデータベースは、データベースファイルのリカバリが必要となった場合に、アンロードしないでください。リカバリが必要なデータベースファイルでアンロードユーティリティ (dbunload) を実行すると、データベースを起動できないことを示すメッセージが返されます。dbunload を再試行する前に、バージョン 9 のデータベースサーバを使用して、データベースを起動してから停止します。バージョン 9 以降のデータベースをアンロードするには、データベースを読み込み専用モードで開始できるようにしておく必要があります。</li> </ul>
パスワードの大文字と小文字の区別	<p>新しく作成された SQL Anywhere17 データベースでは、データベースでの設定にかかわらず、すべてのパスワードは大文字と小文字が区別されます。</p> <p>既存のデータベースを再構築する場合、SQL Anywhere でのパスワードの大文字と小文字の区別は、次のように決まります。</p> <ul style="list-style-type: none"> <li>パスワードを最初に入力したのが大文字と小文字を区別しないデータベースだった場合、そのパスワードの大文字と小文字は区別されません。</li> <li>パスワードを最初に入力したのが大文字と小文字を区別するデータベースだった場合、大文字のパスワードと、大文字と小文字が混在したパスワードでは、大文字と小文字が区別されます。ただし、パスワードをすべて小文字で入力した場合、パスワードの大文字と小文字は区別されません。</li> <li>既存のパスワードと新しいパスワードの両方に加えられた変更は、大文字と小文字が区別されます。</li> </ul>

考慮事項	詳細
パスワードの長さ	新しいデータベースまたはアップグレードされたデータベースのデフォルトの最小パスワード長は 6 文字です。以前のパスワードはアップロード/再ロード時に変更されませんが、いったんパスワードを変更すると、新しいパスワードは min_password_length データベースオプションで指定された最小パスワード長の要件を満たす必要があります。
ページサイズ	SQL Anywhere17 データベースのデフォルトのデータベースページサイズは 4096 バイトです。バージョン 17 でサポートされるページサイズは、4096、8192、16384、32768 バイトです。既存のデータベースで、サポートされていないページサイズが使用されている場合は、新しいデータベースのページサイズはデフォルトで 4096 バイトに設定されます。別のページサイズを指定するには、dbinit の -p オプションまたは dbunload の -ap オプションを使用します。
照合	<p>バージョン 9 以前のデータベースサーバでは、CHAR データ型で使用される 1 つの照合しかサポートしていませんでした。この照合は、SQL Anywhere 照合アルゴリズム (SACA) を使用していました。バージョン 10 以降のデータベースサーバでは、照合アルゴリズムとして、SACA と UCA (Unicode 照合アルゴリズム) の 2 つをサポートしています。再構築されたデータベース用に新しい照合や別の照合を指定していない場合は、既存のデータベースの SACA 照合がアンロードされ、再構築されたデータベースで再利用されます。</p> <p>カスタム照合を使用してデータベースを再構築する場合に 1 ステップで再構築すると、その照合は保持されます (内部アンロード)。データベースをアンロードしてから、作成したデータベースにスキーマとデータをロードする場合は、提供されるいずれかの照合を使用する必要があります。</p>
データベースのファイルサイズ	SQL Anywhere のインデックスが変更された結果、データベースをアンロードしてから再ロードして再構築すると、再構築されたデータベースが元のデータベースよりも小さくなる場合があります。このデータベースのサイズの縮小は、問題や、データが失われたことを示すものではありません。

## 既知の問題

dbunload または [データベースアンロードウィザード](#) を実行したときに再構築処理に失敗した場合は、次の手順に従って失敗の原因を診断できます。

1. 新しい、バージョン 17 の空のデータベースを作成します。

```
dbinit -dba DBA,passwd test.db
```

2. reload.sql ファイルを空のデータベースに適用します。

```
dbisql -c "DBF=test.db;UID=DBA;pwd=passwd" reload.sql
```

3. reload.sql ファイルを新しいデータベースに適用したときに受信したメッセージに基づいて、元のデータベースまたは reload.sql ファイルに変更を加えます。

次の表に、再構築の失敗につながる既知の問題とその解決策を示します。

既知の問題	解決策
テーブル名のプレフィクスに所有者名が付いている場合は、プロシージャまたはトリガ内の DECLARE LOCAL TEMPORARY TABLE 文により、構文エラーが発生します。	所有者名を削除します。
CREATE TRIGGER 文で、トリガが定義されたテーブルの所有者名が指定されておらず、ユーザが reload.sql ファイルを実行して参照するときに、テーブルが所有者名で修飾されている必要がある場合、CREATE TRIGGER 文は失敗し、テーブルが見つからなかったことを示すエラーが返されます。	テーブル名のプレフィクスに所有者名を付けます。
オブジェクト名 (テーブル、カラム、変数、パラメータ名など) が SQL Anywhere の以後のバージョンで導入された予約語 (NCHAR など) に対応している場合は、再ロードに失敗します。次に例を示します。	予約語へのすべての参照を、別の名前を使用するように変更します。変数名の場合は、名前の競合を回避する @ を名前のプレフィクスとして使用するのが、一般的な規則です。
<pre>CREATE PROCEDURE p( ) BEGIN   DECLARE NCHAR INT;   SET NCHAR = 1; END;</pre>	
データベースをバージョン 9 以前の dbunload でアンロードする場合、reload.sql ファイルに ml_add_property システムプロシージャの呼び出しが含まれることがありますが、新しいバージョン 17 データベースにはこのプロシージャは存在しません。	バージョン 17 の dbunload ユーティリティを使用してデータベースをアンロードしてください。
バージョン 9 以前の dbunload を使用してデータベースをアンロードする場合、Transact-SQL 外部ジョイン (* = または =* で指定) を使用するビューを再ロードすると、それらのビューが適切に作成されないことがあります。	<p>次のセクション行を再ロードスクリプトに追加します。</p> <pre>SET TEMPORARY OPTION tsql_outer_joins='on';</pre> <p>Transact-SQL 外部ジョインを使用しているすべてのビューを書き直します。</p>
CREATE PROCEDURE 文と ALTER PROCEDURE 文で [NOT] DETERMINISTIC 句がサポートされていません。この句を指定すると、再ロードに失敗し構文エラーが返されます。	[NOT] DETERMINISTIC 句が指定されたユーザ定義プロシージャを含むデータベースをアップグレードする場合、句を削除してからデータベースのアンロードと再ロードを行う必要があります。

このセクションの内容:

[バージョン 9 以前のデータベースの再構築 \(SQL Central\) \[84 ページ\]](#)

データベースアンロードウィザードを使用して、バージョン 9 以前のデータベースを再構築します。再構築されたデータベースは、最新ソフトウェアバージョンでのすべての新機能とパフォーマンス拡張をサポートします。

[アップロードユーティリティを使用したバージョン 9 以前のデータベースの再構築 \(dbunload\) \[85 ページ\]](#)

アンロードユーティリティ (dbunload) の `-an` または `-ar` オプションを使用して、バージョン 9 以前のデータベースを再構築します。再構築されたデータベースは、最新ソフトウェアバージョンでのすべての新機能とパフォーマンス拡張をサポートします。

## 関連情報

[複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

### 1.9.1.8.1 バージョン 9 以前のデータベースの再構築 (SQL Central)

データベースアンロードウィザードを使用して、バージョン 9 以前のデータベースを再構築します。再構築されたデータベースは、最新ソフトウェアバージョンでのすべての新機能とパフォーマンス拡張をサポートします。

#### 前提条件

データベースは、バックアップしてから再構築します。

データベースファイルは、SQL Anywhere SQL Anywhere17 がインストールされているコンピュータと同じコンピュータに保存されている必要があります。

データベースからはテーブルのサブセットはアンロードできません。dbunload ユーティリティを使用してアンロードしてください。

データベースアンロードウィザードで、データベースファイルがすでに実行されていると認識された場合は、アンロードを続行する前にデータベースが停止します。

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

断片化されているドライブでは、データベースのパフォーマンスが低下するので、可能な場合は、新しいデータベースを保存するドライブの断片化を解除してください。

アンロードと再ロードを行うデータベースに対して、排他アクセスを持っていることを確認してください。他のユーザは接続できません。

#### 手順

1. [スタート](#) > [プログラム](#) > [SQL Anywhere17](#) > [管理ツール](#) > [SQL Central](#) の順にクリックします。
2. [ツール](#) > [SQL Anywhere17](#) > [データベースのアンロード](#) をクリックします。
3. データベースアンロードウィザードの概要ページを読んで、[次へ](#)をクリックします。
4. [旧バージョンのサーバで実行中のデータベース、または実行中ではないデータベースをアンロードする](#)をクリックします。データベースの接続情報を指定します。[次へ](#)をクリックします。

5. [新しいデータベースへのアンロードと再ロード](#)をクリックします。[次へ](#)をクリックします。
6. データベースの新しいファイル名を指定します。[次へ](#)をクリックします。

新しいデータベースのページサイズを指定できます。バージョン 17 では、デフォルトのページサイズ (推奨値) は 4096 バイトです。

データベースファイルを暗号化できます。強力な暗号化を選択した場合は、データベースを起動するたびに暗号化キーが必要です。

7. [構造とデータをアンロード] を選択します。[次へ](#)をクリックします。
8. 再構築が完了したときに新しいデータベースに接続するかどうかを指定します。
9. [完了](#)をクリックします。新しいデータベースを調べて、再構築が正常に完了したことを確認します。

## 結果

データベースが最新バージョンにアップグレードされます。デフォルトでは、データベースが停止されてから再起動されます。

## 次のステップ

新しいデータベースを検査し、再構築が正常に完了したことを確認すると同時に、再構築されたデータベースをアプリケーションでテストします。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

### 1.9.1.8.2 アップロードユーティリティを使用したバージョン 9 以前のデータベースの再構築 (dbunload)

アンロードユーティリティ (dbunload) の `-an` または `-ar` オプションを使用して、バージョン 9 以前のデータベースを再構築します。再構築されたデータベースは、最新ソフトウェアバージョンでのすべての新機能とパフォーマンス拡張をサポートします。

## 前提条件

`connection-string` で指定されたデータベースユーザは、SELECT ANY TABLE および SERVER OPERATOR システム権限を持っている必要があります。

データベースは、バックアップしてから再構築します。

データベースファイルは、SQL Anywhere SQL Anywhere17 がインストールされているコンピュータと同じコンピュータに保存されている必要があります。

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

アンロードと再ロードを行うデータベースに対して、排他アクセスを持っていることを確認してください。他のユーザは接続できません。

システムパスで、他のユーティリティより前にバージョン 17 のユーティリティが置かれていることを確認してください。

## コンテキスト

-an オプションを使用すると、元のデータベースをそのまま残して新規データベースが作成されるので、このオプションを使用することをおすすめします。-ar オプションを使用すると、既存のデータベースが新しいバージョン 17 のデータベースに置き換わります。

## 手順

1. バージョン 17 の dbunload ユーティリティは、旧バージョンのデータベースサーバで実行されているデータベースに対して使用できないので、すべての SQL Anywhere と Adaptive Server Anywhere のデータベースサーバを停止します。  
例:

```
dbstop -c "DBF=mydb.db;UID=DBA;PWD=sql"
```

2. 断片化されているドライブでは、データベースのパフォーマンスが低下するので、可能な場合は、新しいデータベースを保存するドライブの断片化を解除してください。
3. データベースをバックアップします。例:

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir
```

4. -an または -ar オプションを指定してアンロードユーティリティ (dbunload) を実行して、新しいデータベースを作成します。

```
dbunload -c "connection-string" -an database-filename
```

例:

```
dbunload -c "DBF=mydb.db;UID=DBA;PWD=sql" -an mydb17.db
```

このコマンドを使用すると、データベースが作成されます (-an を指定した場合)。-ar オプションを指定すると、既存のデータベースは再構築後のデータベースに置き換わります。-ar オプションを使用するには、パーソナルデータベースサーバに接続するか、アンロードユーティリティ (dbunload) と同じコンピュータ上にあるネットワークデータベースサーバに接続します。

## 結果

データベースが最新バージョンにアップグレードされます。デフォルトでは、データベースが停止されてから再起動されます。

## 次のステップ

新しいデータベースを検査し、再構築が正常に完了したことを確認すると同時に、再構築されたデータベースをアプリケーションでテストします。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

[複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

### 1.9.1.9 バージョン 10 以降のデータベースのアップグレード手順

データベースをアップグレードすると、バージョン 17 の機能を有効にするために、システムテーブル、システムプロシージャ、データベースオプションが追加、変更されます。ディスク上でデータが保管、アクセスされるファイルフォーマットは変更されず、ソフトウェアの最新バージョンの新機能とパフォーマンス向上をすべて利用できるわけではありません。

このセクションの内容:

[バージョン 12 以降のデータベースのアップグレード \(SQL Central\) \[88 ページ\]](#)

データベースアップグレードウィザードを使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 12 以降のデータベースをアップグレードし、バージョン 17 の機能を有効にします。

[バージョン 10 以降のデータベースのアップグレード \(コマンドライン\) \[89 ページ\]](#)

dbupgrad ユーティリティを使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 10 以降のデータベースをアップグレードし、バージョン 17 の機能を有効にします。

[バージョン 10 以降のデータベースのアップグレード \(SQL\) \[90 ページ\]](#)

ALTER DATABASE 文を使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 10 以降のデータベースをアップグレードし、バージョン 17 の機能を有効にします。

## 1.9.1.9.1 バージョン 12 以降のデータベースのアップグレード (SQL Central)

データベースアップグレードウィザードを使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 12 以降のデータベースをアップグレードし、バージョン 17 の機能を有効にします。

### 前提条件

バージョン 12 のデータベースにアップグレードする場合、システム権限は不要です。

バージョン 16 以降のデータベースのアップグレードするには、ALTER DATABASE および BACKUP DATABASE システム権限が必要です。

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

#### 警告

データベースファイルをバックアップしてからアップグレードしてください。既存のファイルにアップグレードを適用した場合、アップグレードに失敗すると、これらのファイルは使用できなくなります。データベースをバックアップするには、BACKUP DATABASE システム権限が必要です。

### コンテキスト

データベースアップグレードウィザードでは、バージョン 9.0.2 以前のデータベースをバージョン 17 にアップグレードすることはできません。バージョン 9.0.2 以前の既存のデータベースをバージョン 17 にアップグレードするには、dbunload またはデータベースアンロードウィザードを使用してデータベースをアンロードし、再ロードする必要があります。

### 手順

1. ソフトウェアをアップグレードする場合の一般的な対応策を行います。
2. ▶ スタート ▶ プログラム ▶ SQL Anywhere17 ▶ 管理ツール ▶ SQL Central ▶ の順にクリックします。
3. SQL Anywhere17 のプラグインから、アップグレードするデータベースに接続します。データベースはバージョン 17 のデータベースサーバで実行されている必要があります。
4. ▶ ツール ▶ SQL Anywhere17 ▶ データベースアップグレード ▶ をクリックします。
5. データベースアップグレードウィザードの指示に従います。デフォルトでは、データベースが停止されてから再起動されます。
6. (省略可) アップグレードしたデータベースを使用する前に、データベースを停止し、トランザクションログをコピーしてアーカイブすることをウィザードで選択しなかった場合は、その操作を行います。



## 結果

データベースがアップグレードされ、新しい、および修正されたシステムテーブル、システムプロシージャ、データベースオプションを含むようになります。

## 次のステップ

新しいデータベースを検査し、アップグレードが正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

アップグレードによって新しいシステム権限が導入され、そのようなシステム権限は UPGRADE ROLE システム権限に自動的に追加されます。UPGRADE ROLE システム権限に付与された権限を他のロールおよびユーザに配布し、UPGRADE ROLE システム権限から取り消す必要があります。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

[バージョン 10 以降のデータベースの再構築手順 \[76 ページ\]](#)

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

### 1.9.1.9.2 バージョン 10 以降のデータベースのアップグレード (コマンドライン)

dbupgrad ユーティリティを使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 10 以降のデータベースをアップグレードし、バージョン 17 の機能を有効にします。

## 前提条件

`connection-string` に指定するデータベースユーザには ALTER DATABASE システム権限が必要です。また、このデータベースに他の接続がないことが必要です。

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

アップグレードするデータベースに排他的にアクセスできること、またシステムパスで、バージョン 17 のユーティリティがその他のユーティリティより前に指定されていることを確認します。

#### 警告

データベースファイルをバックアップしてからアップグレードしてください。既存のファイルにアップグレードを適用した場合、アップグレードに失敗すると、これらのファイルは使用できなくなります。

## 手順

1. データベースに対してアップグレードユーティリティ (dbupgrad) を実行します。

```
dbupgrad -c "connection-string"
```

2. アップグレードしたデータベースを使用する前に、データベースを停止し、トランザクションログを圧縮します。

## 結果

データベースがアップグレードされ、新しい、および修正されたシステムテーブル、システムプロシージャ、データベースオプションを含むようになります。

## 次のステップ

新しいデータベースを検査し、アップグレードが正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

アップグレードによって新しいシステム権限が導入される場合があります、そのようなシステム権限は UPGRADE ROLE システム権限に自動的に追加されます。UPGRADE ROLE システム権限に付与された権限を他のロールおよびユーザに配布し、UPGRADE ROLE システム権限から取り消す必要があります。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

[複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

### 1.9.1.9.3 バージョン 10 以降のデータベースのアップグレード (SQL)

ALTER DATABASE 文を使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 10 以降のデータベースをアップグレードし、バージョン 17 の機能を有効にします。

## 前提条件

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

データベースの所有者であるか、ALTER DATABASE システム権限を付与されている必要があります。また、このデータベースに他の接続がないことが必要です。

### 警告

データベースファイルをバックアップしてからアップグレードしてください。既存のファイルにアップグレードを適用した場合、アップグレードに失敗すると、これらのファイルは使用できなくなります。

## 手順

1. Interactive SQL、または SQL 文を実行できる別のアプリケーションからデータベースに接続します。他のどの接続も、データベースを同時に使用できません。
2. ALTER DATABASE 文を実行します。

たとえば、次の文はデータベースをアップグレードします。

```
ALTER DATABASE UPGRADE;
```

3. アップグレードしたデータベースを使用する前に、データベースを停止し、トランザクションログを圧縮します。

## 結果

データベースがアップグレードされ、新しい、および修正されたシステムテーブル、システムプロシージャ、データベースオプションを含むようになります。

## 次のステップ

新しいデータベースを検査し、アップグレードが正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

アップグレードによって新しいシステム権限が導入される場合があります、そのようなシステム権限は UPGRADE ROLE システム権限に自動的に追加されます。UPGRADE ROLE システム権限に付与された権限を他のロールおよびユーザに配布し、UPGRADE ROLE システム権限から取り消す必要があります。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

## 1.9.1.10 データベースミラーリングシステムでのアップグレードと再構築

データベースミラーリングシステム内のすべてのデータベースサーバで、同じマイナーリリースを使用する必要があります。

このセクションの内容:

[データベースの再構築を行わずにデータベースサーバのソフトウェアをミラーリングシステム用のメジャーバージョンにアップグレード \[92 ページ\]](#)

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、各データベースサーバを停止し、ソフトウェアをインストールし、新しいソフトウェアを実行するサーバ上でデータベースを起動します。データベース自体はアップグレードも再構築も行われません。

[データベースの再構築を行わずにデータベースサーバのソフトウェアをミラーリングシステム用の別のマイナーリリースにアップグレード \[94 ページ\]](#)

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、各サーバを停止し、ソフトウェアをインストールし、新しいサーバ上でデータベースを起動します。データベースミラーリングシステム内のすべてのデータベースサーバで、同じマイナーリリースを使用する必要があります。

[データベースの再構築を行わずにデータベースサーバのソフトウェアをデータベースミラーリングシステム用のサポートパッケージにアップグレード \[96 ページ\]](#)

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、各サーバを停止し、ソフトウェアをインストールし、新しいサーバ上でデータベースを起動します。

[データベースミラーリングシステムでのデータベースのアップグレードまたは再構築 \(アンロード/再ロード\) \[97 ページ\]](#)

プライマリデータベースをアップグレードし、アップグレードしたデータベースとトランザクションログをミラーデータベースにコピーします。ミラーリングシステムが一時的に停止します。通常、データベースのアップグレードは、ソフトウェアのメジャーバージョンまたはマイナーリリースを更新する場合に実行されます。

### 1.9.1.10.1 データベースの再構築を行わずにデータベースサーバのソフトウェアをミラーリングシステム用のメジャーバージョンにアップグレード

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、各データベースサーバを停止し、ソフトウェアをインストールし、新しいソフトウェアを実行するサーバ上でデータベースを起動します。データベース自体はアップグレードも再構築も行われません。

#### 前提条件

BACKUP DATABASE システム権限または VALIDATE ANY OBJECT システム権限を持っている必要があります。

デフォルトでは、ネットワークデータベースサーバの停止には SERVER OPERATOR システム権限が必要です。

## コンテキスト

運用環境で手順を実行する前に、アプリケーションを使用して非運用環境で次の手順をテストします。

データベースミラーリングシステム内のすべてのデータベースサーバで、SQL Anywhere の同じマイナーリリースを使用する必要があります。この要件により、更新中にミラーリングシステムを一時的に停止する必要があります。

## 手順

1. プライマリデータベースのバックアップを作成し、そのバックアップをコピーして検証します。

たとえば、次のコマンドを実行して、mydb.db という名称のデータベースをバックアップします。

```
dbbackup -c "DBN=mydb;SERVER=myserver;HOST=primaryhost;UID=DBA;PWD=sql" backup-dir
```

バックアップのコピーを作成するには、次のコマンドを実行します。

```
xcopy backup-dir¥mydb.db validatebackup-dir¥
```

バックアップのコピーを検証します。

```
dbvalid -c "DBF=validatebackup-dir¥mydb.db;UID=DBA;PWD=sql"
```

バックアップのコピーの検証が失敗した場合は、先に進む前に検証失敗の原因となった問題を修正します。修正を行わないと、データを失うリスクがあります。

2. 新しいソフトウェアをプライマリ、ミラー、および監視サーバにインストールします。
3. ミラーリングシステムが読み込み専用のスケールアウトに含まれる場合、コピーノード上にソフトウェアをインストールします。
4. 次の順番にサーバを停止します。
  - a. コピーノード
  - b. ミラーサーバ
  - c. プライマリサーバ
  - d. 監視サーバ

たとえば、Stop ユーティリティ (dbstop) を実行します。

