iAnywhere SOLUTIONS™
A SYBASE COMPANY

SQL Anywhere® Studio の紹介

パート番号：DC03920-01-0902-01
改訂：2005 年 3 月

版権

Copyright © 2005 iAnywhere Solutions, Inc., Sybase, Inc. All rights reserved.

ここに記載されている内容を iAnywhere Solutions, Inc.、Sybase, Inc. またはその関連会社の書面による事前許可を得ずに電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても複製、転載、翻訳することを禁じます。

Sybase、SYBASE のロゴ、Adaptive Server、AnswerBase、Anywhere、EIP、Embedded SQL、Enterprise Connect、Enterprise Portal、GainMomentum、iAnywhere、jConnect MASS DEPLOYMENT、Netimpact、ObjectConnect、ObjectCycle、OmniConnect、Open ClientConnect、Open ServerConnect、PowerBuilder、PowerDynamo、Powersoft、Quickstart Datamart、Replication Agent、Replication Driver、SQL Anywhere、SQL Central、SQL Remote、Support Plus、SWAT、Sybase IQ、Sybase System 11、Sybase WAREHOUSE、SyBooks、XA-Library は米国法人 Sybase, Inc. の登録商標です。Backup Server、Client-Library、jConnect for JDBC、MainframeConnect、Net-Gateway、Net-Library、Open Client、Open Client/Server、S-Designor、SQL Advantage、SQL Debug、SQL Server、SQL Server Manager、Sybase Central、Watcom、Web.SQL、XP Server は米国法人 Sybase, Inc. の商標です。

ここに記載されている上記以外の社名および製品名は、各社の商標または登録商標の場合があります。

目次

はじめに ..... **ix**
SQL Anywhere Studio のマニュアル x
オンラインでの情報のアクセス xiv
表記の規則 xviii
Adaptive Server Anywhere サンプル・データベース xxi
CustDB サンプル・データベース xxii
詳細情報の検索／フィードバックの提供 xxiii

1 SQL Anywhere Studio の紹介 **3**
SQL Anywhere Studio へようこそ 4
SQL Anywhere Studio のインストール 7

2 Adaptive Server Anywhere と Ultra Light の概要 **11**
Adaptive Server Anywhere の概要 12
使用目的 13
特長 14
データベース・サーバ 16
アプリケーション 17
Ultra Light の概要 19
Adaptive Server Anywhere と Ultra Light の比較 21

3 レプリケーション・テクノロジの概要 **25**
概要 26
統合リモート・データベース 28
伝達方法 32
Sybase のレプリケーション・テクノロジ 36

4 SQL Anywhere Studio for Windows CE **45**
はじめに 46

サンプル・アプリケーションの使用法 55
Windows CE データベースの設定 66
Windows CE のデータベース・サーバの使用 78
管理ユーティリティの使用 87
Windows CE のデータのバックアップとリカバリ 102
Windows CE でサポートされていない SQL Anywhere Studio の機能 112

5 SQL Anywhere Studio がサポートしているプラットフォーム 129
概要 130
Windows オペレーティング・システムと NetWare オペレーティング・システム. 132
UNIX、Linux、Macintosh のオペレーティング・システム 139
Mobile Link 同期統合データベース 144
iAnywhere Solutions ODBC ドライバがサポートしているプラットフォーム 145
Ultra Light ターゲット・プラットフォーム 147
オペレーティング・システムのバージョン 149

6 データベース・アプリケーションのアーキテクチャ 157
リレーショナル・データベースの概念 158
データベース・システムの各部分 164
各部分を組み合わせる方法 167
多層コンピューティングのアーキテクチャ 171
複数のデータベースの使用 172
アプリケーション・プログラミング・インタフェース 174
Adaptive Server Anywhere の内部構造 180

7 データベース・テーブルからのデータの選択 183
SQL とデータベース・コンピューティング 184
SELECT 文 187
テーブル全体の選択 188
テーブルからのカラムの選択 190
クエリ結果の順序付け 193
テーブルからのローの選択 197

8 複数のテーブルからのデータの選択 205
概要 206
直積を使用したテーブルのジョイン 208
ON 句を使用したジョインの制限 209

キー・ジョインを使用したテーブルのジョイン....211
ナチュラル・ジョインを使用したテーブルのジョイン....214
外部ジョインを使用したテーブルのジョイン....216

9 サブクエリを使用したデータの選択....217
サブクエリの概要....218
単一ローのサブクエリと複数ローのサブクエリ....221
ジョインに代わるサブクエリの使用....224

10 集合データの選択....227
データの要約....228
集合関数の概要....229
グループ分けされたデータに対する集合関数の適用....231
グループの制限....233

11 データベースの更新....235
概要....236
テーブルへのローの追加....237
テーブル内のローの変更....238
ローの削除....239
トランザクションによる変更のグループ分け....240
整合性の検査....244

12 サンプル・データベース....251
サンプル・データベースについて....252
レッスン 1 ：サンプル・データベースのコピーの作成....254
レッスン 2 ：Adaptive Server Anywhere データベース・サーバの起動....255
レッスン 3 ：データベース・サーバ・ウィンドウの表示....257
レッスン 4 ：データベース・サーバの停止....259
まとめ....260

13 接続の確立....261
接続について....262
ODBC データ・ソースについて....263
レッスン 1 ：ODBC データ・ソースの作成....264
レッスン 2 ：Sample ODBC データ・ソースを使用した接続....267
レッスン 3 ：他の接続方法....268

レッスン 4 ：サンプル・データベースからの切断 269
まとめ 270

14 Interactive SQL の使用 271
Interactive SQL について 272
レッスン 1 ：Interactive SQL の起動 273
レッスン 2 ：Interactive SQL のインタフェース 274
レッスン 3 ：Interactive SQL を使用したデータの表示 281
レッスン 4 ：SQL 文の使用 288
まとめ 297

15 Sybase Central を使用したデータベースの管理 299
Sybase Central について 300
レッスン 1 ：Sybase Central の起動と接続 301
レッスン 2 ：Sybase Central のインタフェース 302
レッスン 3 ：サンプル・データベースの表示 304
レッスン 4 ：テーブルの作成と編集 311
レッスン 5 ：ユーザとグループの管理 314
レッスン 6 ：ストアド・プロシージャの表示と編集 317
レッスン 7 ：データベースのバックアップ 322
サンプル・データベースのリストア 324
まとめ 325

16 データベースの設計と構築 327
新しいデータベースの設計について 328
単純なデータベースの設計と構築 329
レッスン 1 ：データベース・ファイルの作成 330
レッスン 2 ：データベースへの接続 332
レッスン 3 ：テーブルの設計と作成 334
レッスン 4 ：プライマリ・キーの識別と作成 337
レッスン 5 ：カラムのプロパティ設計 339
レッスン 6 ：テーブル間の関係の設計と作成 343
まとめ 346

17 データベースと Mobile Link の同期 347
Mobile Link について 348
概要 350

レッスン 1 ：データベースの作成 352
レッスン 2 ：同期のためのデータベースの準備....357
レッスン 3 ：Mobile Link 同期サーバの起動....362
レッスン 4 ：Mobile Link 同期 クライアント・ユーティリティの実行 363
まとめ 365

18 SQL Remote を使用したデータのレプリケーション 367
SQL Remote について 368
レッスン 1：はじめに 369
レッスン 2：統合データベースの設定 371
レッスン 3：リモート・データベースの設定 376
レッスン 4：データのレプリケーション 379
レッスン 5：データベースとデータベース設定値のリストア 384
まとめ 387

19 PowerDesigner によるデータベースの設計 389
PowerDesigner について 390
レッスン 1：はじめに 392
レッスン 2：カラムの追加 398
レッスン 3：作業の確認....401
レッスン 4：変更の保存とデータベースの生成....402
まとめ 406

20 InfoMaker を使用したレポート作成.... 409
InfoMaker について....410
レッスン 1：はじめに 411
レッスン 2：基本レポートの作成 413
レッスン 3：レポートの強化 417
まとめ 422

21 Microsoft VisualBasic クイック・スタート 423
チュートリアル：Visual Basic アプリケーションの開発....424

用語集 431

索引.... 453

はじめに

このマニュアルの内容

このマニュアルでは、モバイル・コンピューティング、埋め込みコンピューティング、ワークグループ・コンピューティングに使用する、完全なリレーショナル・データ管理同期システムである SQL Anywhere Studio について説明します。

SQL Anywhere Studio は、リレーショナル・データを操作するための一連のソフトウェア・コンポーネントです。このソフトウェアには、データの同期とレプリケーションのテクノロジに加え、Adaptive Server Anywhere および Ultra Light リレーショナル・データベース管理システムも含まれています。また、データベースの設計と配備を行うためのアプリケーションや、カスタム・レポートとデータ・エントリ・フォームを作成するアプリケーションも含まれます。

対象読者

このマニュアルは、SQL Anywhere Studio を使用するすべてのアプリケーション開発者とデータベース管理者を対象としています。

始める前に

このマニュアルでは、リレーショナル・データベースと SQL についての初歩的な知識があることを前提としています。

SQL Anywhere Studio のマニュアル

このマニュアルは、SQL Anywhere のマニュアル・セットの一部です。この項では、マニュアル・セットに含まれる各マニュアルと使用法について説明します。

SQL Anywhere Studio のマニュアル

SQL Anywhere Studio のマニュアルは、各マニュアルを 1 つの大きなヘルプ・ファイルにまとめたオンライン形式、マニュアル別の PDF ファイル、および有料の製本版マニュアルで提供されます。SQL Anywhere Studio のマニュアルは、次の分冊マニュアルで構成されています。

- **『SQL Anywhere Studio の紹介』** このマニュアルでは、SQL Anywhere Studio のデータベース管理と同期テクノロジの概要について説明します。また、SQL Anywhere Studio を構成する各部分について説明するチュートリアルも含まれています。
- **『SQL Anywhere Studio 新機能ガイド』** このマニュアルは、SQL Anywhere Studio のこれまでのリリースのユーザを対象としています。ここでは、製品の今回のリリースと以前のリリースで導入された新機能をリストし、アップグレード手順を説明しています。
- **『Adaptive Server Anywhere データベース管理ガイド』** このマニュアルでは、データベースおよびデータベース・サーバの実行、管理、設定について説明しています。
- **『Adaptive Server Anywhere SQL ユーザーズ・ガイド』** このマニュアルでは、データベースの設計と作成の方法、データのインポート・エクスポート・変更の方法、データの検索方法、ストアド・プロシージャとトリガの構築方法について説明します。
- **『Adaptive Server Anywhere SQL リファレンス・マニュアル』** このマニュアルは、Adaptive Server Anywhere で使用する SQL 言語の完全なリファレンスです。また、Adaptive Server Anywhere のシステム・テーブルとシステム・プロシージャについても説明しています。
- **『Adaptive Server Anywhere プログラミング・ガイド』** このマニュアルでは、C、C++、Java プログラミング言語を使用してデータベース・アプリケーションを構築、配備する方法につい

て説明します。Visual Basic や PowerBuilder などのツールのユーザは、それらのツールのプログラミング・インタフェースを使用できます。また、Adaptive Server Anywhere ADO.NET データ・プロバイダについても説明します。

- **『Adaptive Server Anywhere SNMP Extension Agent ユーザーズ・ガイド』** このマニュアルでは、Adaptive Server Anywhere SNMP Extension Agent を SNMP 管理アプリケーションで使用して Adaptive Server Anywhere のデータベースを管理するための設定方法について説明します。

- **『Adaptive Server Anywhere エラー・メッセージ』** このマニュアルでは、Adaptive Server Anywhere エラー・メッセージの完全なリストを、その診断情報とともに説明します。

- **『SQL Anywhere Studio セキュリティ・ガイド』** このマニュアルでは、Adaptive Server Anywhere データベースのセキュリティ機能について説明します。Adaptive Server Anywhere 7.0 は、米国政府から TCSEC (Trusted Computer System Evaluation Criteria) の C2 セキュリティ評価を授与されています。このマニュアルには、Adaptive Server Anywhere の現在のバージョンを、C2 基準を満たした環境と同等の方法で実行することを望んでいるユーザにとって役に立つ情報が含まれています。

- **『Mobile Link 管理ガイド』** このマニュアルでは、モバイル・コンピューティング用の Mobile Link データ同期システムについてあらゆる角度から説明します。このシステムによって、Oracle、Sybase、Microsoft、IBM の単一データベースと、Adaptive Server Anywhere や Ultra Light の複数データベースの間でのデータ共有が可能になります。

- **『Mobile Link クライアント』** このマニュアルでは、Adaptive Server Anywhere リモート・データベースと Ultra Light リモート・データベースを設定して、これらを同期させる方法について説明します。

- **『Mobile Link サーバ起動同期ユーザーズ・ガイド』** このマニュアルでは、Mobile Link のサーバによって開始される同期について説明します。サーバによって開始される同期とは、統合データベースから同期の開始を可能にする Mobile Link の機能です。

- **『Mobile Link チュートリアル』** このマニュアルには、Mobile Link アプリケーションの設定手順をわかりやすく説明したチュートリアルが複数含まれています。

- **『QAnywhere ユーザーズ・ガイド』** このマニュアルでは、Mobile Link QAnywhere について説明します。Mobile Link QAnywhere は、従来のデスクトップ・クライアントやラップトップ・クライアントだけでなく、モバイル・クライアントや無線クライアント用のメッセージング・アプリケーションの開発と展開を可能にするメッセージング・プラットフォームです。

- **『Mobile Link およびリモート・データ・アクセスの ODBC ドライバ』** このマニュアルでは、Mobile Link 同期サーバから、または Adaptive Server Anywhere リモート・データ・アクセスによって、Adaptive Server Anywhere 以外の統合データベースにアクセスするための ODBC ドライバの設定方法について説明します。

- **『SQL Remote ユーザーズ・ガイド』** このマニュアルでは、モバイル・コンピューティング用の SQL Remote データ・レプリケーション・システムについて、あらゆる角度から説明します。このシステムによって、Adaptive Server Anywhere または Adaptive Server Enterprise の単一データベースと Adaptive Server Anywhere の複数データベースの間で、電子メールやファイル転送などの間接的リンクを使用したデータ共有が可能になります。

- **『SQL Anywhere Studio ヘルプ』** このマニュアルには、Sybase Central や Interactive SQL、その他のグラフィカル・ツールに関するコンテキスト別のヘルプが含まれています。これは、製本版マニュアル・セットには含まれていません。

- **『Ultra Light データベース・ユーザーズ・ガイド』** このマニュアルは、Ultra Light 開発者を対象としています。ここでは、Ultra Light データベース・システムの概要について説明します。また、すべての Ultra Light プログラミング・インタフェースに共通する情報を提供します。

- **Ultra Light のインタフェースに関するマニュアル** 各 Ultra Light プログラミング・インタフェースには、それぞれに対応するマニュアルを用意しています。これらのインタフェースは、RAD

(ラピッド・アプリケーション開発) 用の Ultra Light コンポーネントとして提供されているものと、C、C++、Java 開発用の静的インタフェースとして提供されているものがあります。

このマニュアル・セットの他に、PowerDesigner と InfoMaker には、独自のオンライン・マニュアル (英語版) がそれぞれ用意されています。

マニュアルの形式

SQL Anywhere Studio のマニュアルは、次の形式で提供されています。

- **オンライン・マニュアル** オンライン・マニュアルには、SQL Anywhere Studio の完全なマニュアルがあり、SQL Anywhere ツールに関する印刷マニュアルとコンテキスト別のヘルプの両方が含まれています。オンライン・マニュアルは、製品のメンテナンス・リリースごとに更新されます。これは、最新の情報を含む最も完全なマニュアルです。

  Windows オペレーティング・システムでオンライン・マニュアルにアクセスするには、[スタート] － [プログラム] － [SQL Anywhere 9] － [オンライン・マニュアル] を選択します。オンライン・マニュアルをナビゲートするには、左ウィンドウ枠で HTML ヘルプの目次、索引、検索機能を使用し、右ウィンドウ枠でリンク情報とメニューを使用します。

  UNIX オペレーティング・システムでオンライン・マニュアルにアクセスするには、SQL Anywhere のインストール・ディレクトリに保存されている HTML マニュアルを参照してください。

- **PDF 版マニュアル** SQL Anywhere の各マニュアルは、Adobe Acrobat Reader で表示できる PDF ファイルで提供されています。

  PDF 版マニュアルは、オンライン・マニュアルまたは Windows の [スタート] メニューから利用できます。

- **製本版マニュアル** 製本版マニュアルをご希望の方は、ご購入いただいた販売代理店または弊社営業担当までご連絡ください。

オンラインでの情報のアクセス

SQL Anywhere Studio のオンライン・マニュアルには、様々な情報へアクセスするのに役立つ多数のツールが用意されています。

オンライン・マニュアル (PDF)

オンライン・マニュアル (PDF) は、製本版マニュアルの完全なコピーで、Adobe Acrobat で表示可能な PDF 形式で作成されています。次に示す機能が使用できるかどうかと、その機能の外観は、使用している Acrobat または Acrobat Reader のバージョンによって異なります。

- **[しおり] タブ**　[しおり] タブは、現在のマニュアルにあるすべての章のリストを表示します。任意の章またはページをクリックして表示したり、見出しを展開または折りたたむことができ、マニュアルの構成を見ることもできます。見出しをクリックすると、ウィンドウの右側にページが表示されます。[しおり] タブを使用すると、マニュアルの構成の概要を表示したり、特定の章やページにすばやくアクセスできます。

- **アクティブ・リンク**　アクティブ・リンクは「参照」とも呼ばれ、青色で強調表示されています。リンクの上にカーソルを移動すると、手のひらマークが指差しマークに変わります。リンクをクリックすると、現在表示されているマニュアルまたは他のマニュアル内の、そのリンクが指す場所に移動します。Acrobat が特定のブラウザへのプラグインとして実行されている場合、マニュアル間のリンクは正常に動作しません。これらのリンクは、Internet Explorer または直接 Acrobat を使用している場合に正常に動作します。

- **[前の画面] ボタンと [次の画面] ボタン**　ヘルプ・ウィンドウの左上にある [前の画面] ボタンと [次の画面] ボタンを使用すると、行った手順を再度追跡し、移動した順序でページを再表示できます。[前の画面] ボタンをクリックするたびに、現在表示しているページの直前に表示したページにトピックが移動します。[次の画面] ボタンは、少なくとも一度 [前の画面] ボタンをクリックした後に使用可能になり、一番最後に表示したページに戻るまで有効です。

- **[最初のページ] ボタンと [最後のページ] ボタン**　[最初のページ] ボタンをクリックすると、マニュアルの最初のページに戻ります。また、[最後のページ] ボタンをクリックすると、マニュアルの最後のページに進みます。

オンライン・マニュアル

オンライン・マニュアル (PDF) は、製本版マニュアルの完全なコピーで、Adobe Acrobat で表示可能な PDF 形式で作成されています。

- **[目次] タブ** [目次] タブには、SQL Anywhere Studio マニュアル・セットに含まれるすべてのマニュアルのリストが表示されます。任意のマニュアルをダブルクリックしてそのマニュアルの章を表示したり、各章をクリックしてそのページを表示できます。[目次] タブを使用すると、マニュアルの構成の概要を表示したり、特定の章やページにすばやくアクセスできます。

- **ページの見出し** 各ページの上部にはページの見出しが表示され、現在のページのロケーションが示されます。現在のロケーションは、マニュアル・タイトル、章タイトル、ページ・タイトルの順に表示されます。マニュアルまたは章タイトルをクリックして、マニュアルのその部分に移動できます。ウィンドウの右側にあるページを移動すると、ウィンドウの左側にある [目次] タブのプレースホルダがそれに応じて移動されます。

- **[キーワード] タブ** [キーワード] タブには、章とページに関連するキーワードが含まれています。[キーワード] タブの上部のボックスに語句を入力すると、下部のスクロール・ボックス内にあるキーワードのトピックに移動できます。探している項目を見つけたら、その項目をダブルクリックして、ページを表示します。

- **[検索] タブ** [検索] タブの上部のボックスに語句を入力し、[検索開始] をクリックすると、下部のスクロール・ボックスにその語句またはフレーズを含むすべてのトピックが表示されます。正確に一致する項目を検索する場合は、その語句を二重引用符 (") で囲みます。さらに検索を絞り込むには、ページの下部にあるチェックボックスを選択したり、[探したい語句を入力してください] ボックスの隣にある矢印ボタンをクリックして、ブール検索用語を使用します。[以前の結果から検索] を使用すると、以前の検索で表示されたトピック内で次の検索を絞り込むことができます。[類似する文字に合致] は、検索された語句のつづりに近い語句を探します。[タイトルのみ検索] は、目的の語句に一致するタイトルのみを検索します。探している項目を見つけたら、その項目をダブルクリックして、ページを表示します。

- **[お気に入り] タブ**　今後もう一度アクセスしたいと思うページがある場合は、[お気に入り] リストにその項目を追加することで、すばやくアクセスできます。保存するページを表示して [お気に入り] タブをクリックし、次にウィンドウの下部にある [追加] ボタンをクリックします。[追加] ボタンをクリックすると、[現在のトピック] ボックスにリストされているトピックが ([トピック] リストに) 追加されます。[お気に入り] リストは、トピックを選択し [削除] をクリックすることによって修正できます。[お気に入り] は、オンライン・マニュアルのアップグレード版でも保持されます。

- **[前へ] ボタンと [次へ] ボタン**　ヘルプ・ウィンドウの右上にある [前へ] ボタンと [次へ] ボタンは、マニュアルのページをめくるのと同じです。[前へ] は、1 つ前のトピック・ページに戻ります (マニュアル・ページの順序での前のページに移動します)。[次へ] は、1 つ後ろのトピック・ページに戻ります (マニュアル・ページの順序での次のページに移動します)。

- **[戻る] ボタンと [進む] ボタン**　メニュー・バーの [戻る] ボタンと [進む] ボタンを使用すると、行った手順を再度追跡し、移動した順序でページを再表示できます。[戻る] ボタンをクリックするたびに、現在表示しているページの直前に表示したページにトピックが移動します。[進む] ボタンは、少なくとも一度 [戻る] ボタンをクリックした後に使用可能になり、一番最後に表示したページに戻るまで有効です。

- **[Web リンク] ボタン**　ヘルプ・ウィンドウの上部にある [Web リンク] ボタンを使用すると、iAnywhere の Web サイト、テクニカル・サポート、オンライン登録、ソフトウェアのダウンロード、iAnywhere Solutions のデベロッパーズ・コミュニティなど、iAnywhere Solutions リソースへすばやくアクセスできます。

- **[ブラウズ] ボタン**　[ブラウズ] ボタンをクリックすると、[チュートリアル]、[目的から探す]、[用語解説] にすばやくアクセスできます。

- **[閉じる] ボタン**　一部のページには、折りたたみ可能な見出しが含まれています。折りたためる場合は、見出しの左側にプラス記号 (+) またはマイナス記号 (-) が示されています。[閉じる] ボタンをクリックしてすべての見出しを閉じたり、[開く] ボタンをクリックしてすべての見出しを開くことができます。

- **[PDF] ボタン**　[PDF] ボタンをクリックすると、現在のマニュアルが PDF 形式で開かれます。

表記の規則

この項では、このマニュアルで使用されている書体およびグラフィック表現の規則について説明します。

SQL 構文の表記規則

SQL 構文の表記には、次の規則が適用されます。

- **キーワード**　SQL キーワードはすべて次の例に示す ALTER TABLE のように大文字で表記します。

  ALTER TABLE [*owner*.]*table-name*

- **プレースホルダ**　適切な識別子または式で置き換えられる項目は、次の例に示す *owner* や *table-name* のように表記します。

  ALTER TABLE [*owner*.]*table-name*

- **繰り返し項目**　繰り返し項目のリストは、次の例に示す *column-constraint* のように、リストの要素の後ろに省略記号 (ピリオド 3 つ …) を付けて表します。

  ADD *column-definition* [*column-constraint*, …]

  複数の要素を指定できます。複数の要素を指定する場合は、各要素間をカンマで区切る必要があります。

- **オプション部分**　文のオプション部分は角カッコで囲みます。

  RELEASE SAVEPOINT [*savepoint-name*]

  この例では、角カッコで囲まれた *savepoint-name* がオプション部分です。角カッコは入力しないでください。

- **オプション**　項目リストから 1 つだけ選択するか、何も選択しなくてもよい場合は、項目間を縦線で区切り、リスト全体を角カッコで囲みます。

  [**ASC** | **DESC**]

  この例では、ASC と DESC のどちらか 1 つを選択しても、どちらも選択しなくてもかまいません。角カッコは入力しないでください。

- **選択肢**　オプションの中の 1 つを必ず選択しなければならない場合は、選択肢を中カッコで囲み、縦棒で区切ります。

  [**QUOTES** { **ON** | **OFF** }]

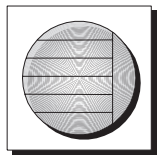
  QUOTES オプションを使用する場合は、ON または OFF のどちらかを選択する必要があります。角カッコと中カッコは入力しないでください。

グラフィック・アイコン

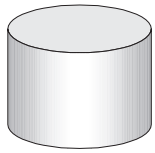
このマニュアルでは、次のアイコンを使用します。

- クライアント・アプリケーション

- Sybase Adaptive Server Anywhere などのデータベース・サーバ



- データベース。高度な図では、データベースとデータベースを管理するデータ・サーバの両方をこのアイコンで表します。



- レプリケーションまたは同期のミドルウェア。ソフトウェアのこれらの部分は、データベース間のデータ共有を支援します。たとえば、Mobile Link 同期サーバ、SQL Remote Message Agent などがあげられます。

- プログラミング・インタフェース

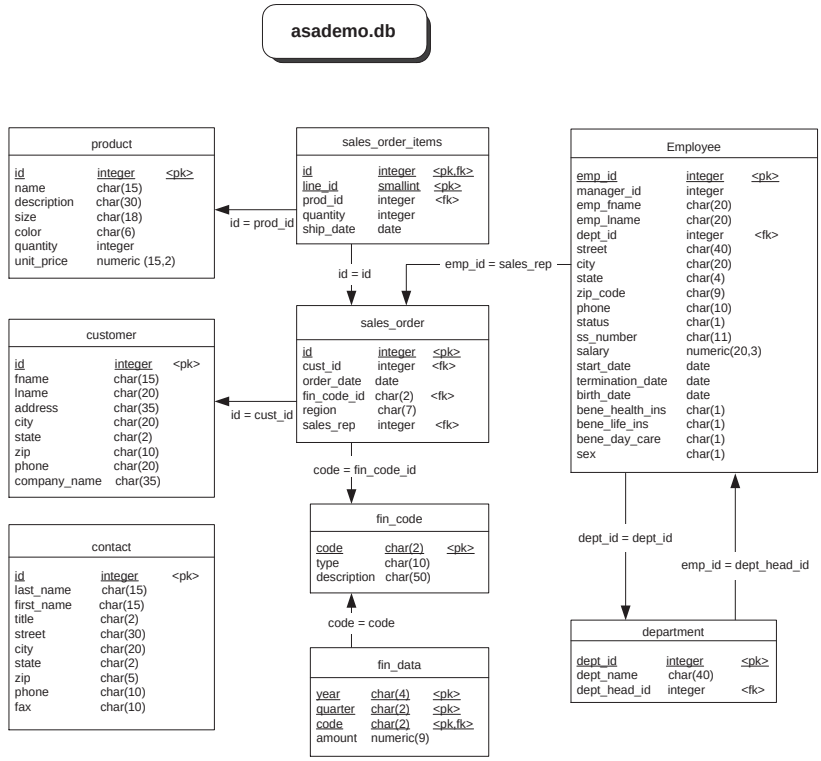
Adaptive Server Anywhere サンプル・データベース

このマニュアルでは、多くの例で Adaptive Server Anywhere サンプル・データベースが使用されています。

サンプル・データベースは、*asademo.db* という名前のファイルに保存され、SQL Anywhere ディレクトリに置かれています。

サンプル・データベースは小規模の企業の例です。データベースには、この企業の内部情報 (従業員、部署、経理) とともに、製品情報や販売情報 (受注、顧客、連絡先) が入っています。データベースに含まれる情報はすべて架空のものです。

次の図は、サンプル・データベース内のテーブルと各テーブル間の関係を示しています。



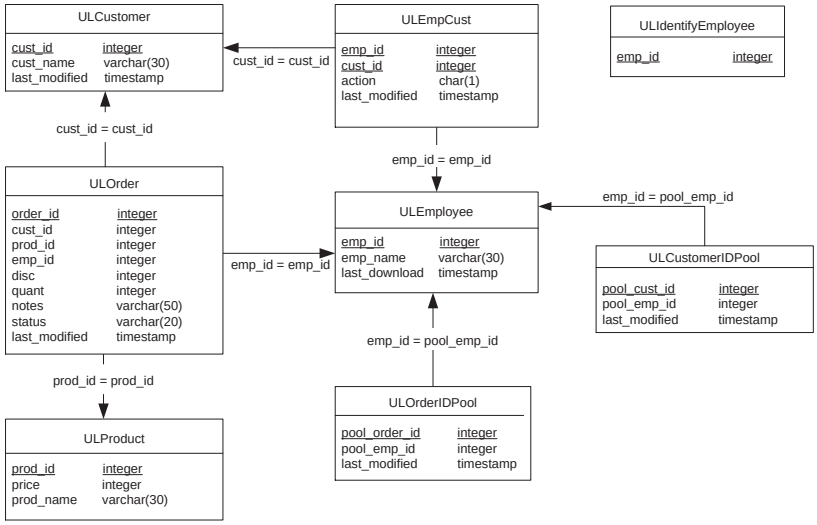
CustDB サンプル・データベース

Mobile Link と Ultra Light のマニュアルでは、多くの例で Ultra Light のサンプル・データベースが使用されています。

Ultra Light サンプル・データベースのリファレンス・データベースは、*custdb.db* という名前のファイルに保存され、SQL Anywhere ディレクトリのサブディレクトリ *Samples¥UltraLite¥CustDB* に置かれています。全面的にこのデータベースを使用して構築したアプリケーションも提供されています。

サンプル・データベースは、あるハードウェア販売会社の販売管理データベースです。データベースには、この販売会社の顧客、製品、営業戦力に関する情報が入っています。

次の図は、CustDB データベース内のテーブルと、各テーブル間の関係を示しています。



詳細情報の検索／フィードバックの提供

このマニュアルに関するご意見、ご提案、フィードバックをお寄せください。

マニュアルおよびソフトウェアに関するフィードバックは、SQL Anywhere のテクノロジについて議論するニュースグループを介してお送りいただけます。ニュースグループは、ニュース・サーバ *forums.sybase.com* にあります (ニュースグループにおけるサービスは英語でのみの提供となります)。

以下のニュースグループがあります。

- sybase.public.sqlanywhere.general
- sybase.public.sqlanywhere.linux
- sybase.public.sqlanywhere.mobilink
- sybase.public.sqlanywhere.product_futures_discussion
- sybase.public.sqlanywhere.replication
- sybase.public.sqlanywhere.ultralite
- ianywhere.public.sqlanywhere.qanywhere

ニュースグループに関するお断り

iAnywhere Solutions は、ニュースグループ上に解決策、情報、または意見を提供する義務を負うものではありません。また、システム・オペレータ以外のスタッフにこのサービスを監視させて、操作状況や可用性を保証する義務もありません。

iAnywhere Solutions のテクニカル・アドバイザとその他のスタッフは、時間のある場合にかぎりニュースグループでの支援を行います。こうした支援は基本的にボランティアで行われるため、解決策や情報を定期的に提供できるとはかぎりません。支援できるかどうかは、スタッフの仕事量に左右されます。

マニュアルに関するご意見、ご提案は、SQL Anywhere ドキュメンテーション・チームの iasdoc@ianywhere.com 宛てに電子メールでお寄せください。このアドレスに送信された電子メールに返信する責任は負いませんが、お寄せ頂いたご意見、ご提案は必ず読ませて頂きます。

第 1 部　SQL Anywhere Studio の概要

第 1 部では、SQL Anywhere Studio について説明します。また、SQL Anywhere Studio のデータベース管理テクノロジとレプリケーション・テクノロジについても説明します。

第 1 章

SQL Anywhere Studio の紹介

この章の内容

この章では、SQL Anywhere Studio を構成するコンポーネントについて説明します。

SQL Anywhere Studio へようこそ

SQL Anywhere Studio を使用して、組織全体のワークグループ、モバイル、組み込みデータベース・システムに情報を伝えることができます。

SQL Anywhere Studio には、次のコンポーネントが含まれています。

リレーショナル・データベース・システム

- **Adaptive Server Anywhere** 製品の核となるリレーショナル・データベースは、トランザクションベースの SQL データベースであり、個人ユーザ用とワークグループ用に設計されています。Adaptive Server Anywhere は、さまざまなバージョンの Windows や UNIX に加え、Novell NetWare など、幅広いオペレーティング・システム上で実行できます。ハードウェアについても同様に、複数の CPU を使用するワークグループ・サーバから、リソースが非常に限られた PC、そして Windows CE デバイスに至るまで幅広いハードウェア上で実行できます。

- **Ultra Light** リレーショナル・データベース・アプリケーションを構築し、Palm Computing Platform や Windows CE などの小型のデバイスに配備するために使用します。Ultra Light には、Mobile Link 同期をサポートする機能が組み込まれています。

  Ultra Light は特に小型のデバイス用に設計されています。このコンポーネントを使用すると、ディスク容量が 50 KB 未満のリレーショナル・データベースを構築できます。

データ同期テクノロジ

- **Mobile Link** 統合データベースと、Ultra Light または Adaptive Server Anywhere の多数のリモート・データベース間のデータの双方向同期に使用します。中央のデータベースには、Adaptive Server Anywhere、Adaptive Server Enterprise、または Oracle、Microsoft SQL Server、IBM DB2 などの DBMS を使用できます。

- **SQL Remote** 中央のデータベースと多数のリモート・データベース間で、データの双方向レプリケーションをメッセージベースで行うために使用します。SQL Remote を使用すると、ラップトップ・コンピュータと中央のデータベース間で、電子メールやダイアルアップによるアクセスを使用してデータをレプリケートできます。

- **Replication Agent**　Sybase Replication Server 経由で Adaptive Server Anywhere データベースから他のデータベースにデータをレプリケートするために使用します。

開発、設計、管理ツール

- **InfoMaker**　データベースへの問い合わせと、データについて高度で効果的なカスタム・レポートを作成するために使用します。InfoMaker もパーソナル・データ・アシスタントで、これを使用して、あらゆる方法でデータを処理できます。

- **PowerDesigner**　データベースの設計、作成、文書化、保守を行うために使用します。

- **管理ツールと開発ツール**　SQL Anywhere Studio には、Sybase Central データベース管理ツール、Interactive SQL データベース・ユーティリティの他に、クエリ・エディタとストアド・プロシージャ・デバッガが用意されています。

  SQL Anywhere Studio には、オプションでインストールできるアクセシビリティ有効化モジュールがあります。このコンポーネントには Sun AccessBridge モジュールがあり、Sybase Central または Interactive SQL を使用したときに常にロードされます。画面リーダなどのサード・パーティのソフトウェアは、このモジュールを活用してソフトウェアの機能にアクセスできます。

InfoMaker と PowerDesigner は、Windows オペレーティング・システムでのみ使用できます。ただし、クライアントとして使用する場合は、サポートされるオペレーティング・システム上で稼動するサーバに対して使用できます。

別途ライセンスが入手可能なコンポーネント

次のコンポーネントは別途ライセンスが入手可能であり、注文してからインストールしてください。

- **Java オプション**　データベース内で Java を使用できるようにする Java 仮想マシンとランタイム・クラスは、別途注文してください。

- **SQL Anywhere Studio のセキュリティ・オプション**　強力な暗号化用のソフトウェアは別途注文する必要があります。強力な暗号化は、次のような場合に使用できます。

  - Adaptive Server Anywhere のデータベース・サーバとクライアント間のトランスポート層のセキュリティ

- Mobile Link 同期サーバとクライアント間のトランスポート層のセキュリティ
- AES_FIPS データベースの暗号化 (AES_FIPS 以外の AES データベース・ファイルの暗号化は基本パッケージに含まれており、別のライセンスは不要です)。

Java とセキュリティのコンポーネントのご注文については、弊社までお問い合わせください。

SQL Anywhere Studio のインストール

SQL Anywhere Studio のインストール方法は、使用するオペレーティング・システムによって異なります。このソフトウエアをインストールする前に、正しくライセンス契約を行っていることを確認してください。

❖ SQL Anywhere Studio をインストールするには、次の手順に従います (Windows オペレーティング・システム版)。

- SQL Anywhere Studio CD-ROM のルート・ディレクトリにある *setup.exe* プログラムを実行すると、インストールが開始されます。セットアップ・ウィザードの指示に従ってインストールを進めてください。

  セットアップ・プログラムを使用すると、インストールするコンポーネントを選択できます。

❖ SQL Anywhere Studio をインストールするには、次の手順に従います (Novell NetWare 版または Windows CE 版)。

- Adaptive Server Anywhere NetWare 版は、NetWare サーバに接続しており、Microsoft の Windows オペレーティング・システムを実行するマシンからインストールします。SQL Anywhere Studio セットアップ・プログラムを実行して、NetWare または Windows CE のどちらかを選択します。

❖ SQL Anywhere Studio をインストールするには、次の手順に従います (UNIX 版)。

- インストールは、使用する UNIX オペレーティング・システムによって異なります。詳細については、SQL Anywhere Studio の『*Read Me First (UNIX 版)*』を参照してください。

SQL Anywhere プログラム・グループ

Windows オペレーティング・システムでは、ソフトウェアをインストールすると SQL Anywhere 「**プログラム・グループ**」が作成されます。このプログラム・グループにアクセスするには、[スタート] ボタンをクリックし、[プログラム] ー [SQL Anywhere 9] の順に選択します。

UNIX に SQL Anywhere Studio をインストールした場合、プログラム・グループは作成されません。

プログラム・グループの項目

プログラム・グループには、次のうちの一部またはすべての項目が含まれています。表示される項目は、ソフトウェアのインストール時に行った選択によって異なります。

- **Adaptive Server Anywhere** 次の項目があります。
  - **[Interactive SQL]** SQL 文をデータベースに送信する Interactive SQL ユーティリティを起動します。
  - **[ネットワーク・サーバのサンプル]** ネットワーク・データベース・サーバを起動し、サンプル・データベースをロードします。
  - **[ODBC アドミニストレータ]** ODBC データ・ソースの設定と編集を行う ODBC アドミニストレータ・プログラムを起動します。
  - **[パーソナル・サーバのサンプル]** パーソナル・サーバを起動し、サンプル・データベースをロードします。
- **[Mobile Link] プログラム・グループ** Mobile Link 同期プログラムとサンプルにアクセスします。
- **[Ultra Light] プログラム・グループ** Ultra Light プログラムとサンプルにアクセスします。
- **[更新のチェック]** お使いのバージョンの SQL Anywhere Studio に利用する最新の更新情報を掲載した Web ページにアクセスします (このオプションは日本語版では提供されていません)。

- **[iAnywhere オンライン・リソース]**　iAnywhere Solutions に関する情報を掲載した Web ページを開きます。
- **[オンライン・マニュアル]**　Adaptive Server Anywhere のオンライン・マニュアルを開きます。
- **[Sybase Central]**　データベース管理ツールである Sybase Central を起動します。

また、InfoMaker と PowerDesigner に関する項目が表示される場合もあります。

第 2 章

Adaptive Server Anywhere と Ultra Light の概要

この章の内容

この章では、Adaptive Server Anywhere リレーショナル・データベース・システムと、Ultra Light という専有容量の小さい Adaptive Server Anywhere データベースについて紹介します。これらの使用法、機能、相違点について説明します。

Adaptive Server Anywhere の概要

この章では、Adaptive Server Anywhere の特長と、どのような目的に使用できるかについて説明します。

Adaptive Server Anywhere では、データの格納や管理を目的とした一連のツールが提供されます。これらのツールを使用して、データベースへのデータ入力や、データベース構造の変更、データの表示または変更ができます。

Adaptive Server Anywhere リレーショナル・データベース管理システムは、SQL Anywhere Studio の中核部分です。Adaptive Server Anywhere は、フル装備の SQL データベースを必要とするタスクを、さまざまな環境で実行できるように設計されています。豊富なリソースを持つ環境においては、使用可能なメモリと CPU リソースを活用して高いパフォーマンスを提供します。さらに、物理リソースとデータベース管理リソースが限られている環境でも適切に動作します。このような環境には、モバイル・コンピューティング環境、組み込みデータベースの使用、小規模および中規模ビジネス用のデータベース・サーバとしての使用などがあります。

使用目的

Adaptive Server Anywhere は、次のような用途に最適です。

- **小規模および中規模ビジネスのデータベース・サーバ**　Adaptive Server Anywhere は、数人から数 100 人のユーザがいる任意の場所で小規模および中規模ビジネスの稼働条件を処理するために構築されています。Adaptive Server Anywhere では、ワークグループおよび企業用に高いパフォーマンスのデータベースが提供され、管理リソースとハードウェア・リソースが限られている環境に適しています。ただし、このような環境に限定される訳ではありません。

  Adaptive Server Anywhere は、複数の CPU と最大 64 GB のメモリを使用できます。数 10 ギガバイトの Adaptive Server Anywhere データベースを運用しているカスタマも存在します。

- **組み込みデータベース**　多くのアプリケーションには「バックグラウンドで機能する」データベースが必要です。個人情報管理システム、ドキュメント管理システム、ネットワーク監視アプリケーションなどの、情報を格納するアプリケーションです。Adaptive Server Anywhere は、これらのアプリケーションのデータベースとして適しています。Ultra Light 導入オプションは、リソースが非常に制限された埋め込み環境で使用します。

  組み込みデータベースの主な機能は、管理することなく完全に動作することです。Adaptive Server Anywhere のこの機能は、要求水準の高い多くの市販アプリケーションで実証されています。

- **モバイル・コンピューティング**　ラップトップ・コンピュータやノート型コンピュータは、現在では多くの職場に普及しています。Adaptive Server Anywhere は、これらのコンピュータの SQL データベースとして適しています。Mobile Link 同期と SQL Remote レプリケーションを使用すれば、Adaptive Server Anywhere により企業全体にトランザクションベースのコンピューティング環境を設定できます。Ultra Light 導入オプションと Mobile Link の同期機能を利用することにより、PDA のようなリソースに制限があるデバイスにも十分なデータベース機能を提供できます。

特長

Adaptive Server Anywhere は、次の技術的な特長に基づいて構築されています。

- **完全な SQL リレーショナル・データベース管理システム**
  Adaptive Server Anywhere は、リカバリ機能、オンライン・バックアップ、参照整合性動作、ストアド・プロシージャ、トリガ、ローレベルの同時制御、スケジュールとイベント、豊富な SQL 言語、SQL RDBMS に期待されるその他のすべての機能を備えた、トランザクション処理リレーショナル・データベース管理システム (RDBMS) です。

- **経済的なハードウェア要件**　Adaptive Server Anywhere では、ほかのデータベース管理システムに比べて、必要とするメモリやディスク・リソースが少なくてすみます。

- **簡単に使用できる**　Adaptive Server Anywhere はセルフチューニングで管理が容易です。通常のリレーショナル・データベース管理システムと異なり、大規模なデータベース管理は必要ありません。

- **スタンドアロンとネットワークの使用**　Adaptive Server Anywhere はスタンドアロンで使用できます。たとえば、データ処理を中心とするアプリケーション内の組み込みデータベース、マルチユーザのクライアント／サーバ内のネットワーク・サーバ、または 3 層環境におけるネットワーク・サーバとして使用できます。組み込みデータベース・システムとして使用すると、必要なときにアプリケーションによって自動的に起動されます。

- **ハイパフォーマンス**　Adaptive Server Anywhere は、簡単な管理と少ないリソースを意図して設計されていると同時に、スケーラブルでハイパフォーマンスな DBMS です。Adaptive Server Anywhere は、複数の CPU 上で実行でき、最新のクエリ・オプティマイザを備え、さらにパフォーマンス監視ツールとチューニング・ツールが用意されています。

- **業界標準のインタフェース**　Adaptive Server Anywhere では、ODBC アプリケーションから使用できるネイティブの ODBC 3.5 ドライバを提供してハイパフォーマンスを実現します。また、ActiveX Data Object (ADO) プログラミング環境から使用できる

OLE DB ドライバも提供します。これには Adaptive Server Anywhere 用の ADO.NET データ・プロバイダがあります。また、これは iAnywhere JDBC ドライバと Sybase jConnect for JDBC と共に提供され、Embedded SQL インタフェースと Sybase Open Client インタフェースもサポートしています。

- **プラットフォームを問わないソリューション**　Adaptive Server Anywhere は、Windows、Novell NetWare、Sun Solaris、Linux など、多くのオペレーティング・システムで実行できます。

  それぞれのプラットフォーム上で使用できるコンポーネントは異なる可能性があります。詳細については、「SQL Anywhere Studio がサポートしているプラットフォーム」129 ページを参照してください。

システムの稼働条件

SQL Anywhere Studio のコンポーネントをサポートしているオペレーティング・システムの詳細については、『SQL Anywhere Studio の紹介』> 「SQL Anywhere Studio がサポートしているプラットフォーム」を参照してください。

必要なネットワーク・ソフトウェア

Adaptive Server Anywhere ネットワーク・サーバを実行するためには、適切なネットワーク・ソフトウェアがインストールされて実行されている必要があります。

Adaptive Server Anywhere ネットワーク・サーバは、Windows、Novell NetWare、Linux、UNIX オペレーティング・システムで使用できます。

Adaptive Server Anywhere は、TCP/IP ネットワーク・プロトコルと Novell NetWare 用の SPX プロトコルをサポートしています。

データベース・サーバ

Adaptive Server Anywhere には、2 種類のデータベース・サーバがあります。

- **パーソナル・データベース・サーバ** このサーバは、同一マシンを使用するシングルユーザ用です。この例として、組み込みデータベース・サーバがあります。これは、開発作業にも適しています。

  パーソナル・サーバの実行プログラムは、次のとおりです。

  - UNIX オペレーティング・システム上では *dbeng9*
  - Windows 上 (Windows CE を除く) では *dbeng9.exe*

- **ネットワーク・データベース・サーバ** パーソナル・サーバの機能に加え、ネットワーク・サーバではネットワークを経由したクライアント／サーバ通信をサポートしています。このサーバはマルチユーザ用です。

  ネットワーク・サーバの実行プログラムは、次のとおりです。

  - UNIX オペレーティング・システム上では *dbsrv9*
  - Windows NT 上 (Windows CE を含む) では *dbsrv9.exe*
  - Novell NetWare 上では、*dbsrv9.nlm* と呼ばれる NetWare Loadable Module (NLM) サーバ

2 種類のサーバで同一の SQL 機能

要求処理エンジンは、2 種類のサーバで同一です。これらの 2 種類のサーバでは、同一の SQL 言語とデータベース機能がサポートされています。パーソナル・サーバでは、ネットワークを経由した通信、10 を超える同時接続、複数の CPU の使用はサポートされていません。パーソナル・サーバ用に開発されたアプリケーションは、ネットワーク・サーバでも同じように動作します。

アプリケーション

この項では、SQL Anywhere Studio 製品の Adaptive Server Anywhere に付属して提供されるデータベース・アプリケーションのいくつかを説明します。これらは、ユーザがデータベースを設計、構築、管理するためのアプリケーションです。

インストールされるコンポーネントは、使用するオペレーティング・システム、ソフトウェアのインストール時に行った選択によって異なります。また、Adaptive Server Anywhere 製品を単独で入手した場合とほかの製品の一部としてインストールした場合とでも、インストールされるコンポーネントの種類は異なります。

たとえば、ほかの製品の一部として Adaptive Server Anywhere を入手した場合は、2 つのデータベース・サーバのどちらかのバージョンが付属していないことがあります。同様に、すべてのオペレーティング・システムですべてのコンポーネントが使用できるわけではありません。NetWare で使用できるのは、ネットワーク・サーバだけで、パーソナル・サーバは使用できません。

Sybase Central

Sybase Central は、データベースとデータベース・オブジェクトの作成と修正、データベース構造の検査、およびデータベース・サーバ管理を行うためのグラフィカル・ユーザ・インタフェースを提供します。Sybase Central を使用すると、新しいデータベースの作成、テーブルの追加、カラムのテーブルへの追加などの作業を実行できます。

Interactive SQL

Interactive SQL は、SQL 文を入力してデータベースに送信するためのアプリケーションです。Interactive SQL を使うと、データベースへの問い合わせ、データベースのデータの変更、さらにデータベース構造の変更もできます。Sybase Central で可能な操作はすべて Interactive SQL でもできますが、管理作業は Sybase Central で行うほうが簡単です。

ユーティリティ

データベースのバックアップなどの管理タスクを実行するために、一連のユーティリティを使用できます。ユーティリティは、繰り返し使用できるようにバッチ・ファイルに組み込むと便利です。

管理ユーティリティの詳細については、『ASA データベース管理ガイド』> 「データベース管理ユーティリティ」を参照してください。

Ultra Light の概要

ハンドヘルド・コンピュータ、ポケベル、携帯電話などの小型のコンピューティング・デバイスの出現により、Adaptive Server Anywhere よりもメモリ消費の少ないデータベースへの需要が高まってきました。リレーショナル・データベース・エンジンの一部の機能を省略するのは一般的な方法です。しかし、iAnywhere は今までの経験から、アプリケーションや顧客によって、それぞれ必要な機能も異なると考えました。さらに、簡易バージョンを作ると、顧客は必然的に 2 つの異なったデータベースの操作を覚える必要があります。

そのため、iAnywhere では、参照データベースとアプリケーション・ソース・コードを使用して、個々のアプリケーションが使用する Adaptive Server Anywhere の機能だけを含むリレーショナル・データベース・エンジンを作成する新しいテクノロジとして、Ultra Light を開発しました。高速で実行するために、各クエリは完全なアクセス・プランと共に格納され、必要なタスクだけを実行するのに必要なコードが Ultra Light データベース・エンジンに組み込まれます。それぞれの Ultra Light データベース・エンジンは異なっていますが、そのほとんどはわずか数 10 キロバイトで、ポケベルくらいの大きさのデバイスで簡単に実行できます。

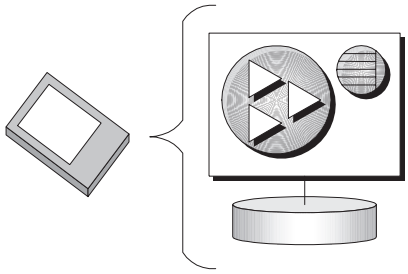
Ultra Light アプリケーションを構築すると、Adaptive Server Anywhere は参照データベースとして機能します。SQL 文、データ型などは Adaptive Server Anywhere とまったく同じです。Ultra Light は Adaptive Server Anywhere の配備テクノロジであり、データベース・システムは同じです。各 Ultra Light データベース・エンジンが実行することになっているタスクは、Adaptive Server Anywhere で実行するのと完全に互換性のある方法で実行されます。

Ultra Light には、Mobile Link 同期テクノロジが組み込まれており、アプリケーションは情報ネットワークにリンクされます。

Ultra Light は、Palm Computing Platform、Windows CE、小型のデバイスで使用される Java などのオペレーティング・システムをサポートしています。

Ultra Light アーキテクチャ

電子手帳のような小さいデバイスでデータベース・アプリケーションを使用する場合、Ultra Light 配備テクノロジを利用できます。Ultra Light では、データベース・サーバとアプリケーションは一体のプロセスであり、データベース・サーバはアプリケーションに固有のものとなります。



この場合、データベースは、ディスク上にファイルとして格納されていない場合もあります。データベースを格納する方法は、配備するプラットフォームによって異なります。

Adaptive Server Anywhere と Ultra Light の比較

この項では、ユーザがニーズに合ったテクノロジを選択できるように、Adaptive Server Anywhere と Ultra Light の相違を説明します。

すでにリソースがわかっている場合は、簡単に Adaptive Server Anywhere か Ultra Light かを選択できます。ただし、両方のテクノロジでサポートされている Windows CE などのプラットフォームで配備する場合、次の点を考慮してください。

Adaptive Server Anywhere

長年、Sybase Adaptive Server Anywhere は、ワークグループ・コンピューティング、モバイル・コンピューティング、埋め込みコンピューティングのニーズに合わせて設計されたリレーショナル・データベース・テクノロジを提供してきました。この製品は、徹底してこの目的に合わせて設計されています。Adaptive Server Anywhere を選択する場合の利点は、次のとおりです。

- Adaptive Server Anywhere はメモリ、CPU、ディスク容量に制限のある環境で効率的に処理を実行するように設計されています。クエリ・オプティマイザやデータ・キャッシュ・メカニズムなどの中核の機能は、特にリソースを大量に消費することなく処理を実行できるように設計されています。同時に、Adaptive Server Anywhere は、多数のユーザのサポート、複数の CPU でのスケーラビリティ、高度な同時実行性機能など、ワークグループ・サーバを利用するのに必要な機能を備えています。主として PC に配備する場合は、Adaptive Server Anywhere が適しています。

- Adaptive Server Anywhere は、プラットフォームを問わないソリューションです。同一のデータベースが、Windows (Windows 95 とその後継製品、Windows NT とその後継製品、Windows CE)、Linux を含む UNIX、Novell NetWare 上で動作します。データベース・ファイルを 1 つのオペレーティング・システムから別のオペレーティング・システムに移動することもできます。

- Adaptive Server Anywhere は、管理を必要としないで動作するように設計されているため、組み込みデータベースとしての使用に適しています。この製品には、セルフ・チューニング・クエリ・オプティマイザ、組み込みスケジュール作成、イベント処理の各機能、自動起動メカニズム、自動停止メカニズムが搭載されています。

- 長年にわたり、多くの顧客と運用経験を積んだことにより、さまざまな機能を実際の稼動環境でテストすることができました。プロジェクトの成功を左右するのは、ストアド・プロシージャ、トリガ、宣言参照整合性、完全なトランザクション処理、リカバリなど、チェックリストにある標準機能だけではありません。あらゆる些細な機能も関係しています。

- SQL Anywhere の同期テクノロジ (SQL Remote と Mobile Link) により、Adaptive Server Anywhere データベースを組織内のインフラストラクチャに統合できます。

- ODBC や OLE DB など、Embedded SQL や Java 以外のインタフェースを使用するときに、対象のプラットフォームが Ultra Light を必要とするほどのリソースの制約がない場合は、Adaptive Server Anywhere が適しています。

Ultra Light：極小サイズのデータベース

Ultra Light は、参照データベースとアプリケーション・ソース・コードを使用して、個々のアプリケーションが使用する Adaptive Server Anywhere の機能だけを含むリレーショナル・データベース・エンジンを作成する新しいテクノロジです。Ultra Light を選択する場合の利点は、次のとおりです。

- Ultra Light データベース・エンジンには、アプリケーションをコンパイルしたときに指定したタスクを実行するのに必要なコードしか組み込まれていません。そのため、Ultra Light データベースでは、動的 SQL を使用してアドホック・クエリを実行できません。

さらに、Ultra Light の各データベースとデータベース・エンジンは、単一のアプリケーションで使用するように設計されています。単一のデータベースで 2 つ以上のアプリケーションを使用する予定がある場合は、Adaptive Server Anywhere を選択してください。

- Ultra Light のプログラミング・モデルでは、プラットフォームに依存しないデータベース・アクセス・コードを使用できます。そのため、基本のデータ・アクセス・レイヤを変更しないで、ユーザ・インタフェースを新しいプラットフォームやデバイスに移植できます。
- Palm Computing Platform などの小型のデバイスで配備する場合は、Ultra Light が適しています。

第 3 章

レプリケーション・テクノロジの概要

この章の内容

データの「**レプリケーション**」とは、物理的にまったく異なるデータベース間でデータを共有することです。1 つのデータベースにある共有データをアプリケーションが変更すると、レプリケーションに設定された他のすべてのデータベースに変更内容が伝達されます。変更内容は、さまざまな手段およびチャネルを通じて伝達できるので、データ整合性を保持すると同時に、柔軟なレプリケーション設定が可能になります。データのレプリケーションは「**同期**」とも呼ばれます。

Sybase には 3 つのレプリケーション・テクノロジがあります。「**Mobile Link**」と「**SQL Remote**」は、統合データベースと多数のリモート・データベース間のレプリケーション用に設計されています。「**Replication Server**」は、比較的少数のデータベース間でほぼリアルタイムのレプリケーションを行うことを目的に設計されています。