```
dbstop -y -c "UID=DBA;PWD=sql;Server=myserver"
```

5. (省略可) データベースをアップグレードまたは再構築します。
6. 次の順序で新しいサーバ上でデータベースを起動します。
  - a. 監視サーバ、プライマリサーバ、ミラーサーバ
  - b. コピーノード

## 結果

ミラーリングシステムのデータベースは、ソフトウェアの新しいバージョンで実行されます。

## 次のステップ

データベースミラーリングシステムを検査し、更新が正常に完了したことを確認すると同時に、データベースミラーリングシステムをアプリケーションでテストします。

### 1.9.1.10.2 データベースの再構築を行わずにデータベースサーバのソフトウェアをミラーリングシステム用の別のマイナーリリースにアップグレード

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、各サーバを停止し、ソフトウェアをインストールし、新しいサーバ上でデータベースを起動します。データベースミラーリングシステム内のすべてのデータベースサーバで、同じマイナーリリースを使用する必要があります。

## 前提条件

BACKUP DATABASE システム権限および VALIDATE ANY OBJECT システム権限を持っている必要があります。

デフォルトでは、ネットワークデータベースサーバの停止には SERVER OPERATOR システム権限が必要です。

運用環境で手順を実行する前に、アプリケーションを使用して非運用環境で次の手順をテストします。

## コンテキスト

更新中、ミラーリングシステムは一時的に停止します。

## 手順

1. プライマリデータベースのバックアップを作成し、そのバックアップをコピーして検証します。

たとえば、次のコマンドを実行して、mydb.db という名称のデータベースをバックアップします。

```
dbbackup -c "DBN=mydb;SERVER=myserver;HOST=primaryhost;UID=DBA;PWD=passwd"  
backup-dir
```

バックアップのコピーを作成します。

```
xcopy backup-dir¥ validatebackup-dir¥
```

バックアップのコピーを検証します。

```
dbvalid -c "DBF=validatebackup-dir¥backupmydb.db;UID=DBA;PWD=passwd"
```

バックアップのコピーの検証が失敗した場合は、先に進む前に検証失敗の原因となった問題を修正します。修正を行わないと、データを失うリスクがあります。

2. 次の順番にサーバを停止します。

- a. コピーノード
- b. ミラーサーバ
- c. プライマリサーバ
- d. 監視サーバ

たとえば、Stop ユーティリティ (dbstop) を実行します。

```
dbstop -y -c "UID=DBA;PWD=passwd;Server=myserver"
```

3. 新しいソフトウェアをプライマリ、ミラー、および監視サーバにインストールします。

4. 次の順番にサーバを起動します。

- a. 監視サーバ
- b. プライマリサーバ
- c. ミラーサーバ

5. ミラーリングシステムが読み込み専用のスケールアウトに含まれる場合、コピーノード上にソフトウェアをインストールしてから、コピーノードを再起動します。

## 結果

ミラーリングシステムのデータベースは、ソフトウェアの新しいバージョンで実行されます。

## 次のステップ

データベースミラーリングシステムを検査し、アップグレードが正常に完了したことを確認すると同時に、データベースミラーリングシステムをアプリケーションでテストします。

### 1.9.110.3 データベースの再構築を行わずにデータベースサーバのソフトウェアをデータベースミラーリングシステム用のサポートパッケージにアップグレード

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、各サーバを停止し、ソフトウェアをインストールし、新しいサーバ上でデータベースを起動します。

#### 前提条件

BACKUP DATABASE システム権限が必要です。

VALIDATE ANY OBJECT システム権限が必要です。

デフォルトでは、ネットワークデータベースサーバの停止には SERVER OPERATOR システム権限が必要です。

#### コンテキスト

システムのサーバは 1 台ずつしか停止できないため、サーバソフトウェアのサポートパッケージの更新中、ミラーリングシステムは動作を続行できます。更新プロセス中、プライマリサーバからミラーサーバへのフェイルオーバーが少なくとも 1 回発生します。フェイルオーバー中、プライマリサーバとミラーサーバへの接続は切断されます。

運用環境で手順を実行する前に、アプリケーションを使用して非運用環境で次の手順をテストします。

#### 手順

1. プライマリデータベースのバックアップを作成し、そのバックアップをコピーして検証します。

たとえば、次のコマンドを実行して、mydb.db という名称のデータベースをバックアップします。

```
dbbackup -c "DBN=mydb;SERVER=myserver;HOST=primaryhost;UID=DBA;PWD=passwd"  
backup-dir
```

バックアップのコピーを作成します。

```
xcopy backup-dir¥ validatebackup-dir¥
```

バックアップのコピーを検証します。

```
dbvalid -c "DBF=validatebackup-dir¥mydb.db;UID=DBA;PWD=passwd"
```

バックアップのコピーの検証が失敗した場合は、先に進む前に検証失敗の原因となった問題を修正します。修正を行わないと、データを失うリスクがあります。

2. ミラーリングシステムが読み込み専用のスケールアウトシステムの一部である場合、各コピーノードで以下を行います。



- a. コピーノードサーバを停止します。
  - b. ソフトウェアをインストールします。
  - c. コピーノードを起動します。
3. ミラーサーバを停止します。
  4. ミラーサーバにソフトウェアをインストールします。
  5. ミラーサーバを起動し、同期状態になっていることを確認します。
  6. 監視サーバを停止します。
  7. 監視サーバにソフトウェアをインストールします。
  8. 監視サーバを起動します。
  9. プライマリデータベースに接続し、次の文を実行して、フェイルオーバーを開始します。

```
ALTER DATABASE SET PARTNER FAILOVER;
```

フェイルオーバー中、プライマリサーバとミラーサーバへの接続は切断されます。現在のプライマリサーバがミラーサーバになります。

10. ミラーサーバを停止します。
11. ミラーサーバにソフトウェアをインストールします。
12. ミラーサーバを起動し、同期状態になっていることを確認します。

## 結果

ミラーリングシステムのデータベースは、ソフトウェアの新しいバージョンで実行されます。

## 次のステップ

データベースミラーリングシステムを検査し、アップグレードが正常に完了したことを確認すると同時に、データベースミラーリングシステムをアプリケーションでテストします。

### 1.9.1.10.4 データベースミラーリングシステムでのデータベースのアップグレードまたは再構築 (アンロード/再ロード)

プライマリデータベースをアップグレードし、アップグレードしたデータベースとトランザクションログをミラーデータベースにコピーします。ミラーリングシステムが一時的に停止します。通常、データベースのアップグレードは、ソフトウェアのメジャーバージョンまたはマイナーリリースを更新する場合に実行されます。

## 前提条件

BACKUP DATABASE システム権限が必要です。

VALIDATE ANY OBJECT システム権限が必要です。

デフォルトでは、ネットワークデータベースサーバの起動と停止には SERVER OPERATOR システム権限が必要です。

データベースをアップグレードするには、ALTER DATABASE システム権限が必要です。また、このデータベースに他の接続がないことが必要です。

アンロードユーティリティ (dbunload) を使用してデータベースを再構築 (アンロード/再ロード) するには、SELECT ANY TABLE システム権限が必要です。再ロードを伴うアンロードの場合は、SERVER OPERATOR システム権限も必要です。

運用環境で手順を実行する前に、アプリケーションを使用して非運用環境で次の手順をテストします。

## 手順

1. (省略可) 新しいバージョンのソフトウェアをシステムの各コンピュータにインストールします。データベースミラーリングシステム内のすべてのデータベースサーバで、SQL Anywhere の同じメンテナンスバージョンを使用する必要があります。新しいマイナーバージョンにアップグレードする場合は、サーバの停止中にソフトウェアをインストールする必要があります。
2. プライマリデータベースのバックアップを作成し、そのバックアップをコピーして検証します。

たとえば、次のコマンドを実行して、mydb.db という名称のデータベースをバックアップします。

```
dbbackup -c "DBN=mydb;SERVER=myserver;HOST=primaryhost;UID=DBA;PWD=passwd"  
backup-dir
```

バックアップのコピーを作成します。

```
xcopy backup-dir¥ validatebackup-dir¥
```

バックアップのコピーを検証します。

```
dbvalid -c "DBF=validatebackup-dir¥mydb.db;UID=DBA;PWD=passwd"
```

バックアップのコピーの検証が失敗した場合は、先に進む前に検証失敗の原因となった問題を修正します。修正を行わないと、データを失うリスクがあります。

3. 次の順番にサーバを停止します。
  - a. コピーノード
  - b. ミラー
  - c. プライマリ
4. プライマリデータベースとトランザクションログを別のバックアップディレクトリにバックアップします。以前のバックアップ以降プライマリデータベースに変更が加えられている可能性があるため、この手順が必要です。

```
xcopy /y c:¥primary-database.db backup_dir¥  
xcopy /y c:¥primary-database.log backup_dir¥
```

5. 次のオプションのいずれかを選択して、プライマリデータベースをアップグレードまたは再構築します。

オプション	アクション
データベースのアップグレード	<p>プライマリデータベースに対してアップグレードユーティリティ (dbupgrad) を実行します。次に例を示します。</p> <pre>dbupgrad -c "UID=DBA;PWD=passwd;DBF=C:¥primary-database.db"</pre> <p>データベースがアップグレードされ、新しいトランザクションログが作成され、データベースが停止します。古いトランザクションログは削除できます。</p>
データベースの再構築 (アンロード/再ロード)	<pre>dbunload -c "UID=DBA;PWD=passwd;DBF=C:¥primary-database.db" -ar</pre>

6. アップグレードまたは再構築したデータベースを起動および停止して、データベースのトランザクションログを作成します。たとえば、次のコマンドを実行します。

```
dbping -d -c "UID=DBA;PWD=passwd;DBF=C:¥primary-database.db"
```

7. アップグレードまたは再構築されたデータベースとその新しいトランザクションログをミラーサーバとスケールアウトコピーノードにコピーします。

8. 次の順番にサーバを起動します。

- a. プライマリ
- b. ミラー
- c. コピーノード

## 結果

ミラーリングシステムのデータベースがアップグレードまたは再構築され、ミラーリングシステムが動作中になります。

## 次のステップ

データベースミラーリングシステムを検査し、アップグレードが正常に完了したことを確認すると同時に、データベースミラーリングシステムをアプリケーションでテストします。

### 1.9.1.11 データベースアップグレードのトラブルシューティング: サポートパッケージの適用時は JDBC アプリケーションが動作していないことを確認してください

サポートパッケージの適用後、JDBC アプリケーションが動作を停止する場合があります。

次のようなメッセージを受信します。The sajdbc4.jar build does not match the shared object build が含まれます。

このメッセージは、サポートパッケージの適用時、Interactive SQL または SQL Central の高速ランチャや独自の JDBC アプリケーションが実行されていると返されることがあります。こういった場合、Java VM は、ロードしたすべての DLL または共有オブジェクトはロックしていますが JAR ファイルはロックしていません。その結果、サポートパッケージを適用することによって、対応する dbjdbc17 および dbjodbc17 DLL または共有オブジェクトが更新されずに sajdbc4 と jodbc4 の JAR ファイルが更新されてしまいます。JDBC アプリケーションが再起動されたとき、JDBC JAR ファイルと対応する DLL または共有オブジェクトのビルドが一致せず、上記のメッセージが返されます。

まず、JDBC ベースのアプリケーションをすべてシャットダウンし、サポートパッケージを再適用してください。サポートパッケージの再適用で効果がない場合、次の方法で問題を解決してください。

サポートパッケージのインストーラにより DLL と共有オブジェクトが正しく更新されたことを確認

- Windows の場合、サポートパッケージの適用時に dbjdbc17.dll ファイルと dbjodbc17.dll ファイルが正しく更新されたことを確認してください。
- Linux/UNIX の場合、サポートパッケージの適用時に libdbjdbc17.so.1 および libdbjodbc17.so.1 共有オブジェクトが正しく更新されたことを確認してください。
- Mac OS X の場合、サポートパッケージの適用時に libdbjdbc17.dylib および libdbjodbc17.dylib 共有オブジェクトが正しく更新されたことを確認してください。

#### i 注記

システムには 32 ビットバージョンと 64 ビットバージョンの両方の DLL または共有オブジェクトがインストールできます。DLL と共有オブジェクトの確認の際には、データベースサーバのビット設定ではなく、JAVA VM のビット設定と一致するものを確認する必要があります。

システムに DLL と共有オブジェクトのコピーが複数ないことを確認

DLL または共有オブジェクトが正常にアップデートされている場合、同じビット設定の DLL または共有オブジェクトのコピーが複数ないことを確認する必要があります。DLL の複数のコピーは、複数のクラスローダに DLL や共有オブジェクトがロードされないようにする Java の制限をバイパスするために DLL または共有オブジェクトを Java VM の拡張フォルダにコピーした場合に発生することがあります。

JAR ファイルが正常にアップデートされていることを確認

DLL または共有オブジェクトが正常にアップデートされ、システムにコピーが重複していなければ、さまざまな JAR ファイルが正常にアップデートされたことを確認します。各 JAR ファイルを確認するには、次のコマンドを実行し、JAR ファイルによってレポートされた SQL Anywhere のバージョンおよびビルド番号が、インストールしたサポートパッケージの SQL Anywhere のバージョンおよびビルド番号と一致することを確認します。

sajdbc4.jar を確認するには、次のコマンドを実行します (path を JAR ファイルのパスに置き換えてください)。