概要

この章では、データ・レプリケーションの基本概念について説明します。

データ・レプリケーションの利点

データ可用性

データ・レプリケーション・システムの重要な利点は、ローカルでデータが使用できることであり、費用がかかり、信頼性が低く、時間もかかるような、単一の統合データベースへの接続は行わないということです。中央サーバへの接続手段がなくても、ローカルでデータにアクセスできるので、長距離ネットワーク接続に失敗した場合でも、データを利用できなくなるということがありません。

応答時間

レプリケーションは、次の 2 つの理由で、データ要求に対する応答時間を短縮しています。要求は、広域ネットワークにアクセスせずにローカル・サーバ上で処理されるので、検索速度が向上します。また、ローカル処理によって統合データベース・サーバの作業が軽減されて、プロセッサ競合時間が減少します。

レプリケーション・テクノロジの課題

どのレプリケーション・テクノロジの場合でも、レプリケーションによって柔軟性が高まることと引き換えに、いくつかの問題に対処する必要があります。

トランザクション単位の整合性

レプリケーション・システムの課題の 1 つは、それぞれのデータベースで常にトランザクション単位の整合性が保たれることを保証することです。

Replication Server と SQL Remote では、トランザクション・ログの各単位をレプリケートするときに、レプリケーション中のトランザクションを維持するような方法が使用されます。つまり、トランザクション全体をレプリケートするか、トランザクションをまったくレプリケートしないかのどちらかの方法が使われます。これにより、各データベースでトランザクションの整合性が保証されます。

Mobile Link では、コミットした複数のトランザクションに加えた変更内容が統合されます。これらの変更は、単一のトランザクションでもう一方のデータベースに適用されます。

データの一貫性

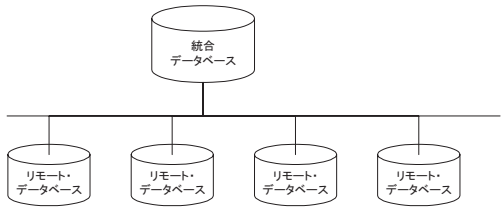
レプリケーション・システムのもう 1 つの課題は、インストール環境全体のデータの一貫性を維持することです。レプリケーション・システムは、全体として「**緩やかな一貫性**」を維持しています。つまり、変更はすべて一貫性が保たれるように各サイトにレプリケートされますが、時間的にはわずかなズレがあるため、ある瞬間だけを見ると、各サイトに存在するデータのコピーが異なる場合もあります。

統合リモート・データベース

Mobile Link と SQL Remote では、「統合データベース」と「リモート・データベース」セット間でデータのレプリケーションが行われます。

「**統合データベース**」は、レプリケートされるデータをすべて含むデータベースです。「**リモート・データベース**」は、統合データベースと同じサイトで、または物理的に離れたサイトで実行される、統合データベースのコピーです。

次の図は、小規模なインストール環境を図で示したものです。



リモート・ユーザ

レプリケーションのインストール環境には多くのリモート・データベースが含まれています。各リモート・データベースには、統合データベースにある情報のサブセットが含まれています。各リモート・データベースは、物理的に離れているデータベースで、通常別のコンピュータ上にあります。すべてのリモート・データベースは、統合データベースと一貫性を保っている必要があります。

すべての共有データのマスタ・コピーが統合データベースで保存されていれば、レプリケーションの設定全体を、1 つの分散データベースとみなすことができます。

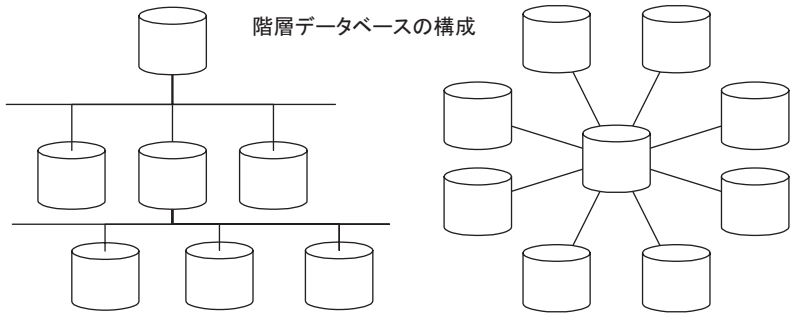
レプリケーションを統合データベースに送信する各リモート・サイトは、統合データベースの「**リモート・ユーザ**」とみなすことができます。リモート・サイトがマルチユーザ・サーバの場合、サイト全体を統合データベースの単一のリモート・ユーザとみなすことができます。

階層データベースの構成

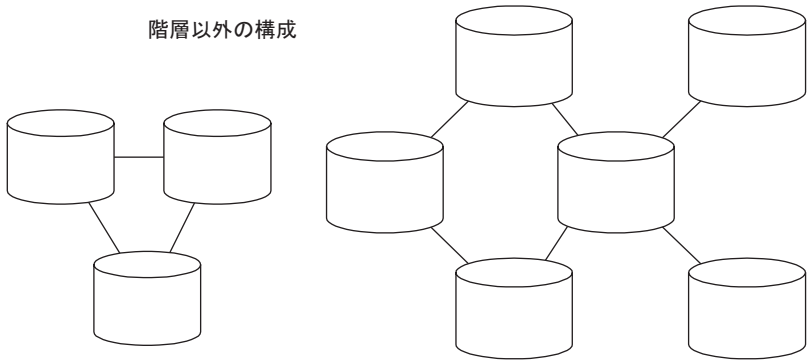
「**階層構成**」のデータベースの場合、すべてのデータベースが 1 つの親データベースを持ちます。ただし、統合データベースには親データベースはありません。

SQL Remote は、データベースの階層構成をサポートしますが、ピア・ツー・ピア型のレプリケーションなど、階層にならない構成はサポートしません。Mobile Link は、通常階層構成で使用されますが、ほかの構成でも使用できます。

階層構成内で、別のデータベースと直接データを共有するデータベースはすべて、常にもう一方のデータベースの上位か下位に位置しています。



階層をとらない構成のデータベースの場合、階層の上下ははっきりと定義されません。



Mobile Link や SQL Remote のインストール環境では、各データベースに、階層内で上にあるデータベースによってレプリケートされたデータのすべてまたはサブセットがあります。

リモート・データベースには、レプリケーションに関係しないかぎり、統合データベースに存在しないテーブルを入れることができます。SQL Remote では、リモート・データベースにあるテーブルとカラムの名前が、統合データベースのそれと一致している必要があります。一方、Mobile Link では、リモート・データベース内で、統合データベースとは異なるカラムとテーブル名でデータを格納することができるので、さらに高い柔軟性が得られます。

双方向レプリケーション

すべての Sybase レプリケーション・テクノロジで、双方向レプリケーションがサポートされています。統合データベースで行われた変更はリモート・データベースに伝達されます。また、リモート・データベースで行われた変更は統合データベースに伝達されてから他のリモート・データベースに伝達されます。Sybase Replication Server では、特定のデータを 1 つのロケーションだけで変更する必要があります。SQL Remote と Mobile Link では、同じデータを同時に複数のロケーションで変更でき、あらゆる競合を解決することができます。

伝達方法

トランザクションによって、あるデータベース上の共有データが変更される場合、そのトランザクション (つまり変更) は、レプリケーションに設定されているほかのデータベースにもレプリケートする必要があります。この作業を実行する方法はいくつかあります。

セッションベースのレプリケーション：Mobile Link

セッションベースのレプリケーション・スキームでは、ある種のダイレクト通信リンクを介してリアルタイムで同期が発生します。たとえば、モデム、ネットワーク、ワイヤレス・モデムを介して接続します。リモート・サイトは、分、時間、日、または週単位の間隔で接続します。

セッションベースの同期プロセスは、未解決の課題をお互いに電話で会話をしながら解決していくプロセスに似ています。プロセスは特定のフォーマットに従います。まず、Mobile Link リモート・サイトは、Mobile Link 同期サーバへの接続をオープンして、前回の同期以降リモート・データベースに行われたすべての変更が含まれる完全なリストをアップロードします。このデータを受信すると、サーバは統合データベースを更新してから、関連する変更をすべて返信します。リモート・サイトは変更内容をすべて組み込み、確認を返信して接続を終了します。

メッセージベースのレプリケーション：SQL Remote

SQL Remote では、「**メッセージ**」を使用してデータベース間でデータを交換します。メッセージは、通常特定のディレクトリに置かれたファイル、または特別なフォーマットの電子メール・メッセージです。各データベースに付属の「**Message Agent**」は、データベース内のデータ変更に関するメッセージを送信します。また、同じ Message Agent が、ほかのデータベースからメッセージを受信したり、受信したメッセージの内容に応じてデータベースを変更したりもします。このシステムにより、電子メールや定期的なダイヤルアップ・リンクによるメッセージベースの接続を随時行うだけで、直接接続されていないデータベース間のレプリケーションが可能です。

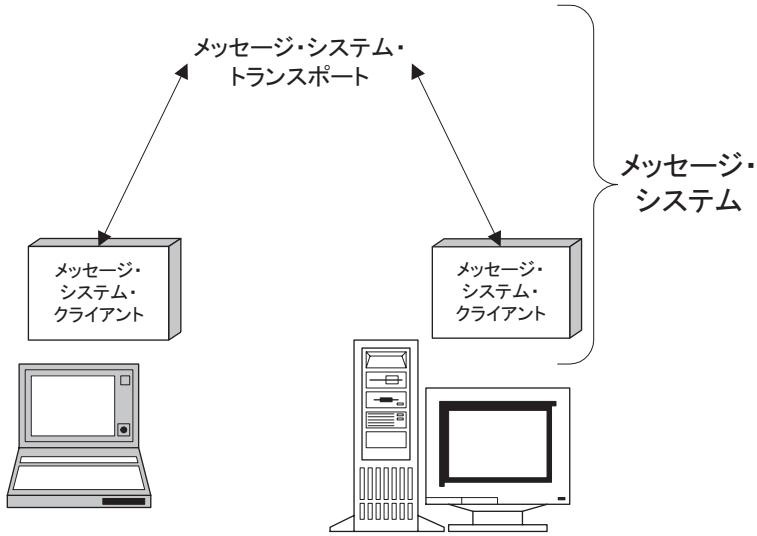
メッセージベースの通信では、各メッセージによって宛先アドレスやその他の制御情報が伝達されます。したがって、情報交換を行うアプリケーション間を直接接続する必要がありません。たとえば、電子メールのメッセージには宛先アドレスが含まれているだけで、送信側のサーバと受信側は直接接続されていません。

蓄積転送方式を使用したメッセージ・サービス

セッションベースのクライアント／サーバ型アプリケーションは、TCP/IP や Novell NetWare の SPX などのネットワーク通信プロトコル・スタックに依存します。これと同じように、メッセージベースのアプリケーションは、SMTP (インターネットの Simple Mail Transfer Protocol)、MAPI (Microsoft の Messaging API)、VIM (Lotus の Vendor Independent Messaging) や単純な共有ファイル・リンクなどのメッセージ・サービスに依存します。

メッセージ・サービスは、「蓄積転送」方式を使用して各メッセージをそれぞれの宛先へ送信します。たとえば、電子メール・システムは、受信者がメール・フォルダを開けてそのメールを読むまでそのメールを蓄積し、フォルダを開けたときに、そのメッセージを転送します。

メッセージ・システムの最上部にレプリケーション・システムを構築すると、SQL Remote のようなメッセージベースのレプリケーション・システムに蓄積転送システムを実装して、メッセージを送信する必要がなくなります。セッションベースのクライアント／サーバ・アプリケーションが、クライアント／サーバ間での情報の受け渡しに独自のプロトコル・スタックを実装しないのと同様に、SQL Remote では既存のメッセージ・システムを使用してメッセージの受け渡しを行います。



配信の保証

作業を確実に行うために、メッセージベースのレプリケーション・システムは、すべてのメッセージが確実に宛先に到着することだけでなく、送信順にメッセージが適用されることも保証する必要があります。SQL Remote には、レプリケーションの更新を正しい順序で確実に適用するためのプロトコルが組み込まれています。

接続ベースのレプリケーション：Replication Server

レプリケーション・テクノロジの中には、データベース間の連続的 (または少なくともほぼ連続的) な接続に依存するものがあります。この接続を介して、2 つのデータベースは連続的に対話を行います。この種のシステムは、変更のレプリケートを迅速に行う点で優れています。特に、十分なリソースとチャネル処理能力が与えられている場合、レプリケーションは僅か数秒程度のタイムラグで確実に行われます。

Replication Server はほぼリアルタイムのレプリケーション・システムであり、主に少数規模のデータベース間でのレプリケーションを目的に設計されています。Replication Server は通常、高速で信頼性が高い

連続的な接続において使用されます。このサーバには蓄積転送技術が採用されており、接続を切断した後に再度接続すると、レプリケーションを自動的に継続します。

この種のシステムの欠点は、信頼性が高く連続的な接続が可能な反面、管理にコストがかかる点です。こうした欠点があるため、接続ベースのテクノロジが適しているのは、固定した 2 台の大規模なデータベースの間でレプリケーションを行うような場合に限定されます。リモート・マシンがモバイル・マシンの場合や、時々しか接続されないような場合は、メッセージベースやセッションベースのテクノロジの方が柔軟性の高いソリューションとなります。

Sybase のレプリケーション・テクノロジ

Sybase は、次の 3 つのレプリケーション・テクノロジを提供します。

- **Mobile Link**　中央の統合データベースと多数のリモート・データベース間でデータの双方向レプリケーションを行うことを目的としたセッションベースのテクノロジです。Sybase 以外のデータベースを含む、各種の統合データベース・サーバをサポートします。リモート・サイトにおける管理およびリソースの要件は最小限に抑えられているので、あらゆるモバイル・アプリケーションに最適です。各同期セッションの終了時に、データベースは一貫性を保っています。

- **SQL Remote**　トランザクションの双方向レプリケーションを目的としたメッセージベースのテクノロジです。統合データ・サーバや多数のリモート・データベースを含む双方向レプリケーション用に設計されています。リモート・サイトにおける管理およびリソースの要件を最小限に抑え、モバイル・データベースに最適です。このシステムは、メッセージベースです。設定にもよりますが、統合データベースとリモート・データベースの間のタイムラグは通常、秒、分、時間単位です。

- **Replication Server**　トランザクションの双方向レプリケーションを行う接続ベースのテクノロジです。高速ネットワークで接続された少数の企業データベース間でレプリケーションを行う場合に適しています。通常は、各サイトに管理者がいます。このような設定では、タイムラグを数秒程度に抑えることが可能です。

レプリケーション・テクノロジの選択

Sybase の各レプリケーション・テクノロジは、特定のアプリケーションに向いています。以下の説明によって各テクノロジの違いが理解できるので、自分が利用するのに最適なテクノロジを選択できます。

以下の要素のうち、どれが自分のアプリケーションに重要かを検討してください。

統合データベース・システム

通常のレプリケーション環境では、大規模なデータベースは情報の中央レポジトリとして働きます。ニーズに応じたデータベース・システムを選べる場合もありますが、すでに中央データベースがあり、それと連携するようにレプリケーション・システムを調整しなければならない場合もあります。

Adaptive Server Anywhere、Sybase Adaptive Server Enterprise、Oracle、Microsoft SQL Server、IBM DB2 など、一般的に使われている多くのデータベース・サーバで、Mobile Link を使用できます。

SQL Remote のインストール環境では、Adaptive Server Anywhere または Sybase Adaptive Server Enterprise を統合データベースとして使用してください。

リモート・データベース・システム

また、Sybase レプリケーション・テクノロジでは、サポート可能なリモート・データベースの種類が異なります。Mobile Link の場合、Adaptive Server Anywhere または Ultra Light のどちらかをリモート・データベースとして使用できます。

SQL Remote の場合、Adaptive Server Anywhere のリモート・データベースだけをサポートします。

ネットワークの特徴

Mobile Link と SQL Remote は両方とも随時接続を行う環境に適しています。すなわち、リモート・サイトが独立して数時間または数日間稼働する必要がある環境です。しかし、ネットワークに接続できればもっと頻繁に同期することもできます。一方、Replication Server は、連続的な接続によって大容量データを迅速にレプリケートできるように設計されたものです。

Mobile Link はセッションベースです。同期中は、リアルタイム接続が必要です。この接続が中断されて同期が完了しなかった場合は、次の同期までプロセスは完了しません。しかし、SQL Remote では、非同期で送受信できるメッセージを通じて情報をリレーします。メッセージは、ファイルとしてハード・ディスク上に残すことも、電子メール・メッセージとして処理することもできます。これらのメッセージは受信されるといつでも処理されるので、レプリケーションが頻繁に発生するようになります。

同期の頻度

状況によっては、情報を即座にレプリケートする必要があります。早急な処理が必要ない場合、レプリケーションは 1 日に 1 回か 2 回で十分です。実際、ネットワーク接続ができなければ、それ以上頻繁にはレプリケーションはできません。

Mobile Link と SQL Remote はどちらも、数時間または数日に 1 回程度の頻度でレプリケーションが発生する状況を基本的に想定して設計されています。Mobile Link と SQL Remote は、同期の頻度が高くてリソースとネットワークの要件が厳しい状況にも対処できます。しかし、十分なリソースがあっても、Mobile Link の同期は数分に 1 回しか発生しません。継続モードで実行していれば、SQL Remote では数秒ごとにレプリケーションが発生します。

Replication Server は、リアルタイムに近いレプリケーションを要求する設定ができるように設計されています。

リモート・サイトの数

リモート・ユーザの数が非常に多い場合は、Mobile Link または SQL Remote が最適です。SQL Remote はメッセージベースで設計されているので、通常のインストール環境であれば数千人のリモート・ユーザを処理できます。Mobile Link のスケーラビリティは、統合データベース管理システムのスケーラビリティによる制限しか受けません。Replication Server は、少数のサイト向けに設計されています。

前述のサイト数は一応の目安なので、必ずしもこの数をすべてのシステムにおいて、リモート・サイトの最大数として制限する必要はありません。実際のサイト数は、レプリケートされる情報の量、同期の頻度、アプリケーションの設計内容によって異なります。

トランザクションの順序

SQL Remote は、1 トランザクションごとに、トランザクション・ログのスキャンやメッセージの準備を適切に行うことによって、データをレプリケートします。それらのメッセージは順序付けされ、リモート・サイトまたは統合サイトに送信されます。SQL Remote が受信メッセージを処理する場合、別のデータベースに適用されたメッセージ順と常に同じ順序で処理します。必要な場合は、先に送信したメッセージがすべて適用されるまで、メッセージの処理を自動的に遅らせます。

一方、Mobile Link は、リモート・データベースで行われた複数のトランザクションの結果を 1 セットの変更データにまとめて、それを統合データベースに適用します。同期は常にトランザクション境界で発生するので、参照の整合性は保たれます。個々の変更の順序付けは、コンポーネント・トランザクションが保存されないうちに行われます。しかし、コミットされていないデータは絶対に同期されないので、データの整合性は保たれます。

特定時点における データの一貫性の実現

Mobile Link の各同期セッションの直後、2 つのデータベースのデータは一貫性が保たれています。リモート・サイトの特定時点におけるデータの一貫性を保証する機能が、Mobile Link のセッションベースでのレプリケーションの長所です。特定の時点 (たとえば午前 10 時) において、リモート・サイトのデータが統合データベースのデータを正確に反映している必要がある場合は、10 時直前に同期するように設定することによって目標が達成されます。同期が正しく行われているかぎり、リモート・サイトにおけるデータの有効性は保証されます。

メッセージの交換によってデータへの変更がレプリケートされる場合、特定リモート・サイトのデータが統合サイトのデータと完全に一貫性を保っていることを任意の時点で保証するのは困難です。たとえば、送信中にメッセージが失われることがあります。SQL Remote は、この障害を自動的に認識してメッセージを再送しますが、このような割り込みによって予想外の遅延が生じます。

レプリケーション・テクノロジの持つ特性の概要

次の表は、各レプリケーション・テクノロジの特性をまとめたものです。表中のそれぞれの特性については、次の項で詳しく説明します。

レプリケーション・テクノロジ	データベースの数	接続	頻度	容量	データベースのタイプ
Mobile Link	多い	随時	中	中	異機種
SQL Remote	多い	随時	低	中	同機種
Replication Server	少ない	常時	低	高	異機種

Mobile Link の特徴

Mobile Link は、レプリケーションのインストール用に設計されたもので、以下の要件があります。

- **多数のデータベース**　Mobile Link は、多数のリモート・データベースをサポートするように設計されています。1 回のインストールで、数千のリモート・データベースをサポートできます。

- **随時接続**　Mobile Link は、サーバが動作しているネットワークに時々または間接的に接続されるデータベースをサポートします。Mobile Link のスケーラビリティは、統合データベース管理システムのスケーラビリティによる制限しか受けません。
- **柔軟な同期スケジュール**　通常、アプリケーションは、分、時間、または日単位の間隔で接続と同期を行います。
- **容量：低～中**　リモート・サイト用のダウンロード情報は、一度に 1 つのリモート・サイトに対して準備されます。Mobile Link システム内のデータが大容量である場合は、接続時間が長くなります。これは、リモート・サイトは同期が完了するまで接続を終了できないためです。
- **異機種データベース**　Mobile Link は、統合データベースとして広く普及しているリレーショナル・データベース・システムを数多くサポートしています。スクリプトを作成して同期プロセスを制御するため、リモート・サイトのスキーマは統合データベースのスキーマと異なる場合があります。

SQL Remote の特徴

SQL Remote は、レプリケーションのインストール用に設計されたもので、以下の要件があります。

- **多数のデータベース**　SQL Remote は、多数のリモート・データベースをサポートするように設計されています。多数のリモート・サイトへのメッセージを同時に準備できるので、1 回のインストールで、数千のリモート・データベースをサポートできます。
- **随時接続**　SQL Remote は、サーバが動作しているネットワークに時々または間接的に接続されるデータベースをサポートします。
- **遅延時間：短～長**　遅延時間が長いというのは、インストールにおいて、あるデータベースにデータが入力されてからそのデータが各データベースにレプリケートされるまでのタイムラグが長いということです。SQL Remote の場合、通常のレプリケーション・メッセージは、秒、分、時間、または日単位の間隔で送信されます。

- **容量：低～中**　レプリケーション・メッセージは随時配信されるので、各リモート・サイトのトランザクションの容量が大きい場合は、メッセージの容量が大きくなります。SQL Remote は、1 つのリモート・データベースについてのレプリケーション・データが比較的低容量であるシステムに最適です。一方、統合サイトにおいては、メッセージを同時に複数のサイトに準備することで、SQL Remote は効率的にメッセージを準備します。

- **同機種データベース**　SQL Remote は、Adaptive Server Enterprise と Adaptive Server Anywhere のデータベースをサポートします。システム内の各データベースは、同様なスキーマを持つ必要があります。

Replication Server の特徴

Replication Server は、レプリケーションのインストール用に設計されたもので、以下の要件があります。

- **少数のデータベース**　Replication Server は、サーバ間のレプリケーションをサポートするように設計されています。通常、1 つのインストール環境で処理できるサーバは 100 未満です。

- **連続接続**　プライマリ・サイトとレプリケート・サイト間の接続に、広域ネットワークを使用する場合があります。しかし、Replication Server は、インストール環境内のサーバ間にデータ交換用のほぼ連続的な接続パスがあるという想定で設計されています。

- **遅延時間：短**　遅延時間が短いということは、インストールにおいて、あるデータベースにデータが入力されてからそのデータが各データベースにレプリケートされるまでのタイムラグが短いということです。Replication Server の場合、通常のレプリケーション・メッセージは、プライマリ・サイトに入力が行われる数秒の間に送信されます。

- **容量：大**　ほぼ連続的に接続が行われておりパフォーマンスに優れている場合、Replication Server は大容量のメッセージを処理できます。

- **異機種データベース**　Replication Server は、主要な DBMS をいくつかサポートしており、レプリケーション中にオブジェクト名のマッピングができます。このため、異機種データベースがサポートされます。

第 2 部　サポート対象のプラットフォーム

第 2 部では、SQL Anywhere Studio がサポートしているプラットフォームについて説明します。また、プラットフォーム固有の情報についても説明します。

第 4 章

SQL Anywhere Studio for Windows CE

この章の内容

この章では、SQL Anywhere Studio を Windows CE プラットフォームで使用する場合の固有の情報について説明します。

ここに記載した情報は、他のマニュアルで説明している内容を補足するのが目的です。

Windows CE で実行するセキュリティとデータベースについては、『SQL Anywhere Studio セキュリティ・ガイド』> 「Windows CE データベースの保護」を参照してください。

はじめに

この項では、Windows CE デバイスに SQL Anywhere Studio をセットアップするのに必要なタスクについて説明します。これらのタスクを完了すると、SQL Anywhere Studio for Windows CE の機能を使えるようになります。

Windows CE デバイスへの SQL Anywhere Studio のインストール

稼働条件

- SQL Anywhere Studio for Windows CE は、サポートされている Microsoft Windows オペレーティング・システムを実行しているマシンからインストールする必要があります。

  Adaptive Server Anywhere がサポートしている Microsoft Windows オペレーティング・システムについては、「オペレーティング・システムのバージョン」149 ページを参照してください。

- デスクトップ・マシンに Windows CE デバイス用 ActiveSync をインストールする必要があります。

Windows CE ファイルのロケーション

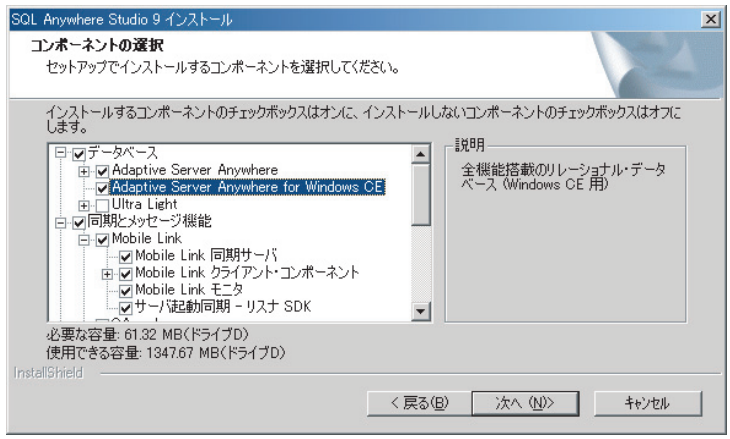
Windows CE では、ファイルはすべてインストール・ディレクトリの *¥Program Files¥Sybase¥ASA9* にインストールされます。サブディレクトリは作成されません。ただし、DLL はすべて *¥Windows* ディレクトリにインストールされます。

❖ **SQL Anywhere Studio for Windows CE をインストールするには、次の手順に従います。**

1. サポートされている Microsoft Windows オペレーティング・システムを実行しているコンピュータに、Windows CE デバイスを接続します。
2. コンピュータで SQL Anywhere Studio *setup.exe* を実行して、インストールを開始します。
3. SQL Anywhere Studio 9 インストール・ウィザードの指示に従います。

4　[コンポーネントの選択] ページで、[データベース] の下にある [Adaptive Server Anywhere for Windows CE] チェックボックスをオンにします。

すでに Adaptive Server Anywhere がコンピュータにインストールされている場合は、[データベース] の下の [Adaptive Server Anywhere] チェックボックスをオフにして、コンピュータに再インストールされるのを防ぎます。



5　ウィザードの残りの指示に従います。

Windows CE デバイスに接続するためのコンピュータの設定

この項では、Windows CE デバイスで実行している Adaptive Server Anywhere のデータベースに接続するためのコンピュータの設定方法を 2 通り説明します。この設定によって、コンピュータの管理ユーティリティから Windows CE のデータベースを管理できるようになります。

Windows CE の管理の詳細については、「管理ユーティリティの使用」87 ページを参照してください。

稼働条件

- Microsoft ActiveSync 3.5 以降

- Adaptive Server Anywhere がサポートしている Windows CE デバイス

  Adaptive Server Anywhere がサポートしている Windows CE デバイスについては、「オペレーティング・システムのバージョン」149 ページを参照してください。

- Windows CE デバイスにインストールした Adaptive Server Anywhere for Windows CE

- サポートされている Microsoft Windows オペレーティング・システムを実行しているコンピュータ

Windows CE デバイスの IP アドレスの特定

Sybase Central や Interactive SQL など、デスクトップ・マシンで実行中のアプリケーションから、Windows CE デバイスで実行中のデータベース・サーバに接続できます。デスクトップ・マシンと Windows CE デバイスの間の ActiveSync リンクを介した接続では、TCP/IP を使用します。

デスクトップ・マシンと Windows CE デバイスを Ethernet で接続している場合は、次の手順を参照してください。ActiveSync が USB 接続を使用している場合は、デスクトップから Windows CE デバイスへの接続はサポートされていません。

シリアル・ケーブル接続と ActiveSync 3.x の使用については、『ASA データベース管理ガイド』> 「ActiveSync 3.0 とシリアル・ケーブルの使用」を参照してください。

❖ Windows CE デバイスの IP アドレスを特定するには、次の手順に従います。

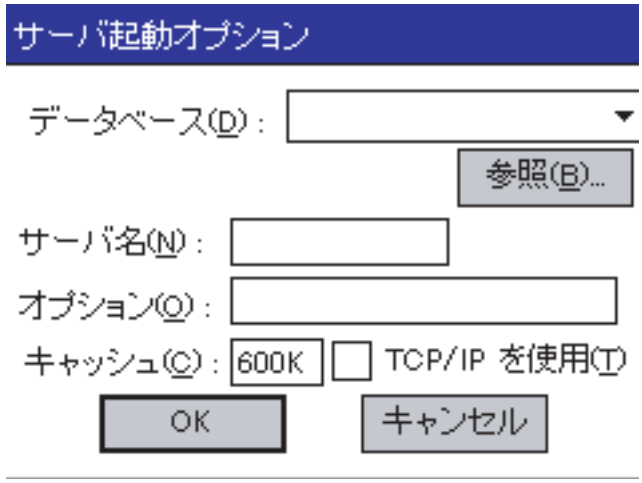
1. Windows CE デバイスで、[ファイル エクスプローラ] を開きます。

   [スタート] メニューから、[プログラム] ー [ファイル エクスプローラ] の順にタップします。

2　[Program Files] － [Sybase] － [ASA9] の順にタップして、Adaptive Server Anywhere ディレクトリに移動します。

3　*dbsrv9* アイコンをタップします。

[サーバ起動オプション] ダイアログが表示されます。



4　開始するデータベース・ファイルを指定します。

「**/Program Files/Sybase/ASA9/asademo.db**」と入力して、デバイスで実行中のサンプル・データベースを開始します。

5　[TCP/IP を使用] チェックボックスをオンにします。

デスクトップ・マシンから Windows CE デバイスで実行中のデータベースに接続するには、TCP/IP 接続が必要です。

6　[サーバ起動オプション] ダイアログの [オプション] フィールドに、「**-z**」と入力します。

サーバは -z オプションを使用して、起動中に IP アドレスを書き出します。Windows CE デバイスをネットワークから切断して再接続すると、アドレスが変わることがあります。

詳細については、『ASA データベース管理ガイド』> 「-z サーバ・オプション」を参照してください。

7 [OK] をタップして、ネットワーク・サーバで実行中のサンプル・データベースを開始します。

8 デバイスの [Today] 画面に移動します。

9 画面の右下にあるサーバ・アイコンをタップします。

サーバ・メッセージ・ウィンドウが表示されます。

10 サーバの IP アドレスを特定します。

IP アドレスはサーバ・メッセージ・ウィンドウに表示されます。IP アドレスが見えていない場合は、サーバ・メッセージ・ウィンドウの端にあるスクロール・バーを使って画面をスクロールしてください。

これで、ODBC データ・ソースを作成して、デスクトップ・マシンから Windows CE デバイスに接続できます。

詳細については、「Windows CE デバイスに接続するための ODBC データ・ソースの作成」52 ページを参照してください。

レジストリの編集

警告
レジストリの変更は危険です。ユーザ自身の責任で変更してください。レジストリを変更する前に、システムのバックアップをとることをおすすめします。

❖ デスクトップ・マシンを設定するには、次の手順に従います。

1 デスクトップ・マシンと Windows CE デバイスで実行中の Adaptive Server Anywhere データベース・サーバを停止します。

2　Windows CE デバイスをクレードルから外します。

3　デスクトップ・マシンでレジストリ・エディタを開きます。

コマンド・プロンプトで、「**regedit**」と入力します。

4　レジストリ・エディタで次のキーを開きます。
HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows CE Services¥ProxyPorts.

5　[編集] メニューから、[新規] － [DWORD 値] を選択します。

名前を「**ASA**」と入力します。

6　先ほど作成した ASA キーをダブルクリックします。

7　10 進値の「2638」と入力します。

これは、Adaptive Server Anywhere 用のデフォルトの TCP/IP ポートの値です。

注意
Windows CE デバイスをクレードルに置くたびに、ActiveSync がポート 2638 のトラフィックをデバイスに転送します。したがって、Windows CE デバイスがクレードルにある間にデスクトップ・マシンでサーバを起動すると、デフォルト・ポートの 2638 を使用できません。これが問題になる場合は、Windows CE トラフィック専用に別のポートを選択することもできます。

8　[OK] をクリックします。

これで、ODBC データ・ソースを作成して、デスクトップ・マシンから Windows CE デバイスに接続できます。

詳細については、「Windows CE デバイスに接続するための ODBC データ・ソースの作成」52 ページを参照してください。

Windows CE デバイスに接続するための ODBC データ・ソースの作成

「**ODBC (Open Database Connectivity)**」インタフェースは Microsoft Corporation によって定義され、Windows 環境でクライアント・アプリケーションからデータベース管理システムに接続するときに使用する標準インタフェースです。接続は、接続パラメータを指定して行います。通常は、複数の接続パラメータをまとめて「**ODBC データ・ソース**」に格納しておくと便利です。ODBC データ・ソースは、繰り返し使用する接続パラメータを保存するための便利な方法です。

詳細については、『ASA データベース管理ガイド』> 「ODBC データ・ソースの使用」を参照してください。

この項では、Windows デスクトップ・マシンで ODBC データ・ソースを作成して、Windows CE デバイスで実行中のデータベースに接続する方法を説明します。

❖ ODBC データ・ソースを作成して Windows CE デバイスに接続するには、次の手順に従います。

1. デスクトップ・マシンで、ODBC アドミニストレータを開きます。

   [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [Adaptive Server Anywhere] － [ODBC アドミニストレータ] を選択します。

   [ODBC アドミニストレータ] が表示されます。

2. [ユーザ DSN] タブで、[追加] をクリックします。

   [データ・ソースの新規作成] ダイアログが表示されます。

3. [Adaptive Server Anywhere 9.0] を選択して、[完了] をクリックします。

   [ODBC 設定] ダイアログが表示されます。

4. [ODBC] タブの [データ・ソース名] フィールドに、データ・ソースの名前を入力します。

たとえば、「**CEdevice**」と入力します。

5　[ログイン] タブで [ユーザ ID とパスワードの指定] を選択し、[ユーザ ID] と [パスワード] フィールドは空白のままにします。

データベースに接続するたびに、ログイン・ユーザ ID とパスワードを指定する必要があります。

ヒント
データベースに接続する場合、デフォルトのユーザ ID は **DBA** になります。デフォルトのパスワードは、**SQL** です。

6　[データベース] タブで、[サーバ名] フィールドを空白のままにします。

デスクトップ・マシンから接続するたびに、サーバ名を指定する必要があります。この名前は、Windows CE デバイスのサーバ・メッセージ・ウィンドウのタイトル・バーに表示されます。

7　[ネットワーク] タブで、[TCP/IP] チェックボックスをオンにします。

8　隣のフィールドに接続パラメータを入力します。

たとえば、「**Host=127.0.0.1;DoBroadcast=none**」と入力します。

Host　このパラメータは、Windows CE デバイスが受信する IP アドレスを指定します。

Windows CE デバイスに接続するためにレジストリを編集しなければならなかった場合は、デフォルトの IP アドレス「**127.0.0.1**」を使用します。

詳細については、「レジストリの編集」50 ページを参照してください。

レジストリを編集していない場合は、Windows CE デバイスの IP アドレスを使用します。

詳細については、「Windows CE デバイスの IP アドレスの特定」48 ページを参照してください。

詳細については、『ASA データベース管理ガイド』> 「TCP/IP プロトコルの使用」を参照してください。

DoBroadcast このパラメータは、TCP/IP 接続の確立方法を制御します。

「**DoBroadcast=none**」と指定すると、指定されているポートと直接 TCP/IP 接続が確立されます。Windows CE デバイスに接続するためにレジストリを編集しなければならなかった場合は、この設定を使用します。

詳細については、「レジストリの編集」50 ページを参照してください。

「**DoBroadcast=DIRECT**」と指定した場合は、データベース・サーバを検索するときにローカル・サブネットへのブロードキャストは実行されません。ホストの IP アドレスが必要になります。

詳細については、「Windows CE デバイスの IP アドレスの特定」48 ページを参照してください。

詳細については、『ASA データベース管理ガイド』> 「DoBroadcast プロトコル・オプション [DOBROAD]」を参照してください。

9 [OK] をクリックして、データ・ソースを作成します。

これで、作成したデータ・ソースを使って、デスクトップ・マシンから Windows CE デバイスで実行中のデータベースに接続できるようになります。

詳細については、「管理ユーティリティの使用」87 ページを参照してください。

サンプル・アプリケーションの使用法

この項では、SQL Anywhere Studio for Windows CE に付属のサンプル・データベースとサンプル・アプリケーションについて説明します。

サンプル・データベース

サンプル・データベースは、限られた種類のスポーツ衣料品を製造する小企業を想定して作られています。サンプル・データベースに入っている情報は、すべて架空のものです。

詳細については、「サンプル・データベースについて」252 ページを参照してください。

サンプル・アプリケーションが格納されている場所

サンプル・データベースは、「*asademo.db*」というファイルに入っています。このファイルは Windows CE デバイスの *¥Program Files¥Sybase¥ASA9* ディレクトリにあります。

サンプル・データベースにアクセスする方法

SQL Anywhere Studio for Windows CE のインストールには、次のサンプル・アプリケーションが含まれています。

- ASA Server Example
- ADO.NET Sample
- ADOCE Sample
- ESQL Sample
- ODBC Sample

これらのアプリケーションを使用すると、サンプル・アプリケーションにアクセスして、SQL Anywhere Studio for Windows CE の機能を検証できます。

ASA Server Example

ASA Server Example は、事前に設定されたサーバ・オプションと接続パラメータを使用して、ネットワーク・サーバのサンプル・データベースを起動します。

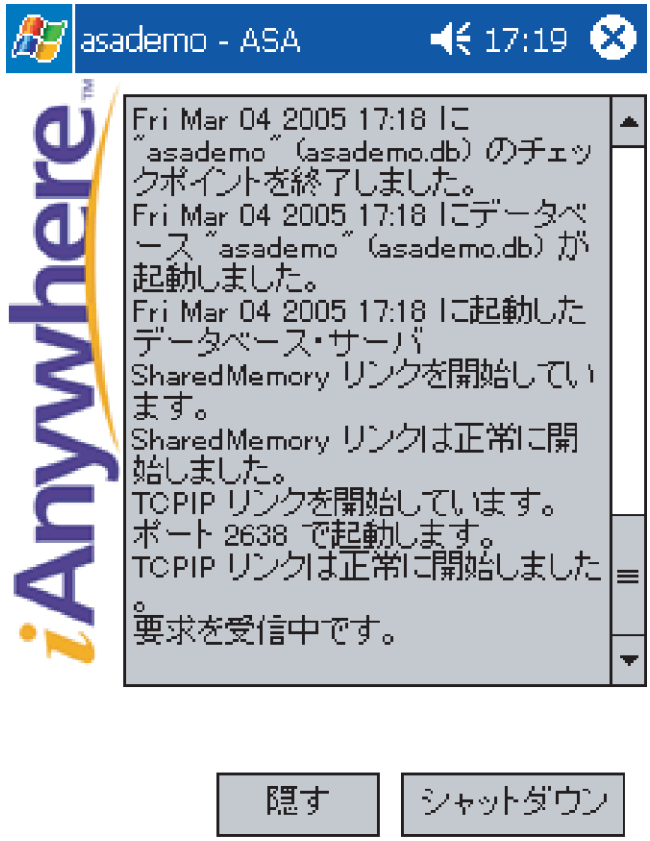
❖ **ASA Server Example を起動するには、次の手順に従います。**

1 Adaptive Server Anywhere のインストール・ディレクトリに移動します。

[スタート] メニューから、[プログラム] － [Sybase] － [ASA9] の順にタップします。

2 ASA ServerExample アイコンをタップして、ネットワークサーバ上のサンプル・データベースを起動します。

ネットワーク・サーバでサンプル・データベースが開始します。開始すると、サーバは Windows CE デバイスの [Today] 画面の右下にアイコンとして表示されます。サーバ・メッセージ・ウィンドウを表示する場合は、このアイコンをタップします。



Windows CE デバイスでデータベース・サーバが実行しています。

これで、デスクトップ・マシンから Windows CE デバイスのサンプル・データベースに接続できます。

詳細については、「管理ユーティリティの使用」87 ページを参照してください。

サンプル・データベースを使い終わったら、サーバを停止してください。

❖ サーバを停止するには、次の手順に従います。

1 [Today] 画面の右下にあるネットワーク・サーバ・アイコンをタップします。

サーバ・メッセージ・ウィンドウが表示されます。

2 [シャットダウン] をタップします。

ADO.NET Sample

ADO.NET Sample を使用するには、デバイスに Microsoft.NET Compact Framework バージョン 1.0 と Service Pack 2 以降をインストールしておく必要があります。

このコンポーネントは、Microsoft Download Center からダウンロードできます。

ADO.NET Sample は、ADO.NET プログラミング・インタフェースを使用する簡単なアプリケーションのデモです。このアプリケーションでは、ネットワーク・サーバーで実行中のサンプル・データベースを起動し、SQL 文でアクセスして、データを変更できます。

このサンプルのソース・コードは、*asa_install_directory¥samples¥asa¥ce_ado_net_sample* にあります。

このプロジェクトは、eMbedded Visual C++ で *asa_install_directory¥samples¥asa¥ce_ado_net_sample* からロードできます。

詳細については、『ASA プログラミング・ガイド』> 「ADO.NET プログラミング・インタフェース」を参照してください。

注意
ADO.NET Sample のユーザ・インタフェースでは、SQL 文を 1 行で入力する必要があります。

❖ ADO.NET Sample を使用するには、次の手順に従います。

1 ADO.NET Sample を起動します。

[スタート] メニューから、[プログラム] ー [Sybase] ー [ASA9] ー [ADO.NET Sample] の順にタップします。

2 [Connect] をタップします。

ネットワーク・サーバでサンプル・データベースが開始します。サーバ・ウィンドウが表示されて、すぐに閉じます。

3 [Exec SQL] をタップして、デフォルトの SQL 文「`SELECT * FROM employee`」を実行してから、[Exec SQL] をタップします。

Employee テーブルのすべてのデータが、データ・ウィンドウに表示されます。

4 データ・ウィンドウの右と下にあるスクロール・バーを使って、employee テーブルのデータ間を移動します。

5 特定のデータ範囲にアクセスします。

「`SELECT emp_id, emp_lname FROM employee`」と入力します。

[Exec SQL] をタップして、SQL 文を実行します。

6 指定したデータ範囲が、データ枠の元のデータを置き換えます。

7 テーブル内のデータを編集します。

「`SELECT * FROM employee ORDER BY emp_id`」と入力し、[Exec SQL] をタップします。

emp_id が 105 の従業員 Matthew Cobb に注目します。

8 「`UPDATE employee SET emp_lname = 'Jones' WHERE emp_lname = 'Cobb'`」と入力します。

9 [Exec SQL] をタップして、SQL 文を実行します。

10 「`SELECT * FROM employee ORDER BY emp_id`」と入力し、[Exec SQL] をタップします。

Matthew の姓が Cobb から Jones に変わりました。

11 変更を元に戻します。

「`UPDATE employee SET emp_lname = 'Cobb' WHERE emp_lname = 'Jones'`」と入力し、[Exec SQL] をタップします。

12 変更が元に戻ったことを確認します。

「`SELECT * FROM employee ORDER BY emp_id`」と入力し、[Exec SQL] をタップします。

Matthew の姓が Jones から元の Cobb に戻りました。

13 別のテーブルのデータにアクセスします。

[Exec SQL] フィールドに、「`SELECT * FROM customer`」と入力します。

[Exec SQL] をタップします。

14 Customer テーブルのすべてのデータがデータ・ウィンドウに表示され、Employee テーブルのデータを置き換えます。

15 [Disconnect] をタップして、ネットワーク・サーバを停止します。

ADO.NET Sample が切断され、ネットワーク・サーバが自動的に停止します。

16 ウィンドウの右上隅の [X] をクリックして、ADO.NET Sample を閉じます。

ADOCE Sample

ADOCE Sample は、Windows CE 開発用 ADO プログラミング・インタフェース、ADOCE を使用する簡単なアプリケーションのデモです。このアプリケーションでは、ネットワーク・サーバーで実行中のサンプル・データベースを起動し、SQL 文を使用してデータにアクセスできます。

このサンプルのソース・コードは、*asa_install_directory¥samples¥asa¥ce_adoce_sample* にあります。

このプロジェクトは、eMbedded Visual C++ で *asa_install_directory¥samples¥asa¥ce_adoce_sample* からロードできます。

ADO プログラミング・インタフェースの詳細については、『ASA プログラミング・ガイド』> 「OLE DB と ADO プログラミング・インタフェース」を参照してください。

注意
ADOCE Sample のユーザ・インタフェースでは、SQL 文を 1 行で入力する必要があります。

❖ ADOCE Sample を使用するには、次の手順に従います。

1. ADOCE Sample を起動します。

   [スタート] メニューから、[プログラム] ー [Sybase] ー [ASA9] ー [ADOCE Sample] の順にタップします。

2. [Connect] をタップします。

   ネットワーク・サーバでサンプル・データベースが開始します。サーバ・ウィンドウが表示されて、すぐに閉じます。

3. [Exec SQL] をタップして、デフォルトの SQL 文「`SELECT * FROM employee`」を実行します。

Employee テーブルのすべてのデータが、データ・ウィンドウに表示されます。

4 データ・ウィンドウの端にあるスクロール・バーを使って、employee テーブルのデータ間を移動します。

5 特定のデータ範囲にアクセスします。

「`SELECT emp_id, emp_lname FROM employee`」と入力します。

[Exec SQL] をタップして、SQL 文を実行します。

6 下方にスクロールしてデータを見ます。データは、データ・ウィンドウにすでに表示されているデータの下にあります。

7 別のテーブルのデータにアクセスします。

[Exec SQL] フィールドに、「`SELECT * FROM customer`」と入力します。

[Exec SQL] をタップします。

8 Customer テーブルから下方にスクロールしてデータを見ます。データは、データ・ウィンドウにすでに表示されているデータの下にあります。

9 [Disconnect] をタップして、ネットワーク・サーバを停止します。

ADOCE Sample が切断され、ネットワーク・サーバが自動的に停止します。

10 ウィンドウの右上隅の [X] をクリックして、ADOCE Sample を閉じます。

ESQL Sample

ESQL Sample は、Embedded SQL プログラミング・インタフェースを使用する簡単なアプリケーションのデモです。このアプリケーションでは、ネットワーク・サーバーで実行中のサンプル・データベースを起動し、SQL 文を使用してデータにアクセスできます。

このサンプルのソース・コードは、
asa_install_directory¥samples¥asa¥ce_esql_sample にあります。

このプロジェクト・ファイルは、eMbedded Visual C++ で
asa_install_directory¥samples¥asa¥ce_esql_sample¥esql_sample.vcw から
ロードできます。

詳細については、『ASA プログラミング・ガイド』> 「Embedded SQL プログラミング・インタフェース」を参照してください。

注意
ESQL Sample のユーザ・インタフェースでは、SQL 文を 1 行で入力する必要があります。

❖ ESQL Sample を使用するには、次の手順に従います。

1. ESQL Sample を起動します。

   [スタート] メニューから、[プログラム] ー [Sybase] ー [ASA9] ー [ESQL Sample] の順にタップします。

2. [Connect] をタップし、デフォルトの接続文字列を使ってサンプル・データベースに接続します。

   ネットワーク・サーバでサンプル・データベースが開始します。サーバ・ウィンドウが表示されて、すぐに閉じます。

3. [Exec SQL] をタップして、デフォルトの SQL 文「`SELECT * FROM employee`」を実行します。

   Employee テーブルのすべてのデータが、データ・ウィンドウに表示されます。

4. データ・ウィンドウの右と下にあるスクロール・バーを使って、employee テーブルのデータ間を移動します。

5. 別のテーブルのデータにアクセスします。

   [Exec SQL] フィールドに「`SELECT * FROM customer`」と入力し、[Exec SQL] をタップします。

Customer テーブルのすべてのデータがデータ・ウィンドウに表示されて、Employee テーブルのデータを置き換えます。

6 [Disconnect] をタップして、ネットワーク・サーバを停止します。

ESQL Sample が切断され、ネットワーク・サーバが自動的に停止します。

7 ウィンドウの右上隅の [X] をクリックして、ESQL Sample を閉じます。

ODBC Sample

ODBC Sample は、ODBC プログラミング・インタフェースを使用する簡単なアプリケーションのデモです。このアプリケーションでは、ネットワーク・サーバーで実行中のサンプル・データベースを起動し、基本的な SQL 文を使用してデータにアクセスできます。

このサンプルのソース・コードは、*asa_install_directory¥samples¥asa¥ce_odbc_sample* にあります。

このプロジェクト・ファイルは、eMbedded Visual C++ で *asa_install_directory¥samples¥asa¥ce_odbc_sample¥odbc_sample.vcw* からロードできます。

詳細については、『ASA プログラミング・ガイド』> 「ODBC プログラミング・インタフェース」を参照してください。

注意
ODBC Sample のユーザ・インタフェースでは、SQL 文を 1 行で入力する必要があります。

❖ ODBC Sample を使用するには、次の手順に従います。

1 ODBC Sample を起動します。

[スタート] メニューから、[プログラム] ー [Sybase] ー [ASA9] ー [ODBC Sample] の順にタップします。

2 [Connect] をタップします。

ネットワーク・サーバでサンプル・データベースが開始します。サーバ・ウィンドウが表示されて、すぐに閉じます。

3 [Exec SQL] をタップして、デフォルトの SQL 文「`SELECT * FROM employee`」を実行します。

Employee テーブルのすべてのデータが、データ・ウィンドウに表示されます。

4 データ・ウィンドウの右と下にあるスクロール・バーを使って、employee テーブルのデータ間を移動します。

5 別のテーブルのデータにアクセスします。

[Exec SQL] フィールドに、「`SELECT * FROM customer`」と入力します。

Customer テーブルのすべてのデータがデータ・ウィンドウに表示されて、Employee テーブルのデータを置き換えます。

6 [Disconnect] をタップして、ネットワーク・サーバを停止します。

ODBC Sample が切断され、ネットワーク・サーバが自動的に停止します。

7 ウィンドウの右上隅の [X] をクリックして、ODBC Sample を閉じます。

Windows CE データベースの設定

この項では、Sybase Central の特殊な機能を使用して、Windows CE のデータベースを設定する方法を説明します。また、既存の Adaptive Server Anywhere データベースを Windows CE デバイスにコピーする方法と、Windows CE のデータベースを消去する方法も説明します。

データベースのプロパティ

Adaptive Server Anywhere の完全バージョンで使用できる SQL 機能のほとんどは、Windows CE バージョンでも使用できます。たとえば、トランザクション処理、参照整合性アクション、プロシージャ、トリガなどです。ただし、Java の機能とリモート・データ・アクセス機能は、Windows CE では使用できません。

Windows CE デバイスで使用するデータベースのプロパティを設定するときは、サポートされていない機能に注意してください。

詳細については、「Windows CE でサポートされていない SQL Anywhere Studio の機能」112 ページを参照してください。

トランザクション・ログ

トランザクション・ログには、データベースに加えられた変更がその変更順に格納されます。データベース・ファイルでメディア障害が発生した場合、トランザクション・ログはデータベースのリカバリに重要な役割を果たします。また、トランザクション・ログを使用すると、より効率的に作業できます。トランザクション・ログは、デフォルトではデータベース・ファイルと同じディレクトリに配置されます。このログは、Windows CE デバイスで初めてデータベースを開始するときに作成されます。

既存のデータベースを Windows CE デバイスにコピーする場合は、データベース・ファイルとトランザクション・ログ・ファイルの両方をコピーできます。トランザクション・ログ・ファイルをデバイスにコピーしなければ、Windows CE デバイスでデータベースを開始したときに新しいログが作成されます。新しいログには、元のトランザクション・ログに格納されていた情報は含まれません。これは、最後にデータベースを使用したときに正しく停止しなかった場合や、データ

ベースを同期している場合に問題になることがあります。したがって、データベース・ファイルとトランザクション・ログ・ファイルの両方を Windows CE デバイスにコピーすることをおすすめします。

詳細については、『ASA データベース管理ガイド』> 「トランザクション・ログ」を参照してください。

Windows CE での jConnect の使用

jConnect は、Adaptive Server Anywhere 用の純正 Java JDBC ドライバです。

Sybase Central には、Java アプリケーションで Adaptive Server Anywhere のデータベースにアクセスできるように、jConnect JDBC ドライバを有効にするオプションがあります。

Windows CE のデータベースを作成すると指定した場合、デフォルトでは jConnect は無効になっていますが、オプションで有効にすることができます。

データベースに jConnect のサポートを追加すると、多数のエントリがシステム・テーブルに追加されます。これによってデータベース・サイズが大きくなり、たとえ jConnect 機能を使用しない場合でも、データベースの実行に必要なメモリが約 200 KB 増えます。

Windows CE のような限られたメモリ環境で作業をする場合、jConnect を使用する必要がなければ、データベースに jConnect のサポートを追加しなくてもよいでしょう。

暗号化

単純暗号化または強力な暗号化を使用して、データベースのセキュリティを確保することができます。データベースを初期化したあとで暗号化の設定を変更するには、データベース全体を再構築するしかありません。

詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「データベースの暗号化」を参照してください。

詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「Windows CE データベースの保護」を参照してください。

その他のプロパティ

以下のプロパティは、データベースの作成中に設定します。いったん設定すると、データベースを再構築しなければ変更できません。

- 大文字と小文字の区別

  詳細については、『ASA SQL ユーザーズ・ガイド』> 「大文字と小文字の区別」を参照してください。

- パスワードの大文字と小文字の区別

  詳細については、『ASA SQL ユーザーズ・ガイド』> 「大文字と小文字の区別」を参照してください。

- 比較時の後続ブランクの処理

  デフォルトでは、後続ブランクを余分の文字と見なしてデータベースが作成されます。たとえば、「Dirk」は「Dirk　」と同じではありません。

  後続ブランクが無視されるように、ブランクを埋め込んだデータベースを作成できます。

- ページ・サイズ

  詳細については、『ASA SQL ユーザーズ・ガイド』> 「テーブルとページのサイズ」を参照してください。

- 照合順

  Windows CE 用のデータベースを作成するときは、Windows が対象の言語に使用するのと同じシングルバイト文字セットまたはマルチバイト文字セットを基にした照合を使用してください。たとえば、英語、フランス語、またはドイツ語を使用する場合は、1252Latin1 照合を使用します。日本語を使用する場合は 932JPN 照合を、韓国語を使用する場合は 949KOR 照合を使用します。

  詳細については、『ASA データベース管理ガイド』> 「照合の知識」を参照してください。

Windows CE データベースの作成

Windows CE 用に Adaptive Server Anywhere データベースを作成する方法は、3 通りあります。Sybase Central の [データベース作成] ウィザードを使用して、Windows CE デバイスに直接コピーできるデータベースを作成します。初期化ユーティリティ (dbinit) を使用することもできます。または、Interactive SQL で CREATE DATABASE 文を使用して、Windows CE デバイスに手動でコピーできるデータベースを作成します。この項では、これらすべての方法について説明します。