```
java -cp path¥sajdbc4.jar sap.jdbc4.sqlanywhere.IBuildNum
```

jodbc4.jar を確認するには、次のコマンドを実行します (path を JAR ファイルのパスに置き換えてください)。

```
java -cp path¥jodbc4.jar ianywhere.ml.jdbcodbc.jdbc4.IBuildNum
```

どの JAR ファイル、DLL、または共有オブジェクトがサポートパッケージのビルド番号と一致しないか判断できたら、アプリケーションでファイルがロックされていないことを確認し、サポートパッケージを再適用します。

## 1.9.1.12 データベースアップグレードのトラブルシューティング: 集合関数と外部参照

SQL Anywhere は、サブクエリでの集合関数の使用法を明確に規定した ISO SQL 標準に準拠しています。この変更は、旧バージョン用に記述された文の動作に影響を与えます。以前は有効だったクエリでエラーメッセージが生成され、結果セットが変わる可能性があります。

集合関数がサブクエリに使用され、集合関数の参照先カラムが外部参照の場合は、集合関数全体が外部参照として扱われます。集合関数はサブクエリ内ではなく外部ブロック内で計算され、サブクエリ内の定数となります。

サブクエリに含まれる外部参照の集合関数の使用には、次の制限が適用されます。

- 外部参照の集合関数を指定できるのは、SELECT リストまたは HAVING 句にあるサブクエリの中だけです。また、これらの句は、外部ブロックの隣になくてはなりません。
- 外部参照の集合関数に指定できるのは、1つの外部カラム参照だけです。
- ローカルカラム参照と外部カラム参照を、同じ集合関数に混在させることはできません。

新しい標準に関連する問題は、ローカル参照しか含まないように集合関数を書き換えると回避することができます。たとえば、サブクエリ (SELECT MAX(S.y + R.y) FROM S) がローカルカラム参照 (S.y) と外部カラム参照 (R.y) の両方で構成されていると、不正になります。このサブクエリは (SELECT MAX(S.y) + R.y FROM S) に書き換えることができます。書き換えると、集合関数はローカルカラム参照だけを持つこととなります。外部参照の集合関数が SELECT や HAVING 以外の句に指定されている場合にも、同様の書き換えを使用できます。

### 例

#### 例 1

次のクエリの場合、SQL Anywhere 以前のバージョンでは有効な結果が生成されました。

```
SELECT Name,
       ( SELECT SUM( p.Quantity )
         FROM SalesOrderItems )
FROM Products p WHERE p.ID = 300;
```

Name	SUM(p.Quantity)
Tee shirt	30,716

SQL Anywhere 以降のバージョンでは、同じクエリを実行したときに、関数またはカラムの参照が GROUP BY 句の中になければならないというエラーメッセージが生成されます。SUM(p.Quantity) は外部参照として扱われ、外部クエリブロック内で計算されるため、この文は有効でなくなっています。上記のクエリは次のクエリと同等です。

```
SELECT Name,
       SUM( p.Quantity ) AS Z,
       ( SELECT Z
         FROM SalesOrderItems )
FROM Products p WHERE p.ID = 300;
```

ただし、集合関数は外部クエリブロック内で計算されるようになったため、外部クエリブロックはグループクエリとして扱われ、カラム名を SELECT リストの GROUP BY 句に記述する必要があります。その意味でもこのクエリは無効で、同じエラーメッセージが生成されます。SQL Anywhere 8 以降のバージョンで SQL Anywhere 7 以前のバージョンの最初のクエリと同じ結果セットを返すには、次のクエリを使用してください。

```
SELECT Name,
```

```
p.Quantity * ( SELECT COUNT( * ) FROM SalesOrderItems )
FROM Products p WHERE p.ID = 300;
```

## 例 2

SQL Anywhere 7 以前のバージョンでは、SUM(p.Quantity) はネストされた SELECT クエリブロックの内部で計算され、p.Quantity のみが外部参照として扱われたため、次のクエリでは 30,716 という結果が生成されました。

```
SELECT ( SELECT FIRST SUM( p.Quantity ) FROM SalesOrderItems ) AS ss
FROM Products p WHERE p.ID = 300;
```

SQL Anywhere 8 以降のバージョンでは、SUM(p.Quantity) は外部参照として扱われ、外部クエリブロックの内部で計算されるため、同じクエリは 28 という結果を返します。つまり、上記のクエリは次のクエリと同等です。

```
SELECT DT.ss
FROM ( SELECT SUM( p.Quantity ) AS asum,
        ( SELECT FIRST asum FROM SalesOrderItems ) AS ss
      FROM Products p WHERE p.ID = 300 ) AS DT;
```

SQL Anywhere 8 以降のバージョンで SQL Anywhere 7 以前のバージョンの最初のクエリと同じ結果を返すには、次のクエリを使用してください。

```
SELECT p.Quantity * ( SELECT COUNT( * ) FROM SalesOrderItems ) AS ss
FROM Products p WHERE p.ID = 300;
```

## 1.9.1.13 SQL Anywhere のインストール (UNIX)

コマンドラインを使用して SQL Anywhere をインストールします。

### 前提条件

インストールファイルはコンピュータにコピー済みです。

### 手順

1. UNIX シェルを開き、インストールイメージのルートに移動します。
2. 次のコマンドを実行して、ソフトウェアをインストールします。

```
./setup
```

(省略可) 使用状況を表示するには、次のコマンドを実行します。

```
./setup -h
```

3. 画面に表示される指示に従います。

## 結果

SQL Anywhere がインストールされます。

## 1.9.1.14 SQL Anywhere のインストール (Windows)

Windows で SQL Anywhere をインストールします。

### 前提条件

SQL Anywhere インストール CD があるか、インストールファイルがコンピュータにコピー済みです。

### 手順

1. インストール CD のルートに移動するか、SQL Anywhere インストール環境が含まれるディレクトリに移動し、`setup.exe` を実行します。
2. 画面に表示される指示に従います。

## 結果

SQL Anywhere がインストールされます。

## 1.9.2 Mobile Link のアップグレード

アップグレードを行う前に、影響がありそうな動作の変更を確認し、一般的なアップグレード前の対応策を実行してください。

### 既存のソフトウェアとの互換性

- バージョン 17 の Mobile Link クライアントは、バージョン 16 以降の Mobile Link サーバと互換性があります。ただし、バージョン 17 の Mobile Link サーバを使用していない場合は、機能が制限されることがあります。

- バージョン 17 の Mobile Link サーバは、バージョン 12 以降のクライアントで使用できます。これより以前のクライアントをサポートする必要がある場合は、そのバージョンのクライアントをサポートしている古いバージョンの Mobile Link サーバを使用してください。
- 記述されている動作変更が既存のアプリケーションに影響を与えないことを確認してください。影響する場合は、既存のアプリケーションを更新してください。

#### i 注記

バージョン 16 と 17 で Mobile Link サーバシステムオブジェクトの変更はないため、バージョン 16 の Mobile Link サーバを使用している場合に実行するアップグレードスクリプトはありません。

## アップグレードの順序

既存の Mobile Link インストール環境をアップグレードするには、次の順序でコンポーネントをアップグレードします。

1. Mobile Link サーバを停止します。
2. 統合データベースをアップグレードします。
3. Mobile Link サーバをアップグレードします。
4. Mobile Link サーバを起動します。
5. Mobile Link クライアントをアップグレードします。

バージョン 17 の Mobile Link サーバは、バージョン 11 以降のクライアントでのみ使用できます。バージョン 17 の Mobile Link サーバと一緒に使用するには、バージョン 11 より前の Mobile Link クライアントはアップグレードする必要があります。

このセクションの内容:

#### [統合データベースのアップグレード \[105 ページ\]](#)

新しい Mobile Link サーバと既存の統合データベースを使用できるようにするには、新しいシステムオブジェクトをインストールするアップグレードスクリプトを実行する必要があります。アップグレードスクリプトは、現在インストールされている Mobile Link システムテーブルの所有者が実行する必要があります。

#### [Mobile Link サーバのアップグレード \[113 ページ\]](#)

バージョン 17 のリモートデータベースを同期する場合、Mobile Link サーバをバージョン 17 にアップグレードしてください。

#### [SQL Anywhere Mobile Link クライアントのアップグレード \[113 ページ\]](#)

運用環境では、統合データベースと Mobile Link サーバの両方をアップグレードした後、SQL Anywhere リモートデータベースのみをアップグレードします。バージョン 16 および 17 の Mobile Link クライアントはバージョン 17 の Mobile Link サーバと同期できるため、Mobile Link サーバのバージョンが 12 以前の場合は、Mobile Link サーバのみをアップグレードする必要があります。

## 関連情報

[Mobile Link バージョン 17.0 の新機能 \[57 ページ\]](#)

[アップグレードと再構築の注意事項 \[73 ページ\]](#)



## 1.9.2.1 統合データベースのアップグレード

新しい Mobile Link サーバと既存の統合データベースを使用できるようにするには、新しいシステムオブジェクトをインストールするアップグレードスクリプトを実行する必要があります。アップグレードスクリプトは、現在インストールされている Mobile Link システムテーブルの所有者が実行する必要があります。

Mobile Link システム設定を更新するには、次の方法を使用することもできます。

- SQL Central の Mobile Link プラグインで、**Mobile Link 17** > **プロジェクト** > **統合データベース** をクリックし、データベース名を右クリックして、**Mobile Link システム設定のチェック** をクリックします。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。
- **同期モデル展開ウィザード**を使用する場合は、統合データベースに接続するとシステム設定がチェックされます。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。

6.0.x の Mobile Link アップグレードスクリプトは削除されました。このアップグレードが必要な場合は、テクニカルサポートに連絡してください。

### 注記

- ml\_add\_missing\_dnl\_scripts ストアドプロシージャを使用して、欠落している download\_cursor スクリプトか download\_delete\_cursor スクリプトまたはその両方を修正します。スクリプトバージョン名を使用してこのプロシージャを呼び出すと、指定したスクリプトバージョンで使用されるすべての同期テーブルに対して、欠落している download\_cursor スクリプトと download\_delete\_cursor スクリプトが無視されるスクリプトとして定義されます。
- 10.0.0 より以前のバージョンで作成された authenticate\_user\_hashed スクリプトを使用する場合は、お使いの RDBMS でバイナリに相当する型を使用して、VARBINARY(20) ではなく VARBINARY(32) を受け入れるようにスクリプトを変更する必要があります。

このセクションの内容:

#### [統合データベースのアップグレード \(SQL Anywhere 10.0.0 以降\) \[106 ページ\]](#)

新しい Mobile Link サーバと、バージョン 10.0.0 以降の既存の SQL Anywhere 統合データベースを使用するには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

#### [統合データベースのアップグレード \(Adaptive Server Enterprise, SAP IQ, Oracle, MySQL、または Microsoft SQL Server\) \[107 ページ\]](#)

新しい Mobile Link サーバと既存の統合データベースを使用するには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

#### [IBM DB2 LUW 統合データベースのアップグレード \[109 ページ\]](#)

新しい Mobile Link サーバと既存の IBM DB2 LUW 統合データベースを使用するには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

#### [統合データベースのアップグレード \(10.0.0 より前の SQL Anywhere\) \[110 ページ\]](#)

新しい Mobile Link サーバと、バージョン 10.0.0 より前の既存の SQL Anywhere 統合データベースを使用するには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

## 関連情報

[サポートセンターへの問い合わせ](#)

### 1.9.2.1.1 統合データベースのアップグレード (SQL Anywhere 10.0.0 以降)

新しい Mobile Link サーバと、バージョン 10.0.0 以降の既存の SQL Anywhere 統合データベースを使用するには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

#### 前提条件

設定スクリプトを実行するには、現在インストールしている Mobile Link システムテーブルの所有者になっている必要があります。

ロック/ブロック検出ロジックを呼び出すには、MONITOR システム権限が必要です。

#### コンテキスト

Mobile Link システム設定を更新するには、次の方法を使用することもできます。

- SQL Central の Mobile Link プラグインで、**Mobile Link 17 > プロジェクト > 統合データベース** をクリックし、データベース名を右クリックして、**Mobile Link システム設定のチェック** をクリックします。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。
- **同期モデル展開ウィザード** を使用する場合は、統合データベースに接続するとシステム設定がチェックされます。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。

#### 手順

1. SQL Anywhere ソフトウェアのアップグレード
2. アップグレード元のバージョン用の適切なアップグレードスクリプトを実行して、Mobile Link システム設定をアップグレードします。

アップグレードスクリプトは、`upgrade_sa.sql` という名前です。アップグレードスクリプトは、`MobiLink¥upgrade ¥version` の SQL Anywhere インストール環境にインストールされています。`version` は、アップグレード対象の SQL Anywhere バージョンを示します。

たとえば、Interactive SQL からデータベースに接続し、次の文を実行します。