Sybase Central を使用した Windows CE データベースの作成

Sybase Central には、Windows CE データベース用に簡単にデータベースを作成する機能があります。Windows デスクトップ・マシンに Windows CE サービスをインストールしてあれば、オプションで Sybase Central を使用して Windows CE データベースを作成できます。Sybase Central は、Windows CE のデータベースに必要な条件を満たし、生成されたデータベース・ファイルを Windows CE デバイスにコピーするオプションを提供します。

❖ Sybase Central で Windows CE のデータベースを作成して、それを Windows CE デバイスに直接コピーするには、次の手順に従います。

1. デスクトップ・マシンで、Sybase Central を開始します。

   [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [Sybase Central] を選択します。

2. [データベース作成] ウィザードを開始します。

   [ツール] メニューから、[Adaptive Server Anywhere 9] － [データベースの作成] を選択します。

3. [このコンピュータにデータベースを作成] を選択し、[次へ] をクリックします。

   データベースは、Windows CE デバイスにコピーされる前にデスクトップ・マシンで初期化されます。

4 データベースを格納するデスクトップ・マインのディレクトリとファイル名を指定し、[次へ] をクリックします。

5 [Windows CE にこのデータベースを作成] を選択し、[次へ] をクリックします。

このオプションを使用すると、新しいデータベースが Windows CE に対応した設定になります。

[次へ] をクリックすると、Windows CE デバイスへの接続が開始されます。次の手順に進むには、Windows CE デバイスがデスクトップ・マシンに接続している必要があります。

6 [データベースを Windows CE デバイスにコピー] を選択すると、デバイスが初期化されたあと、自動的にデータベースがデバイスにコピーされます。

7 データベース・ファイルをコピーする Windows CE のディレクトリを指定します。デフォルトは、メイン・デバイスのディレクトリです。

ヒント

Windows CE デバイスの My Documents ディレクトリにデータベースをコピーすると、データベースの開始が容易になります。

Windows CE デバイスで [サーバ起動オプション] ダイアログを使用してデータベースを開始するときに、ブラウザを使って My Documents ディレクトリ内のデータベース・ファイルを検索するだけで済みます。

データベースを My Documents ディレクトリに格納しない場合は、[サーバ起動オプション] ダイアログの [データベース] フィールドにデータベースのパスを入力する必要があります。

オプションとして、[コピー後にデスクトップのデータベースを削除] を選択できます。

デスクトップのコピーを削除しない場合は、手順 4 で指定したディレクトリにデータベース・ファイルのコピーが格納されます。

[Next] をクリックします。

8　デスクトップ・マシンにデータベースのコピーを保存する場合は、トランザクション・ログ・ファイルを保存するディレクトリを指定できます。

Windows CE デバイスでは、ネットワーク・サーバで初めてデータベースを開始したときに、データベース・ファイルと同じディレクトリにログ・ファイルが作成されます。

[Next] をクリックします。

9　[以下のミラー・ログ・ファイルを保守] チェックボックスをオフにしておきます。

ミラー・ログは、万一トランザクション・ログが破損した場合のデータ・リカバリに役立ちますが、必須ではありません。

10　[jConnect メタ情報サポートをインストール] チェックボックスをオフにします。

このオプションは必須ではありません。オンにすると、データベースの実行に余分のメモリを必要とします。

11　ラジオ・ボタンを使用して、データベースの暗号化レベルを設定します。

強力な暗号化を選択した場合は、暗号化キーを指定する必要があります。キーには最低でも 16 文字の値を選択し、大文字と小文字、数字、文字、特殊文字を組み合わせて使用することをおすすめします。

[警告]
キーのコピーは安全な場所に保管してください。キーは、データベースを起動したり変更したりするたびに必要になります。キーを紛失すると、データベースにまったくアクセスできなくなり、リカバリも不可能になります。

12　ウィザードの残りの指示に従います。

13　[完了] をクリックすると、データベースが作成されてデバイスにコピーされます。

Windows CE デバイスにコピーしているファイルの進行状況を示すダイアログが表示されます。

14 データベースが Windows CE デバイスにコピーされたら、ファイルの場所を確認します。

[スタート] メニューから [プログラム] － [ファイル エクスプローラ] の順にタップして、データベースをコピーした Windows CE ディレクトリに移動します。

データベース・ファイルがリストされます。ログ・ファイルは、Windows CE デバイスで初めてデータベースを開始するときまでリストには表示されません。

dbinit を使用した Windows CE データベースの作成

初期化ユーティリティ (dbinit) を使用してデータベースを作成できます。ただし、このユーティリティから Windows CE デバイスに直接データベースをコピーすることはできません。Windows CE デバイスにデータベースを手動でコピーする必要があります。

詳細については、「Windows CE デバイスへのデータベースの手動コピー」75 ページを参照してください。

❖ dbinit ユーティリティを使用してデータベースを作成するには、次の手順に従います。

1 コマンド・プロンプトで、データベースを格納するディレクトリに移動します。

例：

```
cd temp
```

2 次のコマンドを入力して、データベースを作成します。

```
dbinit database-name.db
```

ヒント
暗号化やページ・サイズなど、データベースのプロパティも dbinit ユーティリティを使用して設定できます。

詳細については、『ASA データベース管理ガイド』> 「dbinit コマンド・ライン・ユーティリティを使用したデータベースの作成」を参照してください。

3　次のコマンドを入力して、データベース・ファイルが正しく作成されたことを確認します。

```
dir
```

作成したデータベース・ファイルがディレクトリに表示されます。

CREATE DATABASE 文を使用した Windows CE データベースの作成

CREATE DATABASE 文を使用して、Interactive SQL でデスクトップ・マシンにデータベースを作成できます。ただし、このアプリケーションから Windows CE デバイスに直接データベースをコピーすることはできません。Windows CE デバイスにデータベースを手動でコピーする必要があります。

詳細については、「Windows CE デバイスへのデータベースの手動コピー」75 ページを参照してください。

❖ CREATE DATABASE 文を使用してデータベースを作成するには、次の手順に従います。

1　デスクトップ・マシンで Interactive SQL を開きます。

[スタート] メニューから、[プログラム] － [Sybase SQL Anywhere 9] － [Adaptive Server Anywhere] － [Interactive SQL] の順に選択します。

2　[接続] ダイアログが表示されます。

[接続] ダイアログが自動的に表示されない場合は、[SQL] メニューから [接続] を選択してください。

3 [ID] タブで、次の値を入力します。

- **ユーザ ID**： DBA
- **パスワード**： SQL

4 [ODBC データ・ソース名] オプションを選択します。

5 [参照] をクリックして、ASA 9.0 Sample データ・ソースを選択します。

6 [OK] をクリックして、サンプル・データベースに接続します。

7 Interactive SQL の [SQL 文] ウィンドウ枠で、次の文を入力します。

```
CREATE DATABASE 'c:¥¥temp¥¥database-name.db'
 TRANSACTION LOG ON
```

ヒント
暗号化やページ・サイズなど、データベースのプロパティも CREATE DATABASE 文を使用して設定できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CREATE DATABASE 文」を参照してください。

8 [SQL] － [実行] を選択します。

デスクトップ・マシンの *c:¥temp* ディレクトリに、データベースとトランザクション・ログが作成されます。

Windows CE デバイスへのデータベースの手動コピー

この項で説明する方法を使用すると、Adaptive Server Anywhere の既存のデータベースを Windows CE デバイスにコピーできます。ただし、Windows CE デバイスにデータベースをコピーした場合、Windows CE でサポートされていないデータベース・プロパティは機能しないことに注意してください。

詳細については、「データベースのプロパティ」66 ページを参照してください。

データベースの作成については、『ASA SQL ユーザーズ・ガイド』>「データベースの作成」を参照してください。

❖ データベースを Windows CE デバイスにコピーするには、次の手順に従います。

1. Windows CE デバイスをデスクトップ・マシンに接続します。
2. デスクトップ・マシンで Windows エクスプローラを開きます。
3. コピーするデータベースが格納されているディレクトリまで参照します。
4. データベース・ファイルを右クリックして、[コピー] を選択します。
5. Windows エクスプローラをもう 1 つ開きます。
6. データベース・ファイルを格納する Windows CE デバイスのディレクトリまで参照します。

ヒント

Windows CE デバイスで [サーバ起動オプション] ダイアログを使用してデータベースを開始するときに、ブラウザを使って My Documents ディレクトリ内のデータベース・ファイルを検索するだけで済みます。

データベースを My Documents ディレクトリに格納していない場合は、[サーバ起動オプション] ダイアログの [データベース] フィールドにデータベースのパスを入力する必要があります。

7 Windows CE デバイスの [Windows エクスプローラ] ウィンドウで空白領域を右クリックして、[貼り付け] を選択します。

ファイルが Windows CE デバイスにコピーされます。

Windows CE データベースの消去

Adaptive Server Anywhere for Windows CE は、[データベース消去] ウィザード、DROP DATABASE 文、消去ユーティリティ (dberase) をサポートしていません。Windows CE デバイスからデータベースを手動で消去する必要があります。

Windows CE からデータベースを消去する方法は、2 通りあります。デバイスのインタフェースを使ってデータベースを消去するか、デバイスをデスクトップ・マシンに接続して Windows エクスプローラを使ってデータベースを消去します。この項では、その両方の方法を説明します。

デバイスのインタフェースを使用したデータベースの消去

❖ デバイスのインタフェースを使用してデータベースを消去するには、次の手順に従います。

1 [スタート] メニューから、[プログラム] ー [ファイル エクスプローラ] の順にタップして、削除するデータベース・ファイルのあるディレクトリに移動します。

2 データベース・ファイルをタップして押したままにします。

ポップアップ・メニューが表示されます。

3 [削除] をタップします。

4 [はい] をタップして、ファイルの削除を確認します。

ヒント
実行中のデータベースは削除できません。

データベースを削除したあと、トランザクション・ログ・ファイルを削除します。データベースを開始したことがない場合は、トランザクション・ログ・ファイルは作成されていません。

Windows エクスプローラを使用したデータベースの消去

❖ **Windows エクスプローラを使用してデータベースを消去するには、次の手順に従います。**

1. Windows CE デバイスをクレードルに置き、ActiveSync を介してデスクトップ・マシンに接続していることを確認します。
2. デスクトップ・マシンで Windows エクスプローラを開きます。
3. データベース・ファイルが格納されている Windows CE ディレクトリまで参照します。
4. データベース・ファイルを右クリックして、[削除] を選択します。

   データベース・ファイルの削除を確認するダイアログが表示されます。[はい] をクリックします。

ヒント
実行中のデータベースは削除できません。

データベースを削除したあと、トランザクション・ログ・ファイルを削除します。データベースを開始したことがない場合は、トランザクション・ログ・ファイルは作成されていません。

Windows CE のデータベース・サーバの使用

この項では、データベース・サーバの概要と、それを Windows CE で開始、停止、設定する方法について説明します。

Windows CE に提供されるデータベース・サーバは、ネットワーク・データベース・サーバで、そのファイル名は *dbsrv9.exe* です。ネットワーク・サーバは、TCP/IP ネットワーク・リンクを介した通信をサポートしています。

通常のクライアント／サーバ構成の場合、データベース・サーバは、クライアント・アプリケーションのマシンより高性能でリソースの多いマシンで稼働します。明らかに、これは Windows CE には当てはまりません。Windows CE では、それほど性能が高くないマシンでデータベース・サーバが稼働します。

Windows CE でネットワーク・サーバを提供すると、デスクトップ・マシンで管理ユーティリティを実行して、Windows CE データベースのタスクを実行できるという利点があります。次に例を示します。

- デスクトップ・マシンの Sybase Central を使用してデータベースを管理できる。
- デスクトップ上の Interactive SQL を使用して、データのロードとアンロード、クエリを実行できる。

詳細については、「管理ユーティリティの使用」87 ページを参照してください。

サーバとデータベースのオプション

サーバとデータベースのオプションは、主に Adaptive Server Anywhere の動作とパフォーマンスのチューニングに使用されます。さまざまなオプションを選択して、キャッシュで使用できるメモリ、ネットワーク・サーバ上のデータベースを起動するのに必要なアクセス権のレベル、使用するネットワーク・プロトコルなどの機能を指定できます。

Windows CE では、オプションは [サーバ起動オプション] ダイアログで設定します。データベース・サーバのオプションをコマンド・ラインで設定する他の Windows オペレーティング・システムとは、この点で異なります。ほとんどのサーバ・オプションは、Windows CE でも使用できます。

データベース・サーバ・オプションの詳細については、『ASA データベース管理ガイド』> 「データベース・サーバ」を参照してください。

サポートされていないオプションについては、「Windows CE でサポートされていないサーバ・オプション」116 ページを参照してください。

チュートリアル：Windows CE 上のデータベース・サーバの使用

このチュートリアルを完了すると、サーバの開始と停止、単一または複数のデータベースの実行など、データベース・サーバに関連する主要タスクを実行できるようになります。

前提条件

- このチュートリアルを開始する前に、「はじめに」46 ページのすべてのタスクを完了してください。

始める前に

チュートリアルで使用する Windows CE のデータベースを 2 つ作成する必要があります。

❖ Windows CE デバイスのデータベースを作成するには、次の手順に従います。

1. Windows CE デバイスが、デスクトップ・マシンに接続していることを確認します。
2. デスクトップ・マシンで Sybase Central を開きます。

   [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [Sybase Central] の順に選択します。

   Sybase Central が表示されます。
3. [データベース作成] ウィザードを開始します。

[ツール] メニューから、[Adaptive Server Anywhere 9] － [データベースの作成] の順に選択します。

[データベース作成] ウィザードが表示されます。

4 *Alpha.db* というデータベースを作成します。

デスクトップのコピーは、Windows CE デバイスにコピーするときに削除されるので、デスクトップ・マシンでそれを格納するディレクトリは無視します。

5 [次へ] をクリックします。

6 [Windows CE にこのデータベースを作成] オプションを選択します。

7 [次へ] をクリックします。

Windows CE デバイスとの接続がテストされます。

8 [データベースを Windows CE デバイスにコピー] オプションを選択します。

[Windows CE ファイル名] フィールドに次のように入力して、Windows CE デバイスの My Documents ディレクトリにデータベースをコピーします。*¥My Documents¥Alpha.db*

9 [コピー後にデスクトップのデータベースを削除] オプションを選択します。

10 [完了] をクリックすると、データベースがデフォルトの設定で作成されます。

データベース・ファイル *Alpha.db* が Windows CE デバイスにコピーされたら、次の手順に進みます。

11 3 ～ 9 の手順を繰り返して、*Beta.db* というデータベースを作成します。

2 番目のデータベースの手順 7 で入力するパスは、*¥My Documents¥Beta.db* です。

データベース・ファイル *Beta.db* が Windows　CE デバイスにコピーされたら、チュートリアルを開始できます。

レッスン 1 ：データベース・サーバを起動する

この項では、Windows CE で単一のデータベースを実行するという簡単な例を紹介します。

❖ サーバでデータベースを開始するには、次の手順に従います。

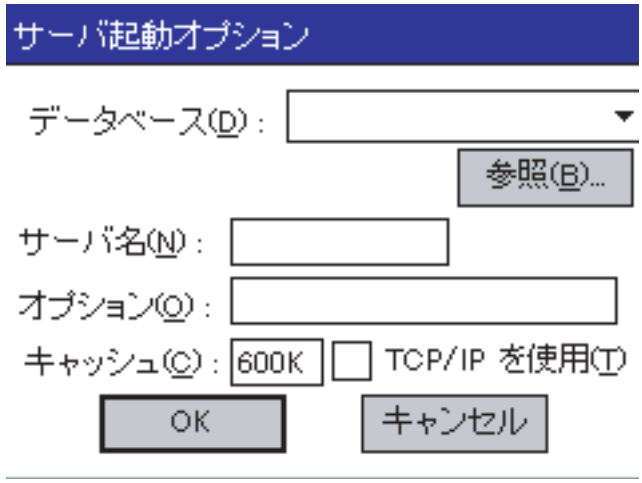
1. Windows　CE デバイスで、[ファイル エクスプローラ] を開きます。

   [スタート] メニューから、[プログラム] － [ファイル エクスプローラ] の順に選択します。

2. [Program Files] － [Sybase] － [ASA9] の順に選択して、Adaptive Server Anywhere ディレクトリに移動します。

3. *dbsrv9* アイコンをタップします。

   [サーバ起動オプション] ダイアログが表示されます。



4 開始するデータベース・ファイルを指定します。

[参照] をタップして、My Documents ディレクトリの *Alpha.db* を見つけます。

5 [サーバ名] フィールドに「**CEserver**」と入力して、ネットワーク・サーバに名前を割り当てます。

6 デフォルトのキャッシュ・サイズ 600kb を使用します。

ヒント
このチュートリアルの目的上、デフォルトのキャッシュ・サイズで十分です。大きいデータベースの場合は、キャッシュ・サイズを大きくするとパフォーマンスが向上します。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「パフォーマンス向上へのキャッシュの使用」を参照してください。

---

7 [TCP/IP を使用] チェックボックスをオンにします。

デスクトップ・マシンから Windows CE デバイスで実行中のデータベースに接続するには、TCP/IP 接続が必要です。あとのレッスンで、デスクトップ・マシンから接続する必要があります。

8 [サーバ起動オプション] ダイアログの [オプション] フィールドに、「**-gd all**」と入力します。

-gd オプションで許可を設定すると、どのユーザでもネットワーク・サーバで追加のデータベースを開始できます。これは、あとのレッスンで必要になります。

詳細については、『ASA データベース管理ガイド』> 「-gd サーバ・オプション」を参照してください。

9 [OK] をタップして、ネットワーク・サーバで実行している Alpha データベースを開始します。

10 デバイスの [Today] 画面に移動します。

11 画面の右下にあるサーバ・アイコンをタップします。

サーバ・メッセージ・ウィンドウが表示されます。

サーバ・メッセージ・ウィンドウに「要求を受信中です。」というメッセージが表示されたら、次のレッスンに進んでください。

次の作業

次は、Windows CE 上のネットワーク・サーバで複数のデータベースを開始する方法を学習します。

レッスン 2 ：サーバで複数のデータベースを開始する

Windows CE では、すでに最初の Adaptive Server Anywhere サーバを実行中に 2 番目のサーバを起動しようとすると、最初のサーバが前面に表示されます。Windows CE のアプリケーションでは、これが標準の動作になります。そのため、Windows CE デバイスでは 2 つのサーバを同時に実行することはできません。複数のサーバを実行する代わりに、1 つのサーバで複数のデータベースを実行できます。

サーバで複数のデータベースを開始するには、まず Windows CE デバイスで実行しているデータベースの 1 つに接続する必要があります。

❖ Sybase Central を使用して Alpha データベースに接続するには、次の手順に従います。

1. デスクトップ・マシンで Sybase Central を開始します。

   [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [Sybase Central] の順に選択します。

2. [ツール] － [接続] を選択します。

   プラグインの選択を求めるメッセージが表示されたら、[Adaptive Server Anywhere 9] を選択して、[OK] をクリックします。

   [接続] ダイアログが表示されます。

3. [ID] タブで、次の値を入力します。

   - **ユーザ ID**：DBA

- **パスワード**：SQL

4 [ODBC データ・ソース名] オプションを選択します。

5 [参照] をクリックし、「Windows CE デバイスに接続するための ODBC データ・ソースの作成」52 ページで作成した **CEdevice** データ・ソースを [データ・ソース名] ダイアログから選択します。

6 [データベース] タブの [サーバ名] フィールドに、サーバ名を入力します。

この例では、前のレッスンで指定した CEserver がサーバ名です。

7 [OK] をクリックして、Windows CE デバイスで実行中の *Alpha.db* データベースに接続します。

サーバを起動して Alpha データベースを開始したら、Windows CE デバイスで他のデータベースを開始できます。

❖ ネットワーク・サーバで複数のデータベースを開始するには、次の手順に従います。

1 Sybase Central の左ウィンドウ枠で、**CEserver** アイコンを選択します。

このアイコンは、Windows CE デバイスで実行中のネットワーク・サーバを表します。

2 [ファイル] メニューから、[データベースの開始] を選択します。

[データベースの開始] ダイアログが表示されます。

3 [データベース・ファイル] フィールドに次のパスを入力します。**\My Documents\Beta.db**

4 [OK] をクリックして、ネットワーク・サーバでデータベースを開始します。

ネットワーク・サーバでデータベースが開始されますが、デスクトップ・マシンから接続を開始する必要があります。

❖ Beta データベースに接続するには、次の手順に従います。

1. Sybase Central の左側のウィンドウ枠で、**Beta** データベース・アイコンを選択します。

   このアイコンは、Windows CE デバイスで先ほど開始したデータベースを表します。

2. [ファイル] メニューから、[接続] を選択します。

   [接続] ダイアログが表示されます。

3. [ID] タブで、次の値を入力します。

   - **ユーザ ID**： DBA
   - **パスワード**：SQL

4. [参照] をクリックし、[データ・ソース名] ダイアログで **CEdevice** を選択します。

5. [OK] をクリックします。

6. [OK] をクリックして、Windows CE デバイスで実行中の Beta データベースに接続します。

これで、Sybase Central を使って Alpha データベースと Beta データベースを表示して操作できます。

次の作業

次は、Windows CE でデータベースを切断してデータベース・サーバを停止する方法を学習します。

レッスン 3 ：サーバを停止する

Windows CE デバイスでネットワーク・サーバを停止する前に、デスクトップ・マシンから接続を切断する必要があります。

❖ Windows CE のデータベースから接続を切断するには、次の手順に従います。

1 Sybase Central で、[ツール] メニューから [切断] を選択します。

[接続] ダイアログが表示されます。

2 Alpha データベースへの接続を選択します。

3 [切断] をクリックします。

4 1 ～ 3 の手順を繰り返して、Beta データベースへの接続を切断します。

Sybase Central の Windows CE データベースから切断したので、ネットワーク・サーバを停止できます。

❖ サーバを停止するには、次の手順に従います。

1 [Today] 画面の右下にあるデータベース・サーバ・アイコンをタップします。

サーバ・メッセージ・ウィンドウが表示されます。

2 [シャットダウン] をタップします。

管理ユーティリティの使用

この項では、Windows CE のデータベースで SQL Anywhere Studio データベース管理ユーティリティを使用するときに考慮すべき点を説明します。

Windows CE のバックアップとリカバリ

データが破損したりメディアに障害が発生した場合にデータを失わないように、バックアップとリカバリが重要になります。Windows CE デバイスの盗難や紛失、またはメディアの障害によるデータ損失を防ぐには、Windows CE のデータベースを物理的に別の場所にバックアップするのが最善策です。バックアップとリカバリのユーティリティのほとんどは、Windows CE で使用できます。ただし、Windows CE ではバックアップを物理的に別の場所に保存できないため、これらのユーティリティは役に立ちません。

代わりに、データベース・ファイル全体をデスクトップ・マシンにコピーしてデータをバックアップします。また、同期を使用して Windows CE のデータベースのコピーをデスクトップ・マシンに維持することもできます。

詳細については、『Mobile Link 管理ガイド』> 「同期の基本」を参照してください。

チュートリアル：Sybase Central を使用した Windows CE データベースの管理

Sybase Central は、Adaptive Server Anywhere データベースのさまざまな設定、プロパティ、ユーティリティをグラフィカル・ユーザ・インタフェースを介して使用するデータベース管理ツールです。Sybase Central は、Mobile Link 同期など他の製品を管理する場合にも使用できます。

このチュートリアルでは、デスクトップ・マシンから Sybase Central を使用して Windows CE デバイス上のデータベースを管理する方法を簡単に紹介します。Windows CE デバイスのサンプル・データベース

に接続する方法を学習します。接続したあと、Sybase Central を使用して、データの表示や編集、ユーザの管理、ストアド・プロシージャの操作などができます。

前提条件

- このチュートリアルを開始する前に、「はじめに」46 ページのすべてのタスクを完了します。
- Windows CE デバイスをデスクトップ・マシンに接続します。

始める前に

始める前に、必ずサンプル・データベースのコピーを作成して、終了時に元のフォームに戻せるようにしておいてください。

❖ サンプル・データベースのコピーを作成するには、次の手順に従います。

1. Windows CE デバイスで、[ファイル エクスプローラ] を開きます。

   [スタート] メニューから、[プログラム] － [ファイル エクスプローラ] の順に選択します。

2. [Program Files] － [Sybase] － [ASA9] の順に選択して、Adaptive Server Anywhere ディレクトリに移動します。

3. 画面の空き領域で、タップしてスタイラス・ペンを押したままにします。

   ポップアップ・メニューが表示されます。

4. [新しいフォルダ] をタップします。

   新しいフォルダに「**Sample Backup**」という名前を付けます。

5. *asademo* データベース・ファイルをタップして押したままにします。

   「asademo」という名前のファイルがリストに 2 つ表示される場合があります。大きい方がデータベース・ファイルで、もう一方はトランザクション・ログ・ファイルです。必ず正しいファイルを選択してください。

6. ポップアップ・メニューが表示されたら、[コピー] をタップします。

7　先ほど作成した *Sample Backup* フォルダをタップして開きます。

フォルダが開きます。

8　画面の空き領域をタップしたままにします。

ポップアップ・メニューが表示されます。

9　[貼り付け] をタップします。

asademo.db ファイルのコピーが、*Sample Backup* フォルダに貼り付けられます。

ヒント
サンプル・データベースのオリジナル・バージョンを使用して作業してください。バックアップ・コピーは変更しないでください。

レッスン 1 ：サンプル・データベースを開始する

Sybase Central から接続するには、Windows CE デバイスでデータベースが実行されている必要があります。

❖ サンプル・データベースを開始するには、次の手順に従います。

1　Windows CE デバイスで、[ファイル エクスプローラ] を開きます。

[スタート] メニューから、[プログラム] － [ファイル エクスプローラ] の順に選択します。

2　[Program Files] － [Sybase] － [ASA9] の順に選択して、Adaptive Server Anywhere ディレクトリに移動します。

3　dbsrv9 アイコンをタップします。

[サーバ起動オプション] ダイアログが表示されます。

4 [データベース] フィールドに、サンプル・データベースのパスを次のように入力します。**\Program Files\Sybase\ASA9\asademo.db**

5 [サーバ名] フィールドに「**CEserver**」と入力して、ネットワーク・サーバに名前を割り当てます。

6 デフォルトのキャッシュ・サイズ 600kb を使用します。

ヒント
このチュートリアルの目的上、デフォルトのキャッシュ・サイズで十分です。大きいデータベースの場合は、キャッシュ・サイズを大きくするとパフォーマンスが向上します。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「パフォーマンス向上へのキャッシュの使用」を参照してください。

---

7 [TCP/IP を使用] オプションを選択します。

デスクトップ・マシンから Windows CE デバイスで実行中のデータベースに接続するには、TCP/IP 接続が必要です。

8 [OK] をタップして、ネットワーク・サーバで実行中のサンプル・データベースを開始します。

サーバ・ウィンドウが表示されて、すぐに閉じます。

9 デバイスの [Today] 画面に移動します。

10 画面の右下にあるサーバ・アイコンをタップします。

サーバ・メッセージ・ウィンドウが表示されます。

サーバ・メッセージ・ウィンドウに「要求を受信中です」というメッセージが表示されたら、次のレッスンに進んでください。

次の作業

次は、Sybase Central から Windows CE デバイスで実行中のデータベースに接続する方法を学習します。

レッスン 2 ：Sybase Central を起動して接続する

サンプル・データベースが Windows CE デバイスで実行しているので、Sybase Central から接続できます。接続すると、デスクトップ・マシンからデータベースを表示して管理できるようになります。

❖ Sybase Central から Windows CE デバイスのデータベースに接続するには、次の手順に従います。

1. デスクトップ・マシンで Sybase Central を開始します。

   [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [Sybase Central] の順に選択します。

2. [ツール] － [接続] を選択します。

   プラグインの選択を求めるメッセージが表示されたら、[Adaptive Server Anywhere 9] を選択して [OK] をクリックします。

   [接続] ダイアログが表示されます。

3. [ID] タブで、次の値を入力します。

   - **ユーザ ID：DBA**
   - **パスワード：SQL**

4. [ODBC データ・ソース名] オプションを選択します。

5. [参照] をクリックし、「Windows CE デバイスに接続するための ODBC データ・ソースの作成」52 ページで作成した *CEdevice* データ・ソースを [データ・ソース名] ダイアログから選択します。

6. [データベース] タブの [サーバ名] フィールドに、サーバ名を入力します。

   この例では、前のレッスンで指定した CEserver がサーバ名です。

7 [OK] をクリックして、Windows CE デバイスで実行しているサンプル・データベースに接続します。

次の作業

これで、Sybase Central からサンプル・データベースのデータを表示して管理できます。

参考資料

Sybase Central の詳細

Sybase Central の詳細については、チュートリアルの「Sybase Central を使用したデータベースの管理」299 ページを参照してください。そのチュートリアルには、データの表示と編集、ユーザの管理、ストアド・プロシージャの操作などのレッスンが含まれています。

ただし、デスクトップ・マシンで実行中のサンプル・データベースに Sybase Central を使用するための情報です。以下のレッスンは Windows CE のデータベースを管理するときには当てはまりません。

- 「レッスン 1 ：Sybase Central の起動と接続」301 ページ

  このチュートリアルで説明する接続手順に従ってください。

- 「レッスン 7 ：データベースのバックアップ」322 ページ

  このレッスンは無視してください。そこで説明するバックアップ手順は、Windows CE ではサポートされていません。

一度に複数のデータベースを管理

Windows CE デバイスのネットワーク・サーバで複数のデータベースを開始すると、一度に複数のデータベースを管理できます。

詳細については、「レッスン 2 ：サーバで複数のデータベースを開始する」83 ページを参照してください。

チュートリアルのクリーンアップ

チュートリアルを完了すると、サーバを停止し、作成してあるバックアップ・コピーからサンプル・データベースを復元する必要があります。

❖ サーバを停止するには、次の手順に従います。

1 [Today] 画面の右下にあるデータベース・サーバ・アイコンをタップします。

   サーバ・メッセージ・ウィンドウが表示されます。

2 [シャットダウン] をタップします。

❖ サンプル・データベースをリストアするには、次の手順に従います。

1 Windows CE デバイスで、[ファイル エクスプローラ] を開きます。

   [スタート] メニューから、[プログラム] － [ファイル エクスプローラ] の順に選択します。

2 [Program Files] － [Sybase] － [ASA9] の順に選択して、Adaptive Server Anywhere ディレクトリに移動します。

3 *Sample Backup* フォルダをタップして開きます。

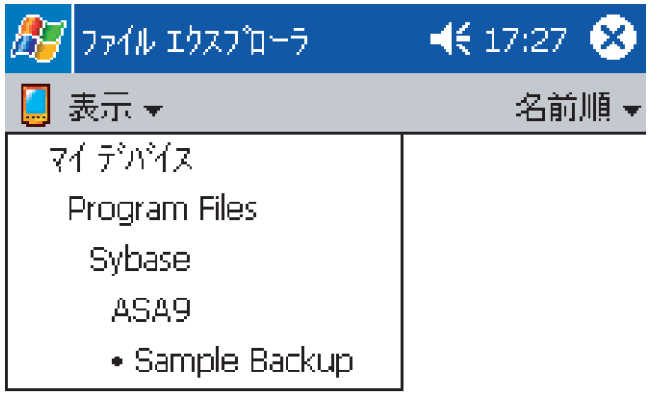
4 *asademo* ファイルをタップして押したままにします。

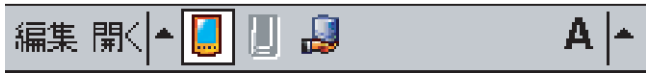
   ポップアップ・メニューが表示されます。

5 [コピー] をタップします。

6 *ASA9* フォルダまで参照します。

   ウィンドウの一番上にあるナビゲーション・ドロップダウン・リストをタップしてから、[ASA9] をタップします。





7 画面の空き領域をタップしたままにします。

ポップアップ・メニューが表示されます。

8 [貼り付け] をタップします。

9 警告ダイアログが表示されます。

[OK] をタップし、バックアップから作成したコピーで既存の *asademo.db* ファイルを置き換えます。

チュートリアル：Interactive SQL を使用した Windows CE データベースの管理

Interactive SQL は、データベース内のデータの変更や問い合わせと、データベース構造の修正ができるアプリケーションです。Interactive SQL には、SQL 文を入力するためのウィンドウ枠と、クエリの処理状況や結果セットを表示するウィンドウ枠があります。

このチュートリアルでは、デスクトップ・マシンから Interactive SQL を使用して Windows CE デバイス上のデータベースを管理する方法を簡単に紹介します。まず、Interactive SQL から Windows CE デバイスのサンプル・データベースに接続する方法を学習します。接続したら、Interactive SQL を使用して SQL 文を実行できます。

始める前に

始める前に、必ずサンプル・データベースのコピーを作成して、終了時に元のフォームに戻せるようにしておいてください。

❖ サンプル・データベースのコピーを作成するには、次の手順に従います。

1. Windows CE デバイスで、[ファイル エクスプローラ] を開きます。

   [スタート] メニューから、[プログラム] － [ファイル エクスプローラ] の順に選択します。

2. [Program Files] － [Sybase] － [ASA9] の順に選択して、Adaptive Server Anywhere ディレクトリに移動します。

3. 画面の空き領域で、タップしてスタイラス・ペンを押したままにします。

   ポップアップ・メニューが表示されます。

4. [新しいフォルダ] をタップします。

   新しいフォルダに「**Sample Backup**」という名前を付けます。

5. *asademo* データベース・ファイルをタップして押したままにします。

「asademo」という名前のファイルがリストに 2 つ表示される場合があります。大きい方がデータベース・ファイルで、もう一方はトランザクション・ログ・ファイルです。必ず正しいファイルを選択してください。

6 ポップアップ・メニューが表示されたら、[コピー] をタップします。

7 先ほど作成した *Sample Backup* フォルダをタップして開きます。

フォルダが開きます。

8 画面の空き領域をタップしたままにします。

ポップアップ・メニューが表示されます。

9 [貼り付け] をタップします。

asademo.db ファイルのコピーが、*Sample Backup* フォルダに貼り付けられます。

ヒント
サンプル・データベースのオリジナル・バージョンを使用して作業してください。バックアップ・コピーは変更しないでください。

レッスン 1 ：サンプル・データベースを開始する

Interactive SQL から接続する場合は、Windows CE でデータベースが実行している必要があります。

❖ サンプル・データベースを開始するには、次の手順に従います。

1 Windows CE デバイスで、[ファイル エクスプローラ] を開きます。

[スタート] メニューから、[プログラム] ー [ファイル エクスプローラ] の順に選択します。

2 [Program Files] － [Sybase] － [ASA9] の順に選択して、Adaptive Server Anywhere ディレクトリに移動します。

3 dbsrv9 アイコンをタップします。

[サーバ起動オプション] ダイアログが表示されます。

4 [データベース] フィールドに、サンプル・データベースのパスを次のように入力します。**\Program Files\Sybase\ASA9\asademo.db**

5 [サーバ名] フィールドに「**CEserver**」と入力して、ネットワーク・サーバに名前を割り当てます。

6 [キャッシュ] フィールドに「**5MB**」と入力して、キャッシュ・サイズを設定します。

Windows CE のデフォルトのキャッシュ・サイズは 600kb ですが、パフォーマンスを向上させるには、これより大きいキャッシュ・サイズをおすすめします。

7 [TCP/IP を使用] を選択します。

8 [OK] をクリックして、ネットワーク・サーバで実行中のサンプル・データベースを開始します。

サーバ・ウィンドウが表示されて、すぐに閉じます。

9 デバイスの [Today] 画面に移動します。

10 画面の右下にあるサーバ・アイコンをタップします。

サーバ・ウィンドウが表示されます。

サーバ・ウィンドウに「要求を受信中です」というメッセージが表示されたら、次のレッスンに進んでください。

次の作業

次は、Interactive SQL から Windows CE デバイスで実行中のデータベースに接続する方法を学習します。

レッスン 2 ：Interactive SQL を開始して接続する

サンプル・データベースが Windows CE デバイスで実行していれば、Interactive SQL から接続して、デスクトップ・マシンからデータベースを表示して管理できます。

❖ Interactive SQL から Windows CE デバイスのデータベースに接続するには、次の手順に従います。

1. デスクトップ・マシンで Interactive SQL を起動します。

   [スタート] － [プログラム] － [SQL Anywhere 9] － [Adaptive Server Anywhere] － [Interactive SQL] の順に選択します。

2. Interactive SQL が起動すると、[接続] ダイアログが自動的に表示されます。

   [接続] ダイアログが自動的に表示されない場合は、[SQL] メニューから [接続] を選択してください。

3. [ID] タブで、次の値を入力します。

   - **ユーザ ID**： DBA
   - **パスワード**： SQL

4. [ODBC データ・ソース名] オプションを選択します。

5. [参照] をクリックし、「はじめに」46 ページで作成した **CEdevice** データ・ソースを [データ・ソース名] ダイアログから選択します。

6. [データベース] タブの [サーバ名] フィールドに、サーバ名を入力します。

   この例では、前のレッスンで指定した CEserver がサーバ名です。

7. [OK] をクリックして、Windows CE デバイスで実行中のサンプル・データベースに接続します。

次の作業

これで、Interactive SQL からサンプル・データベースのデータを表示して管理できます。

参考資料

Interactive SQL の詳細

Interactive SQL の詳細については、チュートリアルの「Interactive SQL の使用」271 ページを参照してください。そのチュートリアルには、データの表示と編集、ユーザの管理、SQL 文の操作などのレッスンが含まれています。

ただし、デスクトップ・マシンで実行しているサンプル・データベースに Interactive SQL を使用するための情報です。次のレッスンは Windows CE のデータベースを管理するときには当てはまりません。

- 「レッスン 1 ：Interactive SQL の起動」273 ページ

  このチュートリアルで説明した接続手順に従ってください。

チュートリアルのクリーンアップ

チュートリアルを完了すると、サーバを停止し、作成してあるバックアップ・コピーからサンプル・データベースを復元する必要があります。

❖ サーバを停止するには、次の手順に従います。

1. [Today] 画面の右下にあるデータベース・サーバ・アイコンをタップします。

   サーバ・メッセージ・ウィンドウが表示されます。

2. [シャットダウン] をタップします。

❖ サンプル・データベースをリストアするには、次の手順に従います。

1. Windows CE デバイスで、[ファイル エクスプローラ] を開きます。

[スタート] メニューから、[プログラム] － [ファイル エクスプローラ] の順に選択します。

2 [Program Files] － [Sybase] － [ASA9] の順に選択して、Adaptive Server Anywhere ディレクトリに移動します。

3 Sample Backup フォルダをタップして開きます。

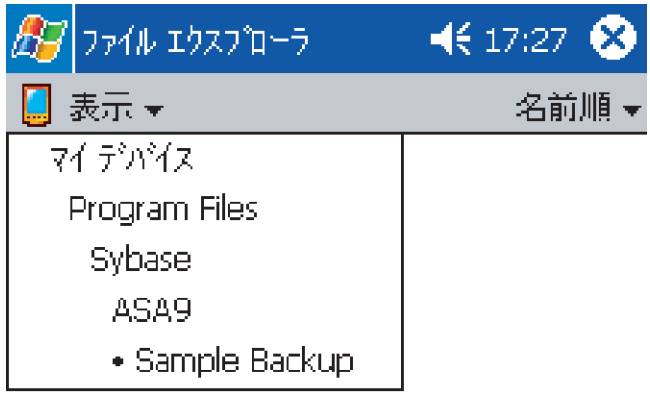
4 *asademo* ファイルをタップして押したままにします。

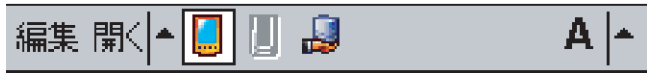
ポップアップ・メニューが表示されます。

5 [コピー] をタップします。

6 ASA9 フォルダまで参照します。

ウィンドウの一番上にある移動用ドロップダウン・リストをタップしてから、[ASA9] をタップします。





7　画面の空き領域をタップしたままにします。

ポップアップ・メニューが表示されます。

8　[貼り付け] をタップします。

9　警告ダイアログが表示されます。

[OK] をタップし、バックアップから作成したコピーで既存の *asademo.db* ファイルを置き換えます。

Windows CE のデータのバックアップとリカバリ

データベースが破損したりエラーが発生した場合にデータを失わないように、バックアップとリカバリが重要になります。Windows CE デバイスの盗難や紛失、またはメディアの障害によるデータ損失を防ぐには、Windows CE のデータベースは物理的に別の場所にバックアップするのが最善策です。バックアップとリカバリのほとんどの機能は、Windows CE で使用できます。

バックアップとリカバリのユーティリティを使用してデータを物理的に別の場所に保管するのではなく、データベース・ファイル全体をデスクトップ・マシンにコピーして、データを Windows CE データベースからデスクトップ・マシンにコピーします。また、同期を使用してデータベースの最新のコピーをデスクトップ・マシンに維持することもできます。

詳細については、『Mobile Link 管理ガイド』> 「同期の基本」を参照してください。

「**バックアップ**」とはデータベースにある情報のコピーであり、元のデータベースとは物理的に別の場所に格納されます。ディスク・ドライブの破損などが原因でデータベースが使用不可能になった場合は、バックアップからリストアできます。発生した障害の性質によって異なりますが、使用不可能になった時点までにデータベースでコミットされていたすべての変更のバックアップをリストアできる場合もあります。

バックアップからのデータベースのリストアは、データベースの**リカバリ**作業の 1 つです。もう 1 つの作業としては、オペレーティング・システムやデータベース・サーバのクラッシュ、不正な終了操作からのリカバリがあります。データベース・サーバは起動時に、前回のセッションの最後にデータベースが正しく終了したかどうかをチェックします。正しく終了しなかった場合は、情報をリストアするための自動リカバリ処理が実行されます。このメカニズムでは、最後にコミットされたトランザクションまでのすべての変更がリカバリされます。

この項では、Windows CE で完全サポートされているバックアップとリカバリのタスクに加え、サポートされていないタスクや、機能が異なるタスクについても説明します。サポートされていないタスクについては、可能な限り代わりの方法を紹介しました。

したがって、この項の説明は『ASA データベース管理ガイド』> 「バックアップとデータ・リカバリ」の一般的な情報を補足するものです。

Windows CE での検証

検証は、データベース、テーブル、トランザクション・ログなどの特定タイプのファイル破損をテストするために実行します。検証は破損を検出する最初のステップで、バックアップ手順の過程で実行されます。

データベースの検証

このタスクは、Sybase Central、Interactive SQL、またはコマンド・ラインを使用して実行できます。

詳細については、『ASA データベース管理ガイド』> 「データベースの検証」を参照してください。

テーブルの検証

このタスクは、Sybase Central または Interactive SQL を使用して実行できます。

詳細については、『ASA データベース管理ガイド』> 「単一テーブルの検証」を参照してください。

トランザクション・ログの検証

トランザクション・ログ・ファイルの検証処理は、運用データベースで使用されている (オンライン) か使用されていない (オフラインまたはバックアップ・コピー) かによって異なります。

オンライン　ミラーされたトランザクション・ログを使ってデータベースを実行します。データベース・サーバは、データベースを起動するたびに自動的にトランザクション・ログを検証します。

オフラインまたはバックアップ・コピー　このタイプの検証にはログ変換ユーティリティ (dbtran) を使用する必要がありますが、Windows CE ではサポートされていません。

代わりに、トランザクション・ログ・ファイルをデスクトップ・マシンにコピーして、それにログ変換ユーティリティを実行できます。ログ変換ユーティリティでログ・ファイルを正常に読み込めれば、そのログは有効です。

詳細については、『ASA データベース管理ガイド』> 「ログ変換ユーティリティ」を参照してください。

Windows CE でのアーカイブ・バックアップの使用

「**アーカイブ・バックアップ**」では、データベース・ファイルとトランザクション・ログを 1 つのアーカイブ・ファイル (通常はテープ・ドライブ上) にコピーします。

テープへの直接のバックアップ

このタイプのバックアップには Sybase Central の [データベース・バックアップ] ウィザードを使用する必要がありますが、Windows CE ではサポートされていません。

代わりに、データベース・ファイルとトランザクション・ログ・ファイルをデスクトップ・マシンにコピーして、[データベース・バックアップ] ウィザードを実行します。

詳細については、『ASA データベース管理ガイド』> 「データベースを直接テープにバックアップする」を参照してください。

アーカイブ・バックアップのリストア

アーカイブ・バックアップ (通常はテープ) を使用すると、メディアに障害が発生した場合、データのリカバリに RESTORE 文を使用します。

または、破損したデータベース・ファイルとトランザクション・ログ・ファイルをデスクトップ・マシンにコピーし、RESTORE 文を使用してアーカイブ・ファイルのデータを復元します。

詳細については、『ASA データベース管理ガイド』> 「アーカイブのバックアップのリストア」を参照してください。

Windows CE でのライブ・バックアップの使用

「**ライブ・バックアップ**」では、第 2 のマシンにトランザクション・ログの冗長コピーを作成します。データベース・サーバを実行しているマシンが使用不能になった場合は、バックアップを使用してシステムを再開できます。

ライブ・バックアップは、Windows CE ではサポートされていません。

Windows CE でのイメージ・バックアップの使用

「**イメージ・バックアップ**」では、データベース・ファイルとトランザクション・ログのコピーをそれぞれ別のファイルとして作成します。イメージ・バックアップには、次の 2 種類があります。

フル・バックアップ　フル・バックアップでは、データベース・ファイルとトランザクション・ログ・ファイルの両方をバックアップします。

インクリメンタル・バックアップ　インクリメンタル・バックアップでは、トランザクション・ログ・ファイルだけがバックアップされます。

どちらを選んでも、3 つのオプションがあります。

- 元のトランザクション・ログを使用する
- トランザクション・ログを削除する
- トランザクション・ログの名前を変更する

元のトランザクション・ログを使用する

トランザクション・ログ・ファイルが変更されないので、最も簡単なイメージ・バックアップ方法です。

このタスクは、Sybase Central または Interactive SQL を使用して実行できます。バックアップ・ディレクトリを指定するときに -s オプションを加えると、コマンド・ラインからこのタスクを実行できます。

詳細については、『ASA データベース管理ガイド』> 「バックアップ・ユーティリティのオプション」を参照してください。

詳細については、『ASA データベース管理ガイド』> 「バックアップを作成し、元のトランザクション・ログを継続して使用する」を参照してください。

トランザクション・ログを削除する

このタイプのイメージ・バックアップでは、ファイルのバックアップを作成し、元のトランザクション・ログを削除してから、新しいトランザクション・ログを作成します。新しいトランザクション・ログを開始すると記憶領域を解放できるので、Windows CE に便利です。

このタスクは、Sybase Central または Interactive SQL を使用して実行できます。バックアップ・ディレクトリを指定するときに -s オプションを加えると、コマンド・ラインからこのタスクを実行できます。

詳細については、『ASA データベース管理ガイド』> 「バックアップ・ユーティリティのオプション」を参照してください。

詳細については、『ASA データベース管理ガイド』> 「バックアップを作成し、元のトランザクション・ログを削除する」を参照してください。

トランザクション・ログの名前を変更する

ファイルのバックアップ・コピーを作成するだけでなく、バックアップ時にトランザクション・ログの名前がオフライン・ログに変更され、バックアップ時に使用するログと同じ名前で新しいトランザクション・ログが開始します。新しいトランザクション・ログを開始すると記憶領域を解放できるので、Windows CE に便利です。

このタスクは、Sybase Central または Interactive SQL を使用して実行できます。

バックアップ・ディレクトリを指定するときに -s オプションを加えると、コマンド・ラインからこのタスクを実行できます。

詳細については、『ASA データベース管理ガイド』> 「バックアップ・ユーティリティのオプション」を参照してください。

詳細については、『ASA データベース管理ガイド』> 「バックアップを作成し、元のトランザクション・ログの名前を変更する」を参照してください。

Windows CE のデータベースのリカバリ

データベースに影響する障害は、2 種類あります。

システム・エラー　トランザクションの途中で、コンピュータまたはオペレーティング・システムが停止した場合に発生します。これは、正しい手順でコンピュータを終了または再起動しなかったり、他のアプリケーションによってオペレーティング・システムがクラッシュしたり、停電した場合に起こる可能性があります。

システム・エラーが発生した後、次にデータベースを起動するときに、データベース・サーバは自動的にリカバリします。システム・エラー以前にコミットされたトランザクションの結果は保存されます。システム・エラーの前にコミットされていなかったトランザクションによる変更はすべてキャンセルされます。

メディア障害　メディア障害は、データベース・ファイルやトランザクション・ログが使用できないときに起こります。これは、データベース・ファイルを格納しているファイル・システムまたはデバイスが使用できなくなった場合や、ファイルが破損した場合などに発生します。

この項では、メディア障害からイメージ・バックアップを使用してリカバリするタスクについて説明します。

データベース・ファイルのリカバリ

データベース・ファイルでエラーが発生した場合、バックアップのデータベース・ファイルを最新の状態にするとリカバリできます。これは、トランザクション・ログの変更をバックアップのデータベース・ファイルに適用して行います。バックアップのトランザクション・ログが複数ある場合は、それぞれを順番に適用し、最後に運用トランザクション・ログを適用して最新の変更を更新します。

このタスクは、Windows CE ではサポートされていません。Windows CE ではこのタスクを [サーバ起動オプション] ダイアログから実行します。

❖ データベース・ファイルのメディア障害からリカバリするには、次の手順に従います。

1. 現在のトランザクション・ログの追加バックアップ・コピーを作成します。データベース・ファイルは失われており、最後のバックアップ以降に行われた唯一の変更の記録はトランザクション・ログにあります。

2. リカバリの処理中に使用するファイルを保存する「リカバリ・ディレクトリ」を作成します。

ヒント
リカバリ・ディレクトリを My Documents ディレクトリ内に入れておくと、サーバを起動するときに [参照] を使用してデータベース・ファイルを見つけることができます。

リカバリ・ディレクトリを My Documents ディレクトリの第 1 レベルのフォルダにします。これ以外にすると、[参照] 機能でファイルが見つかりません。

3 最後のフル・バックアップのデータベース・ファイルをリカバリ・ディレクトリにコピーします。

4 バックアップされたトランザクション・ログに保持されているトランザクションをリカバリ・データベースに適用します。

各ログ・ファイルについて、日付順に次の操作を行います。

- ログ・ファイルをリカバリ・ディレクトリにコピーします。
- [サーバ起動オプション] ダイアログを使用して、データベース・サーバを起動します。[オプション] フィールドに -a オプションと指定し、トランザクション・ログを次のように適用します。**-a** *log_name.log*

データベース・サーバは、トランザクション・ログが適用されると自動的に停止します。

5 オンライン・トランザクション・ログをリカバリ・ディレクトリにコピーします。オンライン・トランザクション・ログのトランザクションをリカバリ・データベースに適用します。

6 リカバリ・データベースに妥当性検査を実行します。

詳細については、「Windows CE での検証」103 ページを参照してください。

7 データベース・ファイルのリカバリ後のバックアップを作成します。

8 データベース・ファイルを運用ディレクトリに移します。

9　運用データベースへのユーザ・アクセスを許可します。

トランザクション・ログ・ファイルのリカバリ （ミラーなし）

❖ ミラーされていないトランザクション・ログのメディア障害からリカバリするには、次の手順に従います (部分リカバリの場合)。

1　ただちにデータベース・ファイルの追加バックアップ・コピーを作成します。トランザクション・ログは失われており、最後にバックアップが行われてから最新のチェックポイントまでの間に加えられた変更の唯一のレコードはデータベース・ファイルです。

2　トランザクション・ログ・ファイルを削除するか、名前を変更します。

3　[サーバ起動オプション] ダイアログの [オプション] フィールドで指定した -f オプションを使用して、データベースを再起動します。

警告
-f オプションは、データベースが SQL Remote または Replication Server のレプリケーション・システムに関連していない場合にだけ使用してください。データベースが SQL Remote レプリケーション・システム内の統合データベースである場合は、リモート・データベースを再抽出する必要があります。

-f オプションを指定しないと、トランザクション・ログの欠如がエラーとして報告されます。オプションを指定すると、サーバは最新のチェックポイント時の状態にデータベースをリストアし、チェックポイントの時点でコミットされていなかったトランザクションをすべてロールバックします。その後新しいトランザクション・ログが作成されます。

トランザクション・ログ・ファイルのリカバリ （ミラー使用）

このリカバリ・タスクには Sybase Central の [ログ・ファイルの変換] ウィザードを使用する必要がありますが、Windows CE ではサポートされていません。

ただし、トランザクション・ログ・ファイルをデスクトップ・マシンにコピーして、[ログ・ファイルの変換] ウィザードを実行できます。

詳細については、『ASA データベース管理ガイド』> 「トランザクション・ログ・ミラーのメディア障害からリカバリする」を参照してください。

コミットされていない操作のリカバリ

このリカバリ・タスクには Sybase Central の [ログ・ファイルの変換] ウィザードを使用する必要がありますが、Windows CE ではサポートされていません。

代わりに、データベース・ファイルとトランザクション・ログ・ファイルをデスクトップ・マシンにコピーして [ログ・ファイルの変換] ウィザードを実行することで、このリカバリ・タスクを実行できます。

詳細については、『ASA データベース管理ガイド』> 「コミットされていない操作のリカバリ」を参照してください。

その他のタスク

トランザクション・ログ・ファイルの場所の変更

このタスクには Sybase Central の [ログ・ファイル設定の変更] ウィザードを使用する必要がありますが、Windows CE ではサポートされていません。

トランザクション・ログ・ファイルのバックアップ・コピーの名前変更

このタスクは、Sybase Central、Interactive SQL、コマンド・ラインなどを使用して実行できます。

詳細については、『ASA データベース管理ガイド』> 「バックアップ中にトランザクション・ログのバックアップ・コピーの名前を変更する」を参照してください。

未処理のトランザクションがある接続の特定

このタスクは、Interactive SQL またはコマンド・ラインを使用して実行できます。

詳細については、『ASA データベース管理ガイド』> 「未処理のトランザクションがある接続の判断」を参照してください。

Windows CE でサポートされていない SQL Anywhere Studio の機能

この項では、SQL Anywhere Studio のコンポーネントや機能のうち、Windows CE でサポートされていないか、動作が異なるものをリストにします。サポートされていない機能の代わりになる機能がある場合は、それについても説明します。

Windows CE でサポートされているコンポーネントとサポートされていないコンポーネントの詳細については、「Windows オペレーティング・システムと NetWare オペレーティング・システム」132 ページを参照してください。

Windows CE でサポートされていない Adaptive Server Anywhere の機能

以下の機能は、Windows CE ではサポートされていません。

コンポーネントまたは機能	考慮事項
パーソナル・データベース・サーバ パーソナル・サーバは *dbeng9* という名前です。この実行プログラムは、ネットワークを介したクライアント／サーバ通信をサポートしていません。	パーソナル・サーバの代わりに、ネットワーク・データベース・サーバが提供されています。ネットワーク・サーバは *dbsrv9* という名前です。この実行プログラムは、ローカル接続とネットワーク経由のクライアント／サーバ通信をサポートしています。 詳細については、『ASA データベース管理ガイド』> 「データベース・サーバの実行」を参照してください。
Open Client Sybase Open Client は、カスタマ・アプリケーション、サードパーティ製品、その他の Sybase 製品に、Adaptive Server Anywhere およびその他の Open Server と通信するために必要なインタフェースを提供します。	Open Client のアプリケーションを実行するには、そのコンポーネントをアプリケーションが動作しているマシンにインストールして構成する必要があります。これらのコンポーネントは Windows CE でサポートされていないため、Open Client はそのプラットフォームで機能しません。

コンポーネントまたは機能	考慮事項
データベースの Java　Adaptive Server Anywhere は Java Runtime Environment です。これは、Java クラスをデータベース・サーバで実行できるようにします。	Adaptive Server Anywhere or Windows CE は、Java の実行環境の機能は果たしません。したがって、Java クラスを Windows CE のデータベース・サーバで実行することはできません。
リモート・データ・アクセス Adaptive Server Anywhere のリモート・データ・アクセス機能によって、他のデータ・ソースのデータにアクセスできます。この機能を使用すると、Adaptive Server Anywhere のデータベースにデータをマイグレートし、データベースにデータのクエリができます。	この機能は、Windows CE ではサポートされていません。 リモート・データ・アクセス機能を使用している Windows CE デバイスにデータベースをコピーした場合、プロキシ・テーブルのデータにアクセスしようとすると、通信エラーが発生します。
外部ストアド・プロシージャ Adaptive Server Anywhere では、ストアド・プロシージャまたはユーザ定義関数から外部ライブラリの関数を呼び出すことができます。	Adaptive Server Anywhere for Windows CE は、システム拡張ストアド・プロシージャなどのプロシージャを外部ライブラリから呼び出す機能をサポートしていません。
キャッシュ・サイズの動的な調整　これは、キャッシュ・サイズを必要に応じて増やしたり、他のアプリケーションにメモリが必要な場合にキャッシュサイズを減らしたりする機能です。	Windows CE では、キャッシュ・サイズの動的な調整をサポートしていません。Windows CE では、キャッシュは [サーバ起動オプション] ダイアログで設定し、変化しません。
SPX プロトコル　SPX は、Novell から提供されている通信プロトコルです。	Windows CE では、SPX プロトコルの代わりに TCP/IP プロトコルを使用します。