```
READ "C:¥Program Files¥SQL Anywhere 17¥MobiLink¥upgrade¥10.0.x¥upgrade_sa.sql"
```

## 結果

統合データベースは、新しい Mobile Link サーバとともに使用できるようになりました。

## 関連情報

[バージョン 10 以降のデータベースのアップグレード \[71 ページ\]](#)

### 1.9.2.1.2 統合データベースのアップグレード (Adaptive Server Enterprise、SAP IQ、Oracle、MySQL、または Microsoft SQL Server)

新しい Mobile Link サーバと既存の統合データベースを使用するには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

## 前提条件

設定スクリプトを実行するには、現在インストールしている Mobile Link システムテーブルの所有者になっている必要があります。

ASE の場合、Mobile Link サーバのログイン ID には、master..systransactions に対する SELECT 権限が必要です。

SAP IQ の場合、ダウンロード用のスナップショットアイソレーションを使用するには、Mobile Link サーバで SP\_IQTRANSACTION に対する EXECUTE 権限が必要です。

Oracle の場合、Mobile Link サーバを統合データベースに接続するために使用される RDBMS ユーザは、Mobile Link システムテーブル、プロシージャなどを修飾子なしで使用できる必要があります (SELECT \* from ml\_user など)。また、RDBMS ユーザの場合は、GV\$TRANSACTION、GV\$SESSION、GV\$LOCK、DBA\_OBJECTS での SELECT 権限、および DBMS\_UTILITY での EXECUTE 権限も必要です。同義語である GV\$TRANSACTION、GV\$SESSION、GV\$LOCK には直接パーミッションを付与することができません。そのため、同義語の派生元である GV\_\$TRANSACTION、GV\_\$SESSION、および GV\_\$LOCK の動的パフォーマンスビューに対してパーミッションを付与する必要があります。このアクセス権を付与するには、SYS として接続する必要があります。

Microsoft SQL Server の場合、統合データベースへの接続のために Mobile Link サーバが使用する RDBMS ユーザは、VIEW SERVER STATE のパーミッション、SYS.DATABASES からの SELECT のパーミッション、SYS.DM\_TRAN\_LOCKS、SYS.PARTITIONS、SYS.SYSPROCESSES からの SELECT のパーミッションが必要です。

## コンテキスト

Adaptive Server Enterprise、Oracle、MySQL、または Microsoft SQL Server で構築された統合データベース内の Mobile Link システムオブジェクトをアップグレードする必要があるのは、17.0 より以前のバージョンの Mobile Link サーバを使用している場合だけです。

Mobile Link システム設定を更新するには、次の方法を使用することもできます。

- SQL Central の Mobile Link プラグインで、**Mobile Link 17 > プロジェクト > 統合データベース** をクリックし、データベース名を右クリックして、**Mobile Link システム設定のチェック** をクリックします。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。
- **同期モデル展開ウィザード** を使用する場合は、統合データベースに接続するとシステム設定がチェックされます。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。

## 手順

1. Adaptive Server Enterprise の場合は、SELECT INTO データベースオプションを設定します。Interactive SQL で次の文を実行します。

```
USE master
go
sp_dboption your-database-name, "SELECT INTO", true
go
USE your-database-name
go
checkpoint
go
```

2. アップグレードするデータベースのバージョンに対応したアップグレードスクリプトを実行します。

アップグレードスクリプトは `upgrade_XXX.sql` という名前です。XXX は、統合データベースの RDBMS を示します。アップグレードスクリプトは、`MobiLink¥upgrade¥version` の SQL Anywhere インストール環境にインストールされています。version は、アップグレード対象の Mobile Link バージョンを示します。

たとえば、バージョン 9.0.2 の Mobile Link システムテーブルが適用された Microsoft SQL Server データベースをアップグレードするには、次のコマンドを実行します。

```
osql -S server_name -U user_name -P password -i
"C:¥Program Files¥SQL Anywhere 17¥MobiLink¥upgrade¥9.0.2¥upgrade_mss.sql"
```

## 結果

統合データベースは、新しい Mobile Link サーバとともに使用できるようになりました。

## 1.9.2.1.3 IBM DB2 LUW 統合データベースのアップグレード

新しい Mobile Link サーバと既存の IBM DB2 LUW 統合データベースを使用するには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

### 前提条件

設定スクリプトを実行するには、現在インストールしている Mobile Link システムテーブルの所有者になっている必要があります。

DB2 LUW 10.1 の場合、ロック/ブロック検出ロジックを呼び出すには、Mobile Link サーバで SYSIBMADM.MON\_LOCKWAITS に対する SELECT のパーミッションおよび SNAPSHOT\_APPL\_INFO に対する EXECUTE のパーミッションが必要です。

DB2 LUW 9 の場合、ロック/ブロック検出ロジックを呼び出すには、Mobile Link サーバで SYSIBMADM.LOCKWAITS に対する SELECT のパーミッションおよび SNAPSHOT\_APPL\_INFO に対する EXECUTE のパーミッションが必要です。

### コンテキスト

IBM DB2 LUW の統合データベースをアップグレードする必要があるのは、旧バージョンの Mobile Link サーバが 17.0 より前のバージョンである場合だけです。

Mobile Link システム設定を更新するには、次の方法を使用することもできます。

- SQL Central の Mobile Link プラグインで、**Mobile Link 17** > **プロジェクト** > **統合データベース** をクリックし、データベース名を右クリックして、**Mobile Link システム設定のチェック** をクリックします。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。
- **同期モデル展開ウィザード** を使用する場合は、統合データベースに接続するとシステム設定がチェックされます。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。

### 手順

1. IBM DB2 LUW アップグレードスクリプトのロケーションを検索します。

アップグレードスクリプトは `upgrade_db2.sql` という名前で、SQL Anywhere インストール環境の `MobiLink/upgrade/version` サブディレクトリにあります。`version` ディレクトリは、アップグレード対象の Mobile Link のバージョンを示します。

2. `upgrade_db2.sql` をコピーし、このコピーを変更します。スクリプトの先頭にある `CONNECT` 文を変更して、接続するインスタンスで動作できるようにします。コピーした SQL スクリプトを統合データベースに適用します。

## 結果

統合データベースは、新しい Mobile Link サーバとともに使用できるようになりました。

### 1.9.2.1.4 統合データベースのアップグレード (10.0.0 より前の SQL Anywhere)

新しい Mobile Link サーバと、バージョン 10.0.0 より前の既存の SQL Anywhere 統合データベースを使用するには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

#### 前提条件

SQL Anywhere 統合データベースは設定済みで、同期はまだ一度も実行していない場合は、アップグレードスクリプトではなく、設定スクリプトを実行する必要があります。この手順は、SQL Anywhere 統合データベースだけに適用されます。

#### コンテキスト

- SQL Anywhere より前のバージョンでは、Mobile Link システムテーブルの所有者は dbo でした。SQL Anywhere データベースの設定スクリプトを実行するには、Mobile Link システムテーブルの所有者として統合データベースにログインする必要があります。テーブルの変更パーミッションを持つユーザとしてこのスクリプトを実行しても十分ではありません。アップグレードスクリプトを実行するには、SQL 文 SETUSER を使用して dbo を同一化する方法もあります。次に例を示します。

```
SETUSER "dbo";
```

SQL Central で統合データベースをアップグレードするには、GRANT CONNECT 文を使用して dbo のパスワードを作成してから、dbo として接続します。次に例を示します。

```
GRANT CONNECT TO dbo IDENTIFIED BY password;
```

この場合、アップグレード後に、ALTER USER を使用して dbo のパスワードを削除します。次に例を示します。

```
ALTER USER TO dbo IDENTIFIED BY "";
```

#### 手順

- バージョン 10.0.0 より前の SQL Anywhere の統合データベースをアップグレードする場合は、先にデータベースをバージョン 17 にアップグレードする必要があります。

- a. データベースサーバを停止します。
- b. データベースをバージョン 17 にアップグレードします。
- c. DBA としてログインした状態で、データベースサーバを起動します。

アップグレードするために DBA としてログインします。

2. アップグレードするデータベースのバージョンに対応したアップグレードスクリプトを実行します。

アップグレードスクリプトは、upgrade\_asa.sql という名前です。アップグレードスクリプトは、MobiLink¥upgrade ¥version の SQL Anywhere インストール環境にインストールされています。version は、アップグレード対象の SQL Anywhere バージョンを示します。

アップグレードスクリプトを実行するには、SETUSER SQL 文を使用して dbo ユーザに同一化します。

たとえば、SQL Anywhere バージョン 9.0.2 の統合データベースをアップグレードする場合は、Interactive SQL でデータベースに接続し、次の文を実行します。

```
SETUSER "dbo";
READ 'C:¥Program Files¥SQL Anywhere 17¥MobiLink¥upgrade¥9.0.2¥upgrade_asa.sql'
```

3. dbo のパスワードを削除します。次に例を示します。