Windows CE で制限のある Adaptive Server Anywhere の機能

以下の機能は、Windows CE では機能が異なるか、制限があります。

コンポーネントまたは機能	考慮事項
ODBC クライアント Microsoft では、ODBC （オープン・データベース・コネクティビティ) というインタフェースを定義しています。これは、Windows95/98/Me と WindowsNT/2000/XP 環境でクライアント・アプリケーションをデータベース管理システムに接続するための標準インタフェースです。アプリケーション開発システムなど、多くのクライアント・アプリケーションは、ODBC インタフェースを使用して、さまざまなデータベース・システムにアクセスします。	Windows CE では ODBC ドライバ・マネージャや ODBC アドミニストレータが提供されていないため、Windows デスクトップ クライアントとは ODBC クライアントの機能が異なります。このプラットフォームでは、Adaptive Server Anywhere はファイルに保管されている ODBC データ・ソースを使用します。 詳細については、『ASA データベース管理ガイド』> 「Windows CE での ODBC データ・ソースの使用」を参照してください。
jConnect jConnect JDBC ドライバは、JavaSoft JDBC 標準を Java で実装したものです。これにより、Java 開発者は多層／異機種環境でもネイティブなデータベース・アクセスができます。	jConnect ドライバは、Windows CE のデータベースを作成するときに有効にできます。これは、Java をサポートしているデスクトップ・マシンにデータベースを移動する場合に便利です。ただし、データベース・サイズが大きくなり、jConnect 機能を使用しない場合でも、データベースを実行するための必要メモリが約 200 KB 増えます。Windows CE の限られたメモリ環境でデータベースを実行する場合は、この追加のメモリ要件を考慮してください。
iAnywhere JDBC ドライバ iAnywhere JDBC ドライバでは、pure Java の jConnect JDBC ドライバよりパフォーマンスと機能面で有利な JDBC ドライバが提供されていますが、pure Java のソリューションではありません。	iAnywhere JDBC ドライバは、Windows CE デバイスに Java 仮想マシンをインストールしている場合にのみ有効にできます。

Windows CE でサポートされていない SQL Remote の機能

SQL Remote は、次の例外を除いて Windows CE ではサポートされていません。

コンポーネントまたは機能	考慮事項
Adaptive Server Anywhere 抽出ユーティリティ (dbxtract)　このユーティリティは、Adaptive Server Anywhere リモート・データベースを Adaptive Server Enterprise または Adaptive Server Anywhere の統合データベースから抽出します。	Windows CE は、このユーティリティをサポートしていません。必要な場合は、Windows CE のデータベースをデスクトップ・マシンにコピーすると、Adaptive Server Anywhere 抽出ユーティリティを使えるようになります。
MAPI メッセージ・タイプ Microsoft の MAPI (Message Application Programming Interface) は、Microsoft Mail のような一般的な電子メール・システムで使用され、SQL Remote メッセージングのメッセージを伝送できます。	こメッセージ・システムは、Windows CE ではサポートされていません。 Windows CE は SQL Remote レプリケーションで使用される他のメッセージ・タイプを複数サポートしています。それらについては、『SQL Remote ユーザーズ・ガイド』> 「サポートされるオペレーティング・システム」を参照してください。
VIM メッセージ・タイプ　Lotus Notes と cc:Mail で使用されている Lotus の VIM (Vendor Independent Messaging) を使って、SQL Remote メッセージングのメッセージを伝送できます。	こメッセージ・システムは、Windows CE ではサポートされていません。 Windows CE は、SQL Remote レプリケーションで使用される他のメッセージ・タイプを複数サポートしています。それらについては、『SQL Remote ユーザーズ・ガイド』> 「サポートされるオペレーティング・システム」を参照してください。

Windows CE でサポートされていない管理ツール

SQL Anywhere Studio には、データベースを管理するための数々のツールが付属しています。たとえば、Sybase Central、Interactive SQL、コマンド・ライン・ユーティリティなどがあります。これらの管理ツールは、Windows CE では使用できません。データベースの管理は、Windows CE デバイスに接続している Windows ベースのデスクトップ・マシンから行います。

詳細については、「管理ユーティリティの使用」87 ページを参照してください。

Windows CE でサポートされていないサーバ・オプション

この項では、Windows CE でサポートされていないか、機能が異なるデータベース・サーバ・オプションについて説明します。

オプション	考慮事項
@data オプション　このオプションを使用すると、コマンド・ラインで環境変数と設定ファイルを指定できます。	Windows CE では、環境変数をサポートしていません。
-? サーバ・オプション　このオプションは、使用率の情報を表示するために使用します。	Windows CE では、このオプションをサポートしていません。
-ca オプション　このオプションは、静的キャッシュ・サイズを強制するために使用します。	Windows CE では、キャッシュ・サイズの動的な調整はサポートされていません。
-ch オプション　このオプションは、自動キャッシュ増加の上限として最大キャッシュ・サイズを設定するために使用します。	Windows CE では、キャッシュ・サイズの動的な調整はサポートされていません。
-cl オプション　このオプションは、自動キャッシュ増加の下限として最小キャッシュ・サイズを設定するために使用します。	Windows CE では、キャッシュ・サイズの動的な調整はサポートされていません。

オプション	考慮事項
-cs オプション　このオプションは、サーバ・メッセージ・ウィンドウにキャッシュ・サイズの変化を表示するために使用します。	Windows CE では、キャッシュ・サイズの動的な調整はサポートされていません。
-cw オプション　このオプションは、データベース・サーバのキャッシュ・サイズを設定する AWE (Address Windowing Extensions) の使用を有効にするために使用します。	Windows CE では、AWE をサポートしていません。
-d オプション　このオプションは、POSIX I/O の使用を強制するために使用します。	Windows CE では、POSIX I/O をサポートしていません。
-ec オプション　このオプションは、すべてのクライアントとの間で伝送される Adaptive Server Anywhere のネイティブ・パケット (DBLib、ODBC、OLE DB) を暗号化します。TDS パケットは暗号化されません。	強力な通信暗号化は、Windows CE ではサポートされていません。このサーバ・オプションでは、**none** (なし) と **simple** (単純) の設定だけがサポートされています。
-gb オプション　このオプションは、データベース・プロセスの優先度クラスを設定するために使用します。	Windows CE では、Priority クラスをサポートしていません。
-ge オプション　このオプションは、外部関数のスタック・サイズを設定するために使用します。	Windows CE では、外部関数をサポートしていません。
-gss オプション　このオプションは、サーバの内部実行スレッド当たりのスタック・サイズを設定するために使用します。	このオプションは、Windows オペレーティング・システムでは無効です。
-gx オプション　このオプションは、システム・コールのブロックを同時に実行できる最大要求数を設定するために使用します。	このオプションは、Windows NT でのみ使用されます。Windows CE など他のプラットフォームでは、-gx 値は -gt 値に相当します。

オプション	考慮事項
-qi オプション　このオプションは、データベース・サーバのトレイ・アイコンとウインドウを表示するかどうかを制御します。	実行中、ネットワーク・サーバは Windows CE デバイスの [Today] 画面の右下にアイコンとして表示されます。この機能は無効にできません。
-s オプション　このオプションは、syslog メッセージのユーザ ID を設定するために使用します。	UNIX オペレーティング・システムのみに該当します。
-sc オプション　このオプションは、共有メモリ通信プロトコルを無効にして、名前付きパイプを使うために使用します。	Windows CE では、名前付きパイプ・プロトコルをサポートしていません。
-tmf オプション　このオプションは、異常な状況で分散トランザクションからリカバリするために使用します。	Windows CE では、分散トランザクション・コーディネータをサポートしていません。
-tmt オプション　このオプションは、分散トランザクションに参加するための再エンリスト・タイムアウトを設定するために使用します。	Windows CE では、分散トランザクション・コーディネータをサポートしていません。
-u オプション　このオプションは、オペレーティング・システムのディスク・キャッシュを使ってファイルを開くために使用します。	このサーバ・オプションは、ハード・ディスク上で実行しない Windows CE のプラットフォームには該当しません。
-ua オプション　このオプションは、非同期 I/O を無効にするために使用します。	Linux オペレーティング・システムのみに該当します。
-ud オプション　このオプションは、サーバをデーモンとして実行するために使用します。	UNIX オペレーティング・システムのみに該当します。
-ut オプション　このオプションは、テンポラリ・ファイルをタッチするために使用します。	UNIX オペレーティング・システムのみに該当します。

オプション	考慮事項
-ux オプション　このオプションは、Linux と Solaris で [サーバ起動オプション] ダイアログを開いたり、サーバ・メッセージ・ウィンドウを開くために使用します。	Linux および Solaris オペレーティング・システムのみに該当します。
-y オプション　このオプションを使用すると、サーバが Windows サービスとして実行します。	Windows CE では、Windows サービスをサポートしていません。

Windows CE でサポートされていない SQL 文

クライアント・アプリケーションは、クエリを使用して情報を検索したり、テーブルに新しいローを挿入したりするデータベース・タスクを実行する場合に、SQL (Structured Query Language) 文を使用します。この項では、Windows CE でサポートされていないか、機能が異なるか、機能に制限のある SQL 文について説明します。

SQL 文の全リストについては、『ASA SQL リファレンス・マニュアル』> 「SQL 文」を参照してください。

以下の SQL 文は、Windows CE ではサポートされていません。

SQL 文	考慮事項
ALTER WRITEFILE 文 (旧式) この文は、ライト・ファイルが参照する読み込み専用データベース・ファイルの名前を変更する場合に使用します。	Windows CE ではライト・ファイルをサポートしていないため、この文をサポートしていません。
CREATE COMPRESSED DATABASE 文 (旧式)　この文は、既存のデータベース・ファイルから圧縮されたデータベースを作成したり、圧縮されたデータベースを展開したりするために使用します。	Windows CE では圧縮データベースをサポートしていないため、この文をサポートしていません。

SQL 文	考慮事項
CREATE EXISTING TABLE 文 この文は、リモート・サーバ上の既存のオブジェクトを表す新しいプロキシ・テーブルを作成するために使用します。	Windows CE ではプロキシ・テーブルをサポートしていないため、この文をサポートしていません。
CREATE EXTERNLOGIN 文 この文は、リモート・サーバとの通信に使用される代替ログイン名とパスワードを割り当てるために使用します。	ローカル・データベースのプロキシ・テーブルにリモート・オブジェクトをマップするには、リモート・サーバに接続する必要があります。Windows CE ではプロキシ・テーブルをサポートしていないため、この文をサポートしていません。
CREATE SERVER 文 この文は、SYSSERVERS システム・テーブルにサーバを追加するために使用します。	Windows CE ではプロキシ・テーブルをサポートしていないため、この文をサポートしていません。
CREATE WRITEFILE 文 (旧式) この文は、データベースのライト・ファイルを作成するために使用します。	Windows CE ではライト・ファイルをサポートしていないため、この文をサポートしていません。
DROP DATABASE 文 この文は、データベースに関連するすべてのデータベース・ファイルを削除するために使用します。	Windows CE では、DROP DATABASE 文はサポートしていません。
DROP SERVER 文 この文は、Adaptive Server Anywhere カタログからリモート・サーバを削除するために使用します。	Windows CE ではリモート・データ・アクセスをサポートしていないため、リモート・サーバをサポートしていません。したがって、DROP SERVER 文 は Windows CE ではサポートされていません。
INSTALL JAVA 文 この文は、データベース内で Java クラスを使用できるようにします。	Windows CE では、データベースで Java がサポートされていません。したがって、INSTALL JAVA 文 はサポートされていません。

SQL 文	考慮事項
REMOVE JAVA 文　この文は、クラスまたは JAR ファイルをデータベースから削除するために使用します。クラスを削除すると、それはカラムまたは変数型として使用できなくなります。クラスまたは JAR は事前にインストールしておきます。	Windows CE では、データベースで Java がサポートされていません。したがって、REMOVE JAVA 文 はサポートされていません。
REORGANIZE TABLE 文　この文は、データベースへの連続アクセスという要件があるために、データベース全体の再構築ができない場合に、テーブルの断片化を解除するために使用します。	この文は、Windows CE ではサポートされていません。
RESTORE DATABASE 文　この文は、アーカイブからバックアップされたデータベースをリストアするために使用します。	アーカイブ・ファイルは、データベース・バックアップのアーカイブ方式を使用して作成されますが、これは Windows CE でサポートされていません。したがって、RESTORE DATABASE 文 は Windows CE ではサポートされていません。
START JAVA 文　この文は、Java 仮想マシンを起動するために使用します。	Windows CE では、データベースで Java がサポートされていません。したがって、START JAVA 文 はサポートされていません。
STOP JAVA 文　この文は、Java 仮想マシンを停止するために使用します。	Windows CE ではデータベースで Java がサポートされていません。したがって、STOP JAVA 文 はサポートされていません。

以下の SQL 文は、Windows CE では機能が異なるか、機能に制限があります。

SQL 文	考慮事項
ALTER DATABASE 文　この文は、旧バージョンのソフトウェアで作成したデータベースをアップグレードするため、または任意のデータベースに Java または jConnect サポート機能を追加するために使用します。	ALTER DATABASE UPGRADE 構文は、データベースに Java または ALTER DATABASE UPGRADE のサポートを追加するために使用します。Windows CE ではデータベースで Java をサポートしていないため、ALTER DATABASE 文をサポートしていません。
BACKUP 文　この文は、データベースとトランザクション・ログをバックアップするために使用します。	Windows CE でサポートされているのは、BACKUP DATABASE DIRECTORY 構文のみです。
CREATE DATABASE 文　この文は、データベースを作成するために使用します。データベースはオペレーティング・システム・ファイルとして格納されます。	CREATE DATABASE 文は、Windows CE ではサポートされていません。 ただし、CREATE DATABASE 文はデスクトップ・マシンのデータベースを初期化できるため、これをあとで Windows CE デバイスにコピーできます。
CREATE EVENT 文　この文は、イベントまたは、イベントに関連付けて定義済みアクションを自動化するイベント・ハンドラを定義するために使用します。また、スケジュールされたアクションを定義するときにも使用します。	DiskSpace イベント・タイプは、Windows CE ではサポートされていません。 ただし、この文は GlobalAutoIncrement または ServerIdle イベント・タイプの定義に使用できます。

SQL 文	考慮事項
CREATE FUNCTION 文　この文は、データベース内に新しい関数を作成するために使用します。	EXTERNAL NAME 句を使用する関数は、外部ライブラリ関数呼び出しのラッパです。外部バックアップは、Windows CE ではサポートされていません。したがって、この句はサポートされていません。 ただし、データベースでユーザ定義の SQL 関数を作成する場合は、Windows CE でも CREATE FUNCTION 文を使うことができます。
CREATE TABLE 文　この文は、データベースに新しいテーブルを作成するために使用します。また、オプションでリモート・サーバ上にもテーブルを作成できます。	AT 句は、別のサーバにリモート・テーブルを作成し、そのリモート・テーブルにマップするプロキシ・テーブルを現在のデータベース上に作成します。 Windows CE ではプロキシ・テーブルをサポートしていないため、この文の AT 句をサポートしていません。 AT 句を除き、この文は Windows CE でもサポートされています。

Windows CE でサポートされていない Sybase Central のウィザード

Sybase Central は、Adaptive Server Anywhere データベースのさまざまな設定、プロパティ、ユーティリティをグラフィカル・ユーザ・インタフェースを介して使用するデータベース管理ツールです。この項では、Windows CE でサポートされていないウィザードや、機能が異なるウィザードについて説明し、可能な場合はその代わりとなる方法を紹介します。

ウィザード	考慮事項
[データベース・バックアップ] ウィザード　このウィザードは、データベース・ファイルとトランザクション・ログ・ファイルのバックアップを含んだ 1 つのファイルを作成します。この種類のバックアップは、「アーカイブ・バックアップ」と呼ばれます。	アーカイブ・バックアップは、Windows CE ではサポートされていません。したがって、[データベース・バックアップ] ウィザードはサポートされていません。 詳細については、『ASA データベース管理ガイド』> 「バックアップの種類」を参照してください。 Sybase Central には、[バックアップ・イメージ作成] ウィザードもあります。これは、データベース・ファイルとトランザクション・ログ・ファイルの別々のバックアップを作成します。このウィザードは、Windows CE でもサポートされています。 詳細については、『ASA データベース管理ガイド』> 「[バックアップ・イメージ作成] ウィザードを使用したデータベースのバックアップ」を参照してください。
[ログ・ファイル設定の変更] ウィザード　[ログ・ファイル設定の変更] ウィザードでは、データベースに関連するトランザクション・ログまたはトランザクション・ログ・ミラーの名前を表示、変更できます。また、データベースがトランザクション・ログやミラーを管理するのを停止したり、開始したりできます。	このウィザードは、ログ・ファイルが格納されている Windows CE ディレクトリにはマップできません。したがって、[ログ・ファイル設定の変更] ウィザードはサポートされていません。

ウィザード	考慮事項
[データベース圧縮] ウィザード (非推奨)　[データベース圧縮] ウィザードは、特定のデータベース・ファイルを読み取って圧縮データベース・ファイルを作成します。通常、圧縮データベース・ファイルは元のサイズの 40 ～ 60% になります。	データベースの圧縮は、Windows CE ではサポートされていません。したがって、[データベース圧縮] ウィサードは Windows CE ではサポートされていません。
[データベース作成] ウィザード　[データベース作成] ウィザードでは、データベースを初期化 (作成) できます。初期化するときに多数のデータベース属性が指定されます。これらの属性は、データベース全体のアンロード、再初期化、再構築以外の方法では、後から変更することはできません。	Sybase Central を実行しているデスクトップ・マシンに Windows CE サービスがインストールされている場合は、このウィザードに Windows CE データベースの作成に特有の機能が備わっています。 詳細については、「Windows CE データベースの作成」69 ページを参照してください。
[データベース消去] ウィザード　このウィザードは、データベースとそれに関連するトランザクション・ログ・ファイルを削除するために使用します。	実行中のデータベースにこのウィザードを実行することはできません。Windows CE のデータベースをデスクトップ・マシンから管理するには、Windows CE デバイスのネットワーク・サーバでデータベースが実行している必要があります。 このウィザードは、ディレクトリへのマップにも関連しています。Windows CE デバイスのディレクトリは、このウィザードからはマップできません。したがって、[データベース消去] ウィサードは Windows CE ではサポートされていません。

ウィザード	考慮事項
[サービス作成] ウィザード [サービス作成] ウィザードは、Adaptive Server Anywhere のサービスを作成、削除、変更するために使用するツールです。アプリケーションを Windows サービスとしてインストールすると、ログオフしても実行できます。	Windows CE は、アプリケーションをサービスとして実行するように設定できません。したがって、[サービス作成] ウィサードは Windows CE ではサポートされていません。
[ログ・ファイル変換] ウィザード [ログ・ファイル変換] ウィザードでは、トランザクション・ログを SQL コマンド・ファイルに変換できます。	このウィザードは、ディレクトリへのマップに関連しています。Windows CE デバイスのディレクトリは、このウィザードからはマップできません。したがって、[ログ・ファイル変換] ウィザードは Windows CE ではサポートされていません。
[データベース展開] ウィザード [データベース展開] ウィザードでは、圧縮ユーティリティや [データベース圧縮] ウィザードで作成した圧縮データベース・ファイルを展開できます。このウィザードは、圧縮されたファイルを読み込んで、データベース・ファイルを展開した状態にリストアします。	データベースの圧縮は、Windows CE ではサポートされていません。したがって、[データベース展開] ウィザードは Windows CE ではサポートされていません。
[データベース・アンロード] ウィザード [データベース・アンロード] ウィザードでは、データベースをアンロードし、指定したディレクトリにデータ・ファイル一式を入れることができます。このウィザードは、データベースの再構築に使用する Interactive SQL コマンド・ファイルを作成します。また、各テーブルのすべてのデータをカンマ区切りの形式で、指定したディレクトリ内のファイルにアンロードします。	このウィザードは、データベース・ファイルが格納されている Windows CE ディレクトリにはマップできません。したがって、[データベース・アンロード] ウィザードはサポートされていません。 ただし、Windows CE のデータベースをデスクトップ・マシンにコピーすると、[データベース・アンロード] ウィザードを使用してアンロードできます。

ウィザード	考慮事項
[データベース・アップグレード] ウィザード　[データベース・アップグレード] ウィザードを使用すると、データベースが古いバージョンの Adaptive Server Anywhere から新しいバージョンにアップグレードされ、新しいリリースのすべての機能を利用できるようになります。	このウィザードは、Windows CE ではサポートされていません。ただし、Windows CE のデータベースをデスクトップ・マシンにコピーし、このウィザードを使ってから Windows CE デバイスにコピーし直すと、アップグレードできます。 詳細については、『ASA データベース管理ガイド』> 「[データベース・アップグレード] ウィザードを使用したデータベースのアップグレード」を参照してください。

第 5 章

SQL Anywhere Studio がサポートしているプラットフォーム

この章の内容

この章では、SQL Anywhere Studio の各コンポーネントがサポートしているプラットフォームについて説明します。

サポートされるプラットフォームについては、弊社営業担当までお問い合わせください。

有効性
サポートされているプラットフォームとしてリストされていても、そのソフトウェアが現在必ずしも使用できるとは限りません。たとえば、特定のソフトウェア・バージョンがすべてのプラットフォームで同時にリリースされるとは限りません。リリースされていないソフトウェアの機能とプラットフォームのサポートは、変更される場合があります。

概要

SQL Anywhere Studio は、さまざまな種類のオペレーティング・システムに対して、データベース管理テクノロジと同期テクノロジを提供します。その主なコンポーネントは、次のとおりです。

- **Adaptive Server Anywhere** SMB (Small and Medium Business) データベース・サーバからデスクトップ・データベース・アプリケーション、および Windows CE デバイスにいたるまで、幅広く利用できるリレーショナル・データベース・システム。
- **Ultra Light** 特に小型デバイス用に設計されているリレーショナル・データベース。Palm OS デバイスと Windows CE デバイス上で動作します。
- **Mobile Link** Adaptive Server Anywhere または Ultra Light の多数のデータベースと統合データベース・サーバ間の、双方向同期を実現する同期システム。
- **SQL Remote** Adaptive Server Anywhere データベースと Sybase Adaptive Server Enterprise データベースの同期システム。
- **管理ツール** SQL Anywhere Studio のすべてのコンポーネントを管理するためのツール・セット。

このマニュアルには、SQL Anywhere Studio のどのコンポーネントをどのオペレーティング・システムで実行できるかを示す表が含まれています。情報は、SQL Anywhere Studio 内のコンポーネント別に編成されています。

この章の内容は、マニュアル作成時の情報です。最新情報については、弊社営業担当までお問い合わせください。

概要

次の表は、主なコンポーネントがサポートしているプラットフォームをまとめたものです。この表には、各プラットフォームでサポートされる個々の機能の違いや、サポートされる UNIX の種類やバージョンなどの詳細情報は示されていません。詳細については、この後に示す表を参照してください。

UNIX プラットフォームのサポートには 2 種類あります。

- **通常版** Sun Solaris と Linux では、SQL Anywhere Studio のリリースを完全に使用できます。
- **クライアント／サーバ限定版** クライアント／サーバ限定版には、核となる Adaptive Server Anywhere が含まれています。Mobile Link 同期サーバが含まれることもあります。クライアント／サーバ限定版のプラットフォームは、HP-UX、IBM AIX、Compaq Tru64、64 ビット Linux の各オペレーティング・システムです。

リストにあるオペレーティング・システムでサポートされているバージョンの詳細については、「オペレーティング・システムのバージョン」149 ページを参照してください。

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Mac OS X	UNIX (通常版)	UNIX (限定版)	NetWare	Windows CE	Palm OS
Adaptive Server Anywhere データベース・サーバ	√	√	√	√	√	√	√	
Adaptive Server Anywhere クライアント	√	√	√	√	√	√[1]	√	
Ultra Light データベース	√[2]						√	√
Ultra Light 開発	√							
Mobile Link 同期サーバ	√	√	√	√	√[3]			
SQL Remote	√	√	√	√		√	√	
Replication Agent	√	32 ビット・ソフトウェア		√				
管理ツール	√	32 ビット JRE	√	√		√[4]		

1. Embedded SQL のみ
2. 開発とテストのみ
3. PA RISC (9.0.1 以降) 上の 32 ビット HP-UX サポートと AIX サポートのみ
4. 限定されたセット

Windows オペレーティング・システムと NetWare オペレーティング・システム

次の表は、SQL Anywhere Studio のさまざまなコンポーネントおよび機能がサポートしているオペレーティング・システムを示します。

特に指定のないかぎり、リストにあるオペレーティング・システムでサポートされているバージョンは、「オペレーティング・システムのバージョン」149 ページに記載されているとおりです。

Adaptive Server Anywhere

システムの稼働条件は、データベースのサイズ、負荷、必要なパフォーマンスによって異なります。以下に示すのは、最低限の要件です。

Adaptive Server Anywhere (パーソナル・データベース・サーバとネットワーク・データベース・サーバの両方) は、わずか 4 MB のメモリで動作します。なお、1 つのクライアント接続ごとに 8 KB の追加メモリが必要です。メモリ容量を増やすとパフォーマンスは大幅に向上します。データベース内で Java を使用する場合、Adaptive Server Anywhere には 5 MB の追加メモリが必要になります。使用するコンピュータには、このメモリ容量に加えて、オペレーティング・システム用のメモリ容量が必要になります。

コンポーネントまたは機能	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
パーソナル・データベース・サーバ	√	√		
ネットワーク・データベース・サーバ	√	√	√	√
ODBC クライアント	√	√	√[1]	
OLE DB クライアント	√	√	√	
Embedded SQL クライアント	√	√	√	√
Open Client クライアント	√	√		
JDBC クライアント (jConnect)	√	32 ビット・ソフトウェア	√	

コンポーネントまたは機能	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
JDBC クライアント (iAnywhere JDBC ドライバ)	√	√		
SQL プリプロセッサ	√	32 ビット・ソフトウェア		
データベース内の Java	√			√
リモート・データ・アクセス[2]	√	√		√[3]
外部ストアド・プロシージャ	√	√[4]		√
動的なキャッシュ・サイズの変更	√	√		
強力な暗号化	√	√	√	√
CREATE DATABASE 文、RESTORE DATABASE 文、DROP DATABASE 文	√	√		√
SPX プロトコル	√			√

1. ODBC ドライバ・マネージャは使用できない場合があります。その場合には、ODBC ドライバに直接リンクする必要があります。
2. Windows 上では iAnywhere Solutions ODBC ドライバが必要です。サポートされているプラットフォームのリストについては、「iAnywhere Solutions ODBC ドライバがサポートしているプラットフォーム」145 ページを参照してください。
3. データベース内の Java と JDBC ドライバが必要です。
4. 外部ストアド・プロシージャの以前のインタフェースは、32 ビット ソフトウェアでのみ使用できます。

Ultra Light 開発プラットフォーム

コンポーネント	Windows NT/2000/XP	Windows (64 ビット)	Windows 98 SE	Windows CE	NetWare
SQL プリプロセッサ (Embedded SQL のみ)	√		√		

コンポーネント	Windows NT/2000/XP	Windows (64 ビット)	Windows 98 SE	Windows CE	NetWare
Ultra Light ジェネレータ (静的型 C++ API、静的型 Java API)	√		√		
Ultra Light for AppForge MobileVB	√				
Ultra Light ActiveX	√				
Native Ultra Light for Java	√				
Ultra Light.NET	√[1]				
Ultra Light for C++	√				