```
GRANT CONNECT TO "dbo";
```

4. DBA 以外のユーザとして Mobile Link サーバを実行している場合は、新しい Mobile Link システムオブジェクトの EXECUTE パーミッションをこのユーザに付与してください。新しいシステムオブジェクトは、アップグレード対象のバージョンによって異なります。次のコードで、すべての Mobile Link システムオブジェクトへの必要なパーミッションが付与されます。このコードを実行する前に、ユーザ名 my\_user を、Mobile Link サーバを実行しているユーザの名前に変更してください。

```
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_column to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_connection_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_database to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_device to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_device_address to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_listening to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_repair to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_status to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_primary_server to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_delivery to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_delivery_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_global_props to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_notifications to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_props to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_props_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_history to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_history_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_deployed_task to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_event to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_event_staging to my_user;
```

```

GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_managed_remote to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_notify to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_remote_db_class to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_command to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_command_property to
my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_script_version to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_scripts_modified to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_sis_sync_state to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_subscription to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_table to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_table_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_user to my_user;
GRANT EXECUTE ON dbo.ml_add_column to my_user;
GRANT EXECUTE ON dbo.ml_add_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_dnet_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_dnet_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_java_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_java_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_conn_script_chk to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_table_script_chk to my_user;
GRANT EXECUTE ON dbo.ml_add_missing_dnlD_scripts;
GRANT EXECUTE ON dbo.ml_add_passthrough to my_user;
GRANT EXECUTE ON dbo.ml_add_passthrough_repair to my_user;
GRANT EXECUTE ON dbo.ml_add_passthrough_script to my_user;
GRANT EXECUTE ON dbo.ml_add_property to my_user;
GRANT EXECUTE ON dbo.ml_add_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_user to my_user;
GRANT EXECUTE ON dbo.ml_delete_device to my_user;
GRANT EXECUTE ON dbo.ml_delete_device_address to my_user;
GRANT EXECUTE ON dbo.ml_delete_listening to my_user;
GRANT EXECUTE ON dbo.ml_delete_passthrough to my_user;
GRANT EXECUTE ON dbo.ml_delete_passthrough_repair to my_user;
GRANT EXECUTE ON dbo.ml_delete_passthrough_script to my_user;
GRANT EXECUTE ON dbo.ml_delete_remote_id to my_user;
GRANT EXECUTE ON dbo.ml_delete_sync_state to my_user;
GRANT EXECUTE ON dbo.ml_delete_sync_state_before to my_user;
GRANT EXECUTE ON dbo.ml_delete_user to my_user;
GRANT EXECUTE ON dbo.ml_delete_user_state to my_user;
GRANT EXECUTE ON dbo.ml_lock rid to my_user;
GRANT EXECUTE ON dbo.ml_qa_add_delivery to my_user;
GRANT EXECUTE ON dbo.ml_qa_add_message to my_user;
GRANT EXECUTE ON dbo.ml_qa_handle_error to my_user;
GRANT EXECUTE ON dbo.ml_qa_stage_status_from_client to my_user;
GRANT EXECUTE ON dbo.ml_qa_staged_status_for_client to my_user;
GRANT EXECUTE ON dbo.ml_qa_upsert_global_prop to my_user;
GRANT EXECUTE ON dbo.ml_ra_add_agent_id to my_user;
GRANT EXECUTE ON dbo.ml_ra_assign_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_cancel_notification to my_user;
GRANT EXECUTE ON dbo.ml_ra_cancel_task_instance to my_user;
GRANT EXECUTE ON dbo.ml_ra_clone_agent_properties to my_user;
GRANT EXECUTE ON dbo.ml_ra_delete_agent_id to my_user;
GRANT EXECUTE ON dbo.ml_ra_delete_events_before to my_user;
GRANT EXECUTE ON dbo.ml_ra_delete_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_agent_events to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_agent_ids to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_agent_properties to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_latest_event_id to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_orphan_taskdbs to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_remote_ids to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_task_results to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_task_status to my_user;
GRANT EXECUTE ON dbo.ml_ra_int_cancel_notification to my_user;

```



```
GRANT EXECUTE ON dbo.ml_ra_int_move_events to my_user;
GRANT EXECUTE ON dbo.ml_ra_manage_remote_db;
GRANT EXECUTE ON dbo.ml_ra_notify_agent_sync to my_user;
GRANT EXECUTE ON dbo.ml_ra_notify_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_reassign_taskdb to my_user;
GRANT EXECUTE ON dbo.ml_ra_set_agent_property to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_agent_auth_file_xfer to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_ack to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_prop to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_remote_dbs to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_task2;
GRANT EXECUTE ON dbo.ml_ra_ss_download_task_cmd to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_end_upload to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_upload_prop to my_user;
GRANT EXECUTE ON dbo.ml_ra_unmanage_remote_id to my_user;
GRANT EXECUTE ON dbo.ml_reset_sync_state to my_user;
GRANT EXECUTE ON dbo.ml_set_device to my_user;
GRANT EXECUTE ON dbo.ml_set_device_address to my_user;
GRANT EXECUTE ON dbo.ml_set_listening to my_user;
GRANT EXECUTE ON dbo.ml_set_sis_sync_state to my_user;
GRANT EXECUTE ON dbo.ml_upload_update_device_address to my_user;
GRANT EXECUTE ON dbo.ml_upload_update_listening to my_user;
```

## 結果

統合データベースは、新しい Mobile Link サーバとともに使用できるようになりました。

## 関連情報

[バージョン 9 以前のデータベースの再構築手順 \[80 ページ\]](#)

### 1.9.2.2 Mobile Link サーバのアップグレード

バージョン 17 のリモートデータベースを同期する場合、Mobile Link サーバをバージョン 17 にアップグレードしてください。

バージョン 17 の Mobile Link サーバを使用する前に、影響がありそうな動作の変更を確認してください。

Mobile Link バージョン 17 サーバでサポートされるのは、バージョン 12 以降の SQL Anywhere クライアントと Ultra Light クライアントだけです。これより前のクライアントをサポートする必要がある場合は、そのバージョンのクライアントをサポートしている古いバージョンの Mobile Link サーバを使用してください。

### 1.9.2.3 SQL Anywhere Mobile Link クライアントのアップグレード

運用環境では、統合データベースと Mobile Link サーバの両方をアップグレードした後、SQL Anywhere リモートデータベースのみをアップグレードします。バージョン 16 および 17 の Mobile Link クライアントはバージョン 17 の Mobile Link サーバ

と同期できるため、Mobile Link サーバのバージョンが 12 以前の場合は、Mobile Link サーバのみをアップグレードする必要があります。

検討すべきアップグレードは複数あります。

- ソフトウェアのアップグレード
- リモートデータベース自体のアップグレード
- アプリケーション全体のアップグレード

## ソフトウェアのアップグレード

dbmlsync と SQL Anywhere データベースサーバは同時にアップグレードします。バージョン 17 の Mobile Link クライアントはバージョン 17 データベースサーバで動作するバージョン 17 データベースのみ同期できます。

バージョン 17 の Mobile Link クライアントでは、同期を実行するためにバージョン 16 以降の Mobile Link サーバが必要です。バージョン 17 の Mobile Link クライアントは、バージョン 16 より前の Mobile Link サーバとは同期しません。

## リモートの SQL Anywhere のアップグレード

Mobile Link SQL Anywhere リモートデータベースをアップグレードするには、-ar オプションを設定してアンロードユーティリティ (dbunload) を実行します。

スキーマの変更やデータベースでのその他の重要な変更などがある場合は、手動でアンロードと再ロードを行う必要があります。

## アプリケーションのアップグレード

Mobile Link アプリケーションの新しいバージョンを配備する場合、同期スクリプトには新しいスクリプトバージョンを使用します。たとえば、既存のアプリケーションで v1 というスクリプトのバージョンを使用している場合は、アップグレードしたアプリケーションでは v2 という名前のスクリプトバージョンを使用します。両方のスクリプトバージョンを同時に使用することができるため、リモートデータベースを一度にではなく、段階的にアップグレードすることが簡単にできます。

バージョン 9.0.0 以降では、Mobile Link サーバの -zd オプションは削除されています。配備環境で -zd オプションを使用している場合、アップグレードを行うには、最後のダウンロードタイムスタンプが最初のパラメータとして採用されるようダウンロードスクリプトを変更します。また、クライアントをアップグレードして名前付きパラメータの使用を開始することで、スクリプトパラメータを任意の順序で配置できます。

このセクションの内容:

### [リモートの SQL Anywhere データベースの手動でのアンロードおよび再ロード \[115 ページ\]](#)

スキーマの変更やデータベースでのその他の重要な変更などがある場合は、手動でアンロードと再ロードを行う必要があります。

## 関連情報

[SQL Anywhere のアップグレード \[70 ページ\]](#)

### 1.9.2.3.1 リモートの SQL Anywhere データベースの手動でのアンロードおよび再ロード

スキーマの変更やデータベースでのその他の重要な変更などがある場合は、手動でアンロードと再ロードを行う必要があります。

#### 前提条件

バージョン 16 以降のデータベースでは、次のシステム権限が必要です。

- BACKUP DATABASE
- VALIDATE ANY OBJECT
- SERVER OPERATOR
- SELECT ANY TABLE

ソフトウェアをアップグレードする場合の一般的な対応策を行います。

#### 手順

1. すべてのデータベースアクティビティを停止します。
2. 同期が正常に完了してから、リモートデータベースを検証し、バックアップします。
3. dbtran ユーティリティを実行してデータベースのトランザクションログの開始オフセットと終了オフセットを表示します。終了オフセットをメモしてください。
4. アンロードプロセス中に変更されないように、トランザクションログの名前を変更しておきます。名前変更したログファイルを、オフラインディレクトリなどの安全なロケーションに移動します。
5. データベースをアンロードします。データベースを別のデータベースに自動的に再ロードする dbunload -a スイッチは使用しないでください。
6. 新しいデータベースを初期化します。
7. dbisql を使用してデータを新しいデータベースに再ロードし、dbunload によって生成された reload.sql ファイルを読み込みます。
8. 新しいデータベースを停止します。
9. 新しいデータベースのトランザクションログを消去します。
10. 次のオプションを使用して、新しいデータベースで dblog を実行します。
  - -z を使用して、メモした終了オフセットを指定します。

- `-x` を使用して、相対オフセットをゼロに設定します。

例:

```
dblog -x 0 -z 137829 database-name.db
```

11. `dbmsync` を起動し、移動した元のログファイルの場所を指定します。
12. 古いトランザクションログファイルが不要な場合は、データベースオプション `delete_old_logs` を設定します。

## 結果

リモートの SQL Anywhere データベースがアンロードされ、再ロードされます。

## 関連情報

[アップグレードと再構築の注意事項 \[73 ページ\]](#)

## 1.9.3 Ultra Light のアップグレード

このバージョンのソフトウェアで既存の Ultra Light アプリケーションを使用する前に、新機能と動作の変更のリストを確認して、アプリケーションに影響がないかどうかを確認してください。

Ultra Light バージョン 17 データベースへのアップグレードの前に、次の点に留意してください。

- Ultra Light バージョン 17 は、以前のバージョンのソフトウェアを使用して作成された Ultra Light データベースを読み込むことができません。
- デバイス上でデータベースをアップグレードすることはできません。
- データベースをアップグレードすると、以前のバージョンのアプリケーション、ユーティリティ、ソフトウェアからはそのデータベースに接続できなくなります。
- SQL Anywhere の複数のバージョンが同じコンピュータにインストールされている場合は、正しいユーティリティを使用するため、システムのパスに注意してください。Ultra Light ユーティリティの名前はバージョン間で共通で、ディレクトリのパス名のみが異なります。

### バージョン 16 以前のデータベースのアップグレード

バージョン 16 以前の Ultra Light のデータベースは、データベースの作成に使用したバージョンのソフトウェアを使用して、SQL または XML ファイルにアンロードする必要があります。ソフトウェアに付属している `ulunload` ユーティリティを使用してください。

古いバージョンのソフトウェアがインストールされていなくて、Windows デスクトップオペレーティングシステムを使用している場合、SQL Anywhere バージョン 17 に組み込まれているいずれかの古い `ulunload` ユーティリティを使用できます。使用している Ultra Light データベースのバージョンに合った `ulunload` ユーティリティを選択してください。

データベースのアンロードが完了したら、バージョン 17 の ulload ユーティリティを使用して XML ファイルをバージョン 17 のデータベースにロードします。データベースを SQL ファイルにアンロードした場合は、dbisql ユーティリティを使用します。

新しいデータベースを作成し、ulload ユーティリティを使用してそれを移植するもう 1 つの方法として、新しいデータベースを作成し、それを同期させる方法があります。

## 既存のソフトウェアとの互換性

- Ultra Light 17 のデータベースファイルは、バージョン 17 のクライアントアプリケーション、またはバージョン 17 の Ultra Light エンジンからの接続のみをサポートします。
- Ultra Light バージョン 17 ランタイムと Ultra Light バージョン 17 エンジンは、バージョン 16 以前の Ultra Light で作成したデータベースファイルでは使用できません。バージョン 16 以前で作成したアプリケーションコードを使用するには、それを再コンパイルします。

このセクションの内容:

### [Ultra Light データベースのアップグレード \(Windows\) \[117 ページ\]](#)

旧バージョンのソフトウェアからアップグレードする場合は、Windows 上でバージョン 17 の Ultra Light データベースを再構築します。

### [Ultra Light データベースのアップグレード \(Linux\) \[119 ページ\]](#)

旧バージョンのソフトウェアからアップグレードする場合は、Linux 上でバージョン 17 の Ultra Light データベースを再構築します。

## 関連情報

### [Ultra Light バージョン 17.0 の新機能 \[61 ページ\]](#)

[複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 \[74 ページ\]](#)

## 1.9.3.1 Ultra Light データベースのアップグレード (Windows)

旧バージョンのソフトウェアからアップグレードする場合は、Windows 上でバージョン 17 の Ultra Light データベースを再構築します。

### 前提条件

- Ultra Light バージョン 9 以前のデータベースからアップグレードする場合は、そのバージョンの SQL Anywhere がインストールされていなければなりません。

- 既存の Ultra Light データベースのバックアップコピーを作成してください。
- 同期されていない変更が含まれている可能性がある運用データベースの場合は、データベースを同期します。

## コンテキスト

Ultra Light バージョン 17 は、以前のバージョンのソフトウェアを使用して作成された Ultra Light データベースを読み込むことができません。

## 手順

1. データベースの内容で XML ファイル (複数可) を作成します。

コマンドプロンプトを開き、以前のバージョンの SQL Anywhere がインストールされたディレクトリに移動し、データベースに対して `ulunload` ユーティリティを使用してください。

Ultra Light データベースがバージョン 11 または 12 の SQL Anywhere で作成されている場合、コマンドプロンプトを開き、`%SQLANY17%\UltraLite\Unload\` ディレクトリに移動し、使用しているバージョンの Ultra Light データベースに適用されるディレクトリ (v11、v12 など) を選択し、そのディレクトリから `ulunload` ユーティリティを使用するという方法もあります。

2. Ultra Light バージョン 17 `ulload` ユーティリティを使用するか、SQL Central の *Ultra Light 17* プラグインで **データベースロードウィザード** を使用して、スキーマとデータを新しいバージョン 17 のデータベースにロードします。

空のデータベースを作成し、`ulload` ユーティリティを使用して XML をロードする場合は、データベースの照合オプションとエンコーディングオプションを確認します。Ultra Light データベースはデフォルトで UTF-8 エンコーディングになり、`ulload` の実行時には、新しいデータベースで既存のデータベースのエンコーディングが使用されます。

3. 生成された XML ファイルをチェックして、UTF-8 エンコーディングの設定を確認します。

## 結果

Ultra Light データベースが最新バージョンにアップグレードされます。

## 1.9.3.2 Ultra Light データベースのアップグレード (Linux)

旧バージョンのソフトウェアからアップグレードする場合は、Linux 上でバージョン 17 の Ultra Light データベースを再構築します。

### 前提条件

- データベースを作成したバージョンの SQL Anywhere をインストールしてください。
- 既存の Ultra Light データベースのバックアップコピーを作成してください。
- 同期されていない変更が含まれている可能性がある運用データベースの場合は、データベースを同期します。

### コンテキスト

Ultra Light バージョン 17 は、以前のバージョンのソフトウェアを使用して作成された Ultra Light データベースを読み込むことができません。

### 手順

1. データベースの内容で XML ファイル (複数可) を作成します。

コマンドプロンプトで、旧バージョンの SQL Anywhere がインストールされている場所に移動し、データベースに対して `ulunload` ユーティリティを使用してください。

2. Ultra Light バージョン 17 `ulload` ユーティリティを使用するか、SQL Central の *Ultra Light 17* プラグインで **データベースロードウィザード** を使用して、スキーマとデータを新しいバージョン 17 のデータベースにロードします。

Ultra Light データベースのデフォルトは UTF8 エンコーディングです。このエンコードがニーズに適合しない場合は、`utf8_encoding` パラメータを明示的にオフに設定します。

3. 生成された XML ファイルをチェックして、UTF-8 エンコーディングの設定を確認します。

### 結果

Ultra Light データベースが最新バージョンにアップグレードされます。

## 1.9.4 SQL Remote のアップグレード

SQL Remote のアップグレードは一度に 1 サイトずつ実行でき、アップグレード時には、最新バージョンでサポートされていない、旧バージョンの SQL Remote で使用可能だった機能に対処する必要があります。

SQL Remote のアップグレード要件は、次のとおりです。

ソフトウェアアップグレードは一度に 1 サイトずつ実行できる

旧バージョンの Message Agent (dbremote) は、バージョン 17 の Message Agent とメッセージを交換することができます。バージョン 5 の SQL Remote では、compression データベースオプションの値が -1 に設定されている場合、バージョン 5 の Message Agent とバージョン 17 の Message Agent でメッセージを交換できます。インストール環境全体のソフトウェアを同時にアップグレードする必要はありません。

データベースのアップグレード

SQL Anywhere バージョン 9 以前を使用していたリモートデータベースまたは統合データベースをアップグレードする場合は、データベースをアンロードして再ロードすることでデータベースファイルフォーマットをアップグレードする必要があります。すべてのデータベースを同時にアップグレードする必要はありません。

**Adaptive Server Enterprise 統合データベースのアップグレード**

SQL Remote では、Adaptive Server Enterprise 統合データベースはサポートされなくなりました。Adaptive Server Enterprise データベースを同期するには、Mobile Link を使用します。

**SQL Remote のアップグレード**

バージョン 11.0.0 では、SQL Remote の VIM と MAPI の各メッセージシステムがサポートされなくなりました。VIM または MAPI を使用するデータベースを SQL Anywhere バージョン 17 にアップグレードする場合は、メッセージタイプを File、FTP、または SMTP に変更してください。メッセージタイプが MAPI または VIM の場合、dbremote.exe は起動しません。

### 例

以下に、バージョン 5 の SQL Remote のアップグレードの一例を示します。

1. 統合データベースサーバと SQL Remote Message Agent をアップグレードしてから、統合データベースをアンロードして再ロードすることでデータベースファイルをアップグレードします。すべてのメッセージがリモートサイトのバージョン 5 のソフトウェアと互換性を持つように、compression データベースオプションを -1 に設定してください。
2. リモートデータベースサーバと Message Agent を 1 つずつアップグレードしてから、リモートデータベースをアンロードして再ロードすることでデータベースファイルフォーマットをアップグレードします。compression データベースオプションを -1 以外の値に設定すると、統合データベースサーバに送信されるメッセージに対して圧縮機能とコード化機能を利用できます。
3. すべてのリモートデータベースサーバと Message Agent をアップグレードし終わったら、統合サイトの compression データベースオプションを -1 以外の値に設定します。

## 関連情報

[バージョン 9 以前のデータベースの再構築手順 \[80 ページ\]](#)



## 1.9.5 モニタのアップグレードおよびリソースの移行

モニタをアップグレードするには、新しいソフトウェアをインストールし、リソースを移行します。メジャーリリース間でモニタをアップグレードするときは、ソフトウェアインストーラを使用し、リソース移行のプロンプトに従います。メジャーリリースへアップグレードするときは、ソフトウェアをインストールし、移行ユーティリティを実行してリソースを移行します。

### 前提条件

不要な警告を回避するため、サーバを停止する前に、リソースのモニタでブラックアウト期間をスケジュールしてください。

### コンテキスト

コンピュータ上で一度に実行できるモニタのバージョンは1つだけです。

バージョン 12、16、17 モニタのデータベースファイルのデフォルトのロケーションを次の表に示します。

#### モニタ Production Edition

オペレーティングシステム	バージョン 12 と 16 のディレクトリ	バージョン 17 のディレクトリ
Windows 7 以降	<code>C:¥Users¥Public¥Documents¥SQL Anywhere Monitor version-number¥samonitor.db</code>	<code>C:¥Users¥Public¥Documents¥SQL Anywhere Monitor 17 ¥samonitor.db</code>
Linux	<code>/opt/samonitorversion-number/samonitor.db</code>	<code>/opt/samonitor17/samonitor.db</code>

#### モニタ Developer Edition

オペレーティングシステム	バージョン 12 と 16 のディレクトリ	バージョン 17 のディレクトリ
Windows 7 以降	<code>C:¥Users¥Public¥Documents¥SQL Anywhereversion-number ¥Monitor¥samonitor.db</code>	<code>C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Monitor ¥samonitor.db</code>
	<code>/opt/sqlanywhereversion-number/samonitor.db</code>	<code>/opt/sqlanywhere17/samonitor.db</code>

### 手順

1. 既存のモニタのデータベースファイル `samonitor.db` のバックアップコピーを作成します。

2. 新しいモニタをインストールします。インストールメディアの `Monitor` ディレクトリにある `setup.exe` ファイルを実行し、表示される指示に従います。インストールが完了したら、新しいモニタを停止します (稼働している場合)。
3. コマンドプロンプトで、次のオプションを使用して、移行ユーティリティ (`run_migrator`) を実行します。

#### **-t temporary-directory**

テンポラリファイルのディレクトリを指定します。デフォルトでは、テンポラリファイルは、移行ユーティリティが存在するディレクトリに作成されます。

#### **i 注記**

モニタ移行で作成されるテンポラリファイルは、移行プロセスの最後で削除されます。-t オプションを使用して、これらのテンポラリファイルのディレクトリを指定します。テンポラリファイルには、古いバージョンのモニタデータベースファイルと同等の領域が必要です。指定のディレクトリに十分な領域があることを確認してください。

#### **source-filename**

古いモニタのデータベースファイルにパスとファイル名を指定します。たとえば、バージョン 16 の `samonitor.db` ファイルのパスを指定します。

#### **destination-filename**

リソースと設定がロードされる新しいモニタのファイルにパスとファイル名を指定します。たとえば、バージョン 17 の `samonitor.db` ファイルのパスを指定します。

次に例を示します。