1. Visual Studio .NET または Visual Studio.NET 2003 と .NET Compact Framework バージョン 1.0.5000 以降が必要です。

32 ビット・ソフトウェアのほとんどは 64 ビット Windows のエミュレーション環境で機能しますが、静的型 Java API は 64 ビット Windows では使用できません。

配備プラットフォームの詳細については、「Ultra Light ターゲット・プラットフォーム」147 ページを参照してください。

開発環境

Palm OS 用 Ultra Light アプリケーションは、次のいずれかの開発環境を使用して構築できます。

- Metrowerks CodeWarrior バージョン 8 または 9。

  CodeWarrior には Palm SDK が含まれています。ターゲットとする特定のデバイスによって異なりますが、この開発ツールに含まれている Palm SDK をより新しいバージョンにアップグレードすることをおすすめします。

- AppForge MobileVB または Crossfire (Ultra Light MobileVB コンポーネントを使用)

Mobile Link 同期サーバ

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
Mobile Link 同期サーバ	√	√		
TCP/IP 同期	√	√		
HTTPS 同期	√	√		
HTTP 同期	√	√		
トランスポート・レイヤ・セキュリティ	√			
Java 同期論理	√	32 ビット・ソフトウェア		
.NET 同期論理	√	√		
メッセージ・サンプル付き Mobile Link (QAnywhere)	√	√		
サーバ起動同期	√	√		

Mobile Link Adaptive Server Anywhere クライアント

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
ASA Mobile Link クライアント (*dbmlsync*)	√	√	√	
TCP/IP 同期	√	√	√	
HTTP 同期	√	√	√	
HTTPS 同期	√	√	√	
トランスポート・レイヤ・セキュリティ	√	√	√	

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
QAnywhere クライアント	√	√	√	

Mobile Link ユーティリティ

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
リダイレクタ	√[1]			
Mobile Link 抽出ユーティリティ (*mlxtract*)	√	√		

1. Windows NT/2000/XP のみ

SQL Remote

SQL Remote コンポーネントが Adaptive Server Enterprise データベースに対して動作するためには、Sybase Open Client ライブラリか Open Server ライブラリまたはその両方が必要です。

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
ASA Message Agent (*dbremote*)	√	√	√	√
ASE Message Agent (*ssremote*)	√	32 ビット・ソフトウェア		
ASE ステーブル・キュー (*ssqueue*)	√	32 ビット・ソフトウェア		
ASA 抽出ユーティリティ (*dbxtract*)	√	√		
ASE 抽出ユーティリティ (*ssxtract*)	√	32 ビット・ソフトウェア		
ファイル・メッセージ・タイプ	√	√	√	√
FTP メッセージ・タイプ	√	√	√	√
MAPI メッセージ・タイプ	√	√		

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
SMTP メッセージ・タイプ	√	√	√	√
VIM メッセージ・タイプ	v			

Replication Agent　Replication Agent を使用するには、Sybase Open Client ライブラリと Open Server ライブラリが必要です。

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
Replication Agent	√	32 ビット・ソフトウェア		

管理ツール

コンポーネント	Windows (32 ビット)	Windows (64 ビット)	Windows CE	NetWare
コマンド・ライン管理ユーティリティ	√	√		√
Sybase Central	√	32 ビット JRE		
Interactive SQL	√	32 ビット JRE		√ [1]
DBConsole	√	32 ビット JRE		√ [2]

1. ほかのオペレーティング・システムの場合よりも限定されたバージョン
2. ほかのオペレーティング・システムの場合よりも限定されたバージョン

コマンド・ライン管理ユーティリティと NetWare で提供されるツール以外の管理ツールは、Java 2 Runtime Environment バージョン 1.4.1 を使用します。

次の段落は、Java 2 SDK Standard Edition マニュアルからの抜粋で、管理ツールに適用されます。

Java ェ 2 SDK は、Intel ハードウェア上で動作する、Windows 95、Windows 98 (Second Edition を含む)、Windows NT 4.0 (Service Pack 5 以降を適用)、Windows Me、Windows 2000 Professional、Windows 2000 Server、Windows 2000 Advanced Server、Windows XP オペレーティング・システムで使用できます。

Pentium 166 MHz 以上のプロセッサが必要です。GUI アプリケーションの実行には、32M バイト以上の物理 RAM が必要です。Java Plug-in 製品を使用するブラウザで動作するアプレットの実行には、48M バイト以上の RAM の使用をおすすめします。メモリが少ない状態で実行すると、ディスクのスワップが発生し、パフォーマンスに重大な影響を与える場合があります。非常に大きなプログラムを十分なパフォーマンスで実行するには、より多くの RAM が必要になる場合があります。

ディスクに 70M バイトの空き領域を確保してから、Java 2 SDK ソフトウェアをインストールしてください。付属のマニュアルも一緒にインストールする場合は、さらに 120M バイトの空き領域が必要です。

UNIX、Linux、Macintosh のオペレーティング・システム

特に指定のないかぎり、リストにあるオペレーティング・システムでサポートされているバージョンは、「オペレーティング・システムのバージョン」149 ページに記載されているとおりです。

Adaptive Server Anywhere

コンポーネント	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
パーソナル・データベース・サーバ	√	√	√	√	√	√
ネットワーク・データベース・サーバ	√	√	√	√	√	√
ODBC クライアント [1]	√	√	√	√	√	√
Embedded SQL クライアント	√	√	√	√	√	√
SQL プリプロセッサ	√	√	√	√	√	√
Open Client クライアント	√	√	√	√	√	√
JDBC クライアント (jConnect ドライバ)	√	√	√	√	√	√
データベース内の Java	√	√	√	√	√	√
リモート・データ・アクセス [2]	√	√	√			
外部ストアド・プロシージャ	√	√	√	√	√	√
iAnywhere JDBC ドライバ[3]	√	√	√	√	√	√

1. ODBC ドライバ・マネージャは使用できない場合があります。その場合には、ODBC ドライバに直接リンクする必要があります。

2. JDBC リモート・データ・ソースを使用するには、データベース内の Java と JDBC ドライバが必要です。ODBC リモート・データ・ソースを使用するには、iAnywhere Solutions ODBC ドライバが必要です。サポートされているプラットフォームのリストについては、「iAnywhere Solutions ODBC ドライバがサポートしているプラットフォーム」145 ページを参照してください。
3. Itanium チップの場合は JRE 1.4 以上が必要です。その他のチップの場合は JRE 1.3 以上が必要です。

Linux で Adaptive Server Anywhere を使用する場合の注意 複数のプロセッサを使用しているときに、2.4.2 SMP カーネルの環境でクラッシュが発生したことがあります。状況によっては、プロセス・メモリが壊れて、アプリケーションがクラッシュする可能性もあります。最新の 2.4.x カーネルでは、この問題は発生していません。

Mobile Link 同期サーバ

コンポーネント	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Mobile Link 同期サーバ	√	√	√	√	√	
TCP/IP 同期	√	√	√	√	√	
HTTPS 同期	√	√	√	√	√	
HTTP 同期	√	√	√	√	√	
トランスポート・レイヤ・セキュリティ	√	√	√			
Java 同期論理	√	√	√	√	√	
.NET 同期論理						
メッセージ・サンプル付き Mobile Link (QAnywhere)	√	√	√	√	√	
サーバ起動同期	√	√	√	√	√	

Mobile Link の HP-UX サポートは 32 ビット PA RISC のみで、バージョン 9.0.1 から利用可能です。Mac OS X の TLS サポートも 9.0.1 以降です。

HP-UX または AIXJava で Java 同期ロジック、メッセージ・サンプル付き Mobile Link、またはサーバ起動同期を使用している場合は、SHLIB_PATH (on HP-UX) または LIBPATH (AIX 上) を設定して、インストールした JRE を格納するディレクトリを含める必要があります。

Adaptive Server Anywhere Mobile Link クライアント

コンポーネント	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
ASA Mobile Link クライアント (*dbmlsync*)	√	√	√			
TCP/IP 同期 (クライアント)	√	√	√			
HTTPS 同期 (クライアント)	√	√	√			
HTTP 同期 (クライアント)	√	√	√			
Mobile Link 抽出ユーティリティ (*mlxtract*)	√	√	√			

SQL Remote

SQL Remote コンポーネントが Adaptive Server Enterprise データベースに対して動作するためには、Sybase Open Client ライブラリか Open Server ライブラリまたはその両方が必要です。これらのライブラリは Compaq Tru64 では使用できません。

コンポーネント	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
ASA Message Agent (*dbremote*)	√	√	√			
ASE Message Agent (*ssremote*)	√	√				
ASE ステーブル・キュー (*ssqueue*)	√	√				

コンポーネント	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
ASA 抽出ユーティリティ (*dbxtract*)	√	√	√			
ASE 抽出ユーティリティ (*ssxtract*)	√	√				
ファイル・メッセージ・タイプ	√	√	√			
FTP メッセージ・タイプ	√	√	√			
MAPI メッセージ・タイプ						
SMTP メッセージ・タイプ	√	√	√			
VIM メッセージ・タイプ						

Replication Agent

Replication Agent を使用するには、Sybase Open Client ライブラリと Open Server ライブラリが必要です。

コンポーネント	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Replication Agent	√	√				

管理ツール

コンポーネント	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
コマンド・ライン管理ユーティリティ	√	√	√	√	√	√
Sybase Central	√	√	√			
Interactive SQL	√	√	√			

コマンド・ライン管理ユーティリティ以外の管理ツールは、Java 2 Runtime Environment バージョン 1.3.1 を使用します。

次の段落は、Java 2 SDK Standard Edition バージョン 1.3.1 マニュアルで説明されている Solaris のシステム条件からの抜粋で、管理ツールに適用されます。

Java 2 SDK Standard Edition v. 1.3.1 (J2SDK 1.3.1) は、Solaris 2.6、Solaris 7、Solaris 8、Solaris 9 オペレーティング環境で使用できます。

Java 2 SDK をインストールする前に、このリリースをサポートするために必要なすべてのパッチ・セットをインストールしてください。パッチを入手するには、SunSolve サポート Web サイトを参照してください。この Web サイトには、各 Solaris オペレーティング環境プラットフォーム用のパッチ・クラスタが用意されています。各パッチ・クラスタは、指定のプラットフォームでサポートされている Java 2 Standard Edition (J2SE) のすべてのバージョンに適用されます。

システムで使用するフォント・パッケージについては、「Solaris Font Package Requirements」も参照してください。

Mobile Link 同期統合データベース

Mobile Link は、次の統合データベースをサポートします。

データベース・サーバ	バージョン
Adaptive Server Anywhere	現在のバージョン
Sybase Adaptive Server Enterprise	11.5 以上
Oracle	8i と 9i
Microsoft SQL Server	7 と 2000
IBM DB2	7 と 8 の UDB

依存関係

サポートされるオペレーティング・システムのリストについては、「Mobile Link 同期サーバ」135 ページと「Mobile Link 同期サーバ」140 ページを参照してください。

Mobile Link 同期サーバが統合データベースに接続するには、ODBC ドライバが必要です。iAnywhere Solutions では、「iAnywhere Solutions ODBC ドライバがサポートしているプラットフォーム」145 ページに記載した各種 ODBC ドライバを用意しています。データベース・ベンダによっては、独自の ODBC ドライバを提供しています。

ODBC ドライバとその状況のリストについては、http://www.ianywhere.com/developer/technotes/odbc_mobilink.html にアクセスして「*Recommended ODBC Drivers for MobiLink*」を参照してください。

iAnywhere Solutions ODBC ドライバがサポートしているプラットフォーム

iAnywhere Solutions ODBC ドライバを使用することで、Mobile Link 同期サーバと Adaptive Server Anywhere リモート・データ・アクセスについて、サードパーティ・データベースへの ODBC アクセスが実現します。

Adaptive Server Anywhere ODBC ドライバは、この項にはリストされていません。Adaptive Server Anywhere ODBC ドライバの詳細については、「Windows オペレーティング・システムと NetWare オペレーティング・システム」132 ページを参照してください。

iAnywhere Solutions ODBC ドライバは、次のオペレーティング・システムとデータベースの組み合わせで使用できます。

オペレーティング・システム	データベース
Windows (32 ビット)	Sybase Adaptive Server Enterprise 11.5 以上 Oracle 8i と 9i IBM DB2 7 と 8 UDB
Solaris 7.0、8.0、9.0	Sybase Adaptive Server Enterprise 11.5 以上 Oracle 8i と 9i
IBM AIX	Sybase Adaptive Server Enterprise 11.5 以上 Oracle 8i と 9i IBM DB2 7 と 8 UDB
Linux [1]	Sybase Adaptive Server Enterprise 11.5 以上 Oracle 8i と 9i

オペレーティング・システム	データベース
HP-UX (32 ビット PA RISC のみ)	Adaptive Server Anywhere IBM DB2 7 と 8 UDB

[1] Red Hat、SuSE、Caldera の各ディストリビューションのみ。

Ultra Light ターゲット・プラットフォーム

次の表は、同期ストリームを含む Ultra Light の配備がサポートされているオペレーティング・システムを示します。

サポートされている開発プラットフォームのリストについては、「Ultra Light 開発プラットフォーム」133 ページを参照してください。

カラムの見出しの説明については、「オペレーティング・システムのバージョン」149 ページを参照してください。

コンポーネント	Windows CE	Palm	Windows XP
Ultra Light 静的型 C/C++ API	√	√	√
Ultra Light Embedded SQL	√	√	√
Ultra Light 静的型 Java API[1]	√		√
Ultra Light for AppForge MobileVB	√	√	√
Ultra Light ActiveX	√		√
Ultra Light.NET	√[2]		√[3]
Native Ultra Light for Java	√[4]		√
Ultra Light C++ API コンポーネント	√	√	√
Ultra Light for M-Business Anywhere	√[5]	√[6]	
TCP/IP 同期	√	√	√
HTTP 同期	√	√	√
HTTPS 同期	√	√	√
HotSync 同期		√	
ActiveSync 同期 (3.5 と 3.6)	√[7],[8]		

コンポーネント	Windows CE	Palm	Windows XP
HTTP または TCP/IP におけるトランスポート・レイヤ・セキュリティ [9]	√[10]	√	√

1. PersonalJava または JDK 1.2 が必要です。
2. .NET Compact Framework バージョン 1.0.3705 以上が必要です。
3. .NET Compact Framework バージョン 1.0.5000 以上が必要です。
4. CE/ARM デバイスに導入するには、デバイスに Java VM が必要です。Native Ultra Light for Java は、PersonalJava 1.2 (またはそれ以上) ベースの VM と使用できます。PersonalJava ベースの Jeode VM と CrEme VM (バージョン 3.24 以上) だけはテスト済みです。
5. ARM プロセッサ上で動作する Pocket PC が必要です。
6. Palm OS 5 が必要です。
7. Native Ultra Light for Java と Ultra Light.NET 以外の Ultra Light コンポーネントではサポートされません。
8. Smartphone 2002 ではサポートされません。
9. Ultra Light コンポーネントではサポートされません。静的型 Java API の場合は JDK 1.2.2 以上が必要です。JDK 1.4 はサポートされません。
10. Pocket PC が必要です。

注意

Ultra Light の動的メモリ条件は、Palm OS 3.5 が搭載されているデバイス、またはメモリが 4 MB 未満のデバイスの場合、非常に小さなデータベース・スキーマでないかぎり動作しない可能性があることを意味しています。

オペレーティング・システムのバージョン

有効性
リリースされていないソフトウェアの機能とプラットフォームのサポートは、変更される場合があります。サポートされているプラットフォームとしてリストされていても、そのソフトウェアが現在必ずしも使用できるとは限りません。

たとえば、特定のソフトウェア・バージョンがすべてのプラットフォームで同時にリリースされるとは限りません。また、古いリリースに記載された新しいプラットフォームに関する情報が正しいという保証はありません。

特に指定のないかぎり、オペレーティング・システム名は次のバージョンに対応します。

オペレーティング・システム	9.0.0	9.0.1	9.0.2
Microsoft Windows (32 ビット)	次のバージョンがサポートされます。 • Windows NT 4.0 以上 • Windows 2000 • Windows XP。x86 アーキテクチャ・プロセッサ上の Windows XP Embedded と Windows XP Tablet PC を含みます。 • Windows XP Embedded イメージで必要なコンポーネントのリストについては、http://www.sybase.com/detail?id=1019835 を参照してください。 • Windows Server 2003 (32 ビット・バージョン) • Windows 95/98/Me	9.0.0 から変更なし	9.0.1 から変更なし

オペレーティング・システム	9.0.0	9.0.1	9.0.2
Microsoft Windows (64 ビット)	Windows Server 2003 (64 ビット・バージョン)。	9.0.0 から変更なし	9.0.1 から変更なし
Microsoft Windows CE	次のいずれかのプロセッサ上で動作する Windows CE 3.0 オペレーティング・システム。 • MIPS プロセッサ • ARM プロセッサ。XScale シリーズを含みます。 • x86 プロセッサ。Ultra Light では、Pocket PC 2002 エミュレータ用に x86 プロセッサをサポートしています。Adaptive Server Anywhere では、x86 デバイスとエミュレーションをサポートしています。 開発を目的とした Windows CE エミュレータの使用もサポートされています。 Windows CE 3.0 では、Pocket PC 2002 を含む Pocket PC や、Handheld PC をサポートします。 ARMV4 モードの XScale (Intel PXA255) プロセッサ上で動作する Windows CE 4.1。 ARMV4 モードの XScale (Intel PXA255) プロセッサ上で動作する Pocket PC 2003 と他のバージョンの Windows CE 4.2。	"Thumb" モードの ARM プロセッサ (ARMV4T) 上で動作する Windows CE 4.1 をサポートしています (バージョン 9.0.1 から)。 Ultra Light では、Smartphone 2002 プラットフォーム、他の Windows CE プラットフォームをサポートしています。 Pocket PC 2000 エミュレータはサポートされていません。	9.0.1 から変更なし
Novell NetWare	5.1 (Service Pack 6 を適用)、6.0 (Service Pack 3 を適用)、6.5。	9.0.0 から変更なし	9.0.1 から変更なし

オペレーティング・システム	9.0.0	9.0.1	9.0.2
Palm OS	Palm OS バージョン 3.5、4.x、5.0 を実行している Palm Computing Platform デバイス。	9.0.0 から変更なし	9.0.1 から変更なし

オペレーティング・システム	9.0.0	9.0.1	9.0.2
Linux	Linux ディストリビューションは、カーネル、C ライブラリ (glibc)、ncurses ライブラリの定義されているバージョンにすべて基づいています。サポートされるバージョンは次のとおりです。 • **カーネル** 2.4.18 または 2.4.2 • **glibc** 2.1.2、2.1.3、または 2.2 • **ncurses** 4.2 次の Linux ディストリビューションは上記の条件を満たしています。 • **Red Hat** 6.1、6.2、7.0、7.1、7.2、7.3、または Advanced Server 2.1 • **SuSE** 6.3、6.4、7.0、7.2、7.3、または 8.0 (7.1 を除く) • **Caldera** 2.4 または 3.e • **TurboLinux** 6.1、6.5、または 7.0 • **Mandrake** 7.2、8.0、または 8.1 Linux ディストリビューションは、x86 アーキテクチャのプロセッサ上でサポートされます。 64 ビットの展開版は、デュアル Itanium 2 プロセッサを使用している Red Hat Advanced Server で使用できます。	次のバージョンのカーネルと glibc を搭載した 32 ビット Linux ディストリビューション。 • **カーネル** 2.4.2 ～ 2.4.21 と glibc 2.2.1 ～ 2.3.2 • **カーネル** 2.6.0 と glibc 2.3.2 次のバージョンのカーネルと glibc を搭載した Itanium チップの 64 ビット Linux ディストリビューション。 • **カーネル** 2.4.9 と glibc 2.2.4 (たとえば、RedHat Advanced Server 2.1) • **カーネル** 2.6.0 と glibc 2.4.9 (たとえば、RedHat Enterprise Server 3)	**このバージョンのリリース前に変更される可能性もあります。**

オペレーティング・システム	9.0.0	9.0.1	9.0.2
Sun Solaris	7.0 [1]、8.0、9.0 の 32 ビットまたは 64 ビットのオペレーティング・システム (SPARC プロセッサを使用)。	9.0.0 から変更なし	このバージョンのリリース前に変更される可能性もあります。
Mac OS X	Mac OS X バージョン 10.2.n。n は 2 以上。	バージョン 10.2 と 10.3 (G4 プロセッサを使用)。 バージョン 10.3 (G5 プロセッサを使用)。	このバージョンのリリース前に変更される可能性もあります。
Compaq Tru64	4.0D、5.1、5.2 (Alpha プロセッサを使用)。	4.2、5.1、5.2 (Alpha プロセッサを使用)。	このバージョンのリリース前に変更される可能性もあります。
Hewlett Packard HP-UX	11.0 または 11i (32 ビット・バージョン、PA-RISC プロセッサを使用)。HP-UX 11i は、64 ビット・バージョン (デュアル Itanium 2 プロセッサを使用) でもサポートされます。	9.0.0 から変更なし	このバージョンのリリース前に変更される可能性もあります。
IBM AIX	4.3.3、5.1、5.2 (Power PC プロセッサを使用)。	9.0.0 から変更なし	このバージョンのリリース前に変更される可能性もあります。

1. 報告されている SIGIO 問題を回避するには、パッチ 106541-14 とパッチ 106980-13 の適用をおすすめします。

第 3 部　データベースの使用

第 3 部では、リレーショナル・データベースのモデルとデータ操作の基本について説明します。

第 6 章

データベース・アプリケーションのアーキテクチャ

この章の内容

この項では、リレーショナル・データベースについて説明する場合に重要な用語と概念をいくつか説明します。

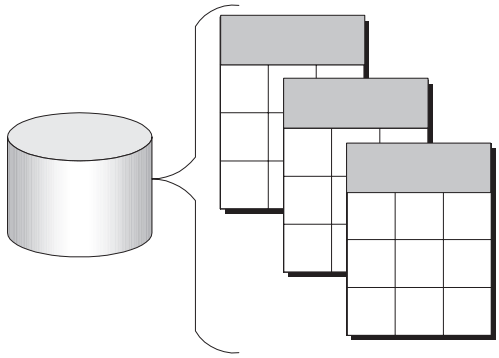
リレーショナル・データベースの概念

「リレーショナル・データベース管理システム」(RDBMS) は、データの格納と検索を行うシステムで、データは相互に関係するテーブルに格納されています。

SQL Anywhere Studio には、2 つのリレーショナル・データベース・システムがあります。「**Adaptive Server Anywhere**」はフル装備の主要な RDBMS で、多くのクライアントを持つホストのネットワーク・データベース・サーバから、コンパクトな組み込みデータベースに至るまで、幅広い用途を持っています。Ultra Light は、専有容量の小さいリレーショナル・データベースです。「**Ultra Light**」の配備テクノロジにより、専有容量の小さいデバイス上でも Adaptive Server Anywhere の機能を利用できます。

データベース・テーブル

リレーショナル・データベースでは、すべてのデータは「**テーブル**」に格納されます。テーブルは「**ロー**」と「**カラム**」で構成されます。



各テーブルには 1 つ以上のカラムがあり、各カラムには整数、文字列 (テキスト)、日付など、特定の「データ型」が割り当てられます。テーブル内の各ローにはカラムごとに 1 つの値が入っています。

たとえば、従業員の情報を格納したテーブルは次のようになります。

emp_ID	emp_lname	emp_fname	emp_phone
10057	Huong	Zhang	1096
10693	Donaldson	Anne	7821

リレーショナル・データベースの特性

リレーショナル・データベースのテーブルには、いくつかの重要な特性があります。

- カラムまたはローの順序に意味はない。
- 各ローのカラムには値が 1 つだけ入るか、NULL が入る。NULL は、そのカラムに値がないことを示します。
- 1 つのカラムのすべての値は同じデータ型である。

次の表に、テーブルとその内容を説明する公式のまたは非公式のリレーショナル・データベース用語のいくつかを、非リレーショナル・データベースで使用されている同等の用語とあわせて示します。このマニュアルでは非公式の用語を使用します。

非公式なリレーショナル用語	公式なリレーショナル用語	非リレーショナル用語
テーブル	関係	ファイル
カラム	属性	フィールド
ロー	組	レコード

各テーブル内に入れる項目

データベース内の各テーブルには、従業員、製品、または顧客など、特定の事柄に関する情報が保持されます。

このような設計によって、情報の冗長性を排し、冗長性によって整合性が失われないようなデータベースを構築できます。たとえば、営業と会計の両方の部署で、顧客についての情報の入力や検索を行う可能性があります。リレーショナル・データベースでは、顧客についての情報は、両方の部署からアクセスできるテーブルに一度格納されるだけです。

データベース設計の詳細については、『ASA SQL ユーザーズ・ガイド』> 「データベースの設計」を参照してください。

テーブル間の関係

別々のテーブルにある情報間の関係を記述するには、プライマリ・キーと外部キーを使用します。「**プライマリ・キー**」はテーブル内の各ローをユニークに識別するためのキーで、「**外部キー**」は別々のテーブルにあるローの間の関係を定義します。

プライマリ・キーと外部キーを使用すると、リレーショナル・データベースで冗長性を排して効率的な方法で情報を保持できます。

テーブルにはプライマリ・キーがある

リレーショナル・データベース内の各テーブルには、「**プライマリ・キー**」が必要です。プライマリ・キーとは 1 つのカラム、またはカラムのセットで、各ローをユニークに識別します。1 つのテーブル内で、2 つのローが同じプライマリ・キーの値を持つことはありません。

例

サンプル・データベースでは、employee テーブルに従業員の情報が格納されています。このテーブルにはプライマリ・キー・カラム emp_id があり、各従業員に割り当てられたユニークな ID 番号が入っています。ID 番号が入っている単一のカラムにプライマリ・キーを割り当てるのが一般的な方法です。名前やその他の識別子は、必ずしもユニークではないため、ID 番号のほうが適しています。

サンプル・データベースの sales_order_items テーブルのプライマリ・キーは、より複雑です。このテーブルには、会社からの注文に含まれる個々の項目に関する情報が入っており、次のカラムで構成されています。

- **id** その注文項目が含まれている注文を識別する注文番号
- **line_id** 注文内での注文項目を識別する行番号
- **prod_id** 受注した製品を識別する製品 ID
- **quantity** 受注した項目の数量
- **ship_date** 受注品が出荷された日付

特定の注文項目はその注文項目が含まれている注文番号と、注文内での行番号によって識別されます。この 2 つの番号は、id カラムと line_id カラムに格納されています。注文項目が同一の id 値を共有する場合 (1 回の注文に複数の注文項目がある場合) と、注文項目が同一の line_id 番号を共有する場合 (各注文における最初の注文項目はすべて line_id が 1) があります。どの注文項目も、これら 2 つの値を共有することはないので、プライマリ・キーは 2 つのカラムで構成されています。

テーブルは外部キーによって関連付けられる

1 つのテーブル内の情報は、「**外部キー**」キーによって他のテーブルの情報と関連付けられます。

例

サンプル・データベースには、従業員情報が入ったテーブル (employee) が 1 つと、部署情報の入ったテーブル (department) が 1 つあります。department テーブルには、次のカラムがあります。

- **dept_id**　部署の ID 番号。これがテーブルのプライマリ・キーになります。
- **dept_name**　部署の名前
- **dept_head_id**　部長の従業員 ID

特定の従業員の所属部署名を探せるように、その従業員の部署名を employee テーブルに入力しておく必要はありません。その代わり employee テーブルには、department カラムの dept_id 値の 1 つと一致する番号の入ったカラムがあります。

employee テーブルの dept_id カラムは、department テーブルに対する外部キーと呼ばれます。外部キーは、対応するプライマリ・キーを持つテーブル内の特定のローを参照します。

この例では、employee テーブル (関係付けの外部キーを持つ) を「**外部テーブル**」または「**参照元テーブル**」といいます。department テーブル (参照先のプライマリ・キーを持つ) を「**プライマリ・テーブル**」または「**参照先テーブル**」といいます。

その他のデータベース・オブジェクト

リレーショナル・データベースは、単なる関連テーブルのセット以上のものがあります。リレーショナル・データベースには、次のオブジェクトもあります。

- **インデックス**　インデックスを使用すると、情報をすばやく検索できます。概念的には、データベース内のインデックスは本の索引と同じです。本の場合、索引は、索引にリストされたそれぞれの用語と、その語が記載されているページを関連付けています