```
C:¥Program Files¥SQL Anywhere 17¥Bin64¥run_migrator.cmd -t c:¥monitorbackup C:¥Users¥Public¥Documents¥SQL Anywhere 16¥Monitor¥samonitor.db C:¥Users¥Public¥Documents¥SQL Anywhere 17¥Monitor¥samonitor.db
```

次の表を使用して、移行ユーティリティ (`run_migrator`) のロケーションを確認します。

オペレーティングシステム	モニタタイプ
Windows	<code>C:¥Program Files¥SQL Anywhere 17¥Bin??¥run_migrator.cmd</code>
Linux	<code>/opt/sqlanywhere17/bin??/run_migrator.sh</code>

## 結果

モニタがアップグレードされます。

## 次のステップ

モニタを起動します。

## 1.9.6 バージョン 16 のデータベースのアップグレードによる SQL Anywhere コックピット の使用

ユーザがコックピットに接続するために必要な COCKPIT\_ROLE ユーザ定義ロールを作成し、ユーザに行使権限を付与します。

### 前提条件

新しいロールを作成し、ユーザに付与するには、MANAGE ROLES システム権限が必要です。

### コンテキスト

コックピットはデータベースサーバをモニタし、それに伴い、サーバで実行中のすべてのデータベースに関する情報を取得して表示します。コックピットに接続するには、サーバで実行中のいずれかのデータベースのユーザのユーザ ID とパスワードが必要です。ユーザには、COCKPIT\_ROLE ユーザ定義ロールの行使権限も必要です。デフォルトでは、このロールはバージョン 16 データベースで定義されていません。したがって COCKPIT\_ROLE ユーザ定義ロールを作成し、その行使権をユーザに付与する必要があります。

コックピットは従来の定義者セキュリティモデルを使用するデータベースをサポートしません。データベースがこのセキュリティモデルを使用している場合、たとえユーザ定義 COCKPIT\_ROLE ロールを有していても、コックピットに接続することはできません。

### 手順

1. バージョン 17 のデータベースサーバでバージョン 16 のデータベースを起動します。
2. Interactive SQL を起動して、データベースに接続します。
3. 次の文を実行して、COCKPIT\_ROLE ユーザ定義ロールを作成します。

```
CREATE ROLE COCKPIT_ROLE;  
GRANT MONITOR, DROP CONNECTION, BACKUP DATABASE, SERVER OPERATOR TO COCKPIT_ROLE;
```

COCKPIT\_ROLE が作成されます。MANAGE ROLES システム権限には、この新しいロールに対する管理権限 (のみ) が自動的に付与されます。このロールは、MANAGE ROLES システム権限を持つユーザなら誰でも管理できます。

4. コックピットを使用する必要があるユーザに COCKPIT\_ROLE の行使権限を付与します。

たとえば、COCKPIT\_ROLE をユーザ JohnDoe に付与するには、次の文を実行します。

```
GRANT ROLE COCKPIT_ROLE TO JohnDoe;
```

5. バージョン 16 のデータベースを実行中のデータベースサーバでコックピットを開始します。

## 1.9.7 バージョン 17 のデータベースのアップグレードによる SQL Anywhere コックピット の使用

バージョン 17 のユーザに ACCESS DISK INFORMATION システム権限を付与して、バージョン 17 コックピット内のすべての機能を使用できるようにします。

### 前提条件

COCKPIT\_ROLE ユーザ定義ロールにロールを追加するには、MANAGE ROLES システム権限が必要です。

### コンテキスト

コックピットはデータベースサーバをモニタし、それに伴い、サーバで実行中のすべてのデータベースに関する情報を取得して表示します。コックピットに接続するには、サーバで実行中のいずれかのデータベースのユーザのユーザ ID とパスワードが必要です。ユーザには、COCKPIT\_ROLE ユーザ定義ロールの行使権限も必要です。

コックピットのすべての機能にアクセスするには、次のシステム権限も必要です。

- MONITOR システム権限
- DROP CONNECTION システム権限
- BACKUP DATABASE システム権限
- SERVER OPERATOR システム権限
- ACCESS DISK INFORMATION システム権限

バージョン 17.0 のビルド 2000 以降の場合、COCKPIT\_ROLE ユーザ定義ロールはこれらのシステム権限にデフォルトで割り当てられています。以前のバージョンのソフトウェアを使用する場合、COCKPIT\_ROLE ユーザ定義ロールに ACCESS DISK INFORMATION システム権限が欠けている場合があります。このシステム権限は、ディスク容量不足のアラートに関する情報を表示するために必要です。

ユーザがコックピットのすべての機能を使用できるようにするには、ACCESS DISK INFORMATION システム権限に対して行使権限を付与してください。

### 手順

1. バージョン 17 のデータベースサーバでバージョン 17 のデータベースを起動します。
2. Interactive SQL を起動して、データベースに接続します。
3. ユーザに ACCESS DISK INFORMATION システム権限を付与します。

オプション	アクション
COCKPIT_ROLE ユーザ定義ロールに対して ACCESS DISK INFORMATION システム権限を追加し、このロールを持つすべてのユーザに ACCESS DISK INFORMATION システム権限を与えます。	次を実行します。  <pre>GRANT ACCESS DISK INFORMATION to COCKPIT_ROLE;</pre>
個々のコックピットユーザに ACCESS DISK INFORMATION システム権限を付与します。	例:  <pre>GRANT ACCESS DISK INFORMATION to userA;</pre>

## 1.9.8 バージョン 16 の OData サーバの変換

SQL 文を使用して ODATA PRODUCER 定義を作成し、dbsrv17 -xs odata プロトコルオプションを使用して、有効化および定義済の ODATA PRODUCER 文を開始します。

### 手順

1. -xs odata オプションを使用してデータベースサーバを実行し、バージョン 16 の Embedded HTTP サーバオプションを OData プロトコルのネットワークプロトコルオプションとして渡し、OData サーバを起動します。

たとえば、次のオプションはバージョン 16 の OData サーバの設定ファイルで定義されているとします。

```
# Embedded HTTP server options
# -----
LogFile = ../../odata.log
LogVerbosity = 1
ServerPort = 8000
ShutdownListenerPort = 8083
SSLKeyStore = ../../samplekeystore.jks
SSLKeyStorePassword = pwd
Producers = OrderEntryProducer, StaffingProducer
```

次のコマンドを実行して、動作を複製します。

```
dbsrv17 -xs odata(SecureServerPort=8000;SSLKeyStore=../../samplekeystore.jks;ServerPort=0;SSLKeyStorePassword=pwd;LogFile=../../odata.log;LogVerbosity=1)
```

### i 注記

SecureOnly プロトコルによる HTTP 受信がオフになっていない限り、キーストア情報が提供されたときに OData サーバは HTTP ポートと HTTPS ポートの両方で受信します。

*Producers* オプションと *ShutdownListenerPort* オプションは不要になります。指定した *Producers* は CREATE ODATA PRODUCER ENABLED 句によって有効になっています。

2. Interactive SQL を起動して、データベースに接続します。
3. バージョン 16 の OData Server 設定ファイルで CREATE ODATA PRODUCER 文によって定義されている OData Producers を複製します。

たとえば、次のオプションはバージョン 16 の OData サーバの設定ファイルで定義されているとします。

```
# Shared OData Producer options
# -----
Authentication = none
ConnectionPoolMaximum = 10
DbProduct = sqlanywhere
PageSize = 100
ReadOnly = false
# OrderEntryProducer options
# -----
OrderEntryProducer.Model = ../../ordermodel.osdl
OrderEntryProducer.ModelConnectionString =
uid=dba;pwd=sql;ServerName=orderentry;dbf=orderentry.db
OrderEntryProducer.ServiceRoot = /orders/
OrderEntryProducer.DbConnectionString =
uid=dba;pwd=sql;ServerName=orderentry;dbf=orderentry.db
```

以下の文を作成して、そのプロパティを複製する OData Producer を定義します。

```
CREATE ODATA PRODUCER OrderEntryProducer
ENABLED
MODEL FILE '../../ordermodel.osdl'
SERVICE ROOT '/orders/'
USING 'ConnectionPoolMaximum=10;PageSize=100;ReadOnly=false;
```

*ModelConnectionString* オプション、*DbProduct* オプション、*DbConnectionString* オプションは不要になります。

## 1.10 このマニュアルの印刷、再生、および再配布

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

1. ここに示したものとそれ以外のすべての著作権と商標の表示をすべてのコピーに含めること。
2. マニュアルに変更を加えないこと。
3. SAP 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

ここに記載された情報は事前の通知なしに変更されることがあります。

# 重要免責事項および法的情報

## コードサンプル

この文書に含まれるソフトウェアコード及び / 又はコードライン / 文字列 (「コード」) はすべてサンプルとしてのみ提供されるものであり、本稼動システム環境で使用することが目的ではありません。「コード」は、特定のコードの構文及び表現規則を分かりやすく説明及び視覚化することのみを目的としています。SAP は、この文書に記載される「コード」の正確性及び完全性の保証を行いません。更に、SAP は、「コード」の使用により発生したエラー又は損害が SAP の故意又は重大な過失が原因で発生させたものでない限り、そのエラー又は損害に対して一切責任を負いません。

## アクセシビリティ

この SAP 文書に含まれる情報は、公開日現在のアクセシビリティ基準に関する SAP の最新の見解を表明するものであり、ソフトウェア製品のアクセシビリティ機能の確実な提供方法に関する拘束力のあるガイドラインとして意図されるものではありません。SAP は、この文書に関する一切の責任を明確に放棄するものです。ただし、この免責事項は、SAP の意図的な違法行為または重大な過失による場合は、適用されません。さらに、この文書により SAP の直接的または間接的な契約上の義務が発生することは一切ありません。

## ジェンダーニュートラルな表現

SAP 文書では、可能な限りジェンダーニュートラルな表現を使用しています。文脈により、文書の読者は「あなた」と直接的な呼ばれ方をされたり、ジェンダーニュートラルな名詞 (例:「販売員」又は「勤務日数」) で表現されます。ただし、男女両方を指すとき、三人称単数形の使用が避けられない又はジェンダーニュートラルな名詞が存在しない場合、SAP はその名詞又は代名詞の男性形を使用する権利を有します。これは、文書を分かりやすくするためです。

## インターネットハイパーリンク

SAP 文書にはインターネットへのハイパーリンクが含まれる場合があります。これらのハイパーリンクは、関連情報を見い出すヒントを提供することが目的です。SAP は、この関連情報の可用性や正確性又はこの情報が特定の目的に役立つことの保証を行いません。SAP は、関連情報の使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。すべてのリンクは、透明性を目的に分類されています (<http://help.sap.com/disclaimer> を参照)。

[go.sap.com/registration/  
contact.html](http://go.sap.com/registration/contact.html)

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

本書のいかなる部分も、SAP SE 又は SAP の関連会社の明示的な許可なくして、いかなる形式でも、いかなる目的にも複製又は伝送することはできません。本書に記載された情報は、予告なしに変更されることがあります。SAP SE 及びその頒布業者によって販売される一部のソフトウェア製品には、他のソフトウェアベンダーの専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。

これらの文書は、いかなる種類の表明又は保証もなしで、情報提供のみを目的として、SAP SE 又はその関連会社によって提供され、SAP 又はその関連会社は、これら文書に関する誤記脱落等の過失に対する責任を負うものではありません。SAP 又はその関連会社の製品及びサービスに対する唯一の保証は、当該製品及びサービスに伴う明示的な保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。

本書に記載される SAP 及びその他の SAP の製品やサービス、並びにそれらの個々のロゴは、ドイツ及びその他の国における SAP SE (又は SAP の関連会社) の商標若しくは登録商標です。本書に記載されたその他すべての製品およびサービス名は、それぞれの企業の商標です。

商標に関する詳細の情報や通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx> をご覧ください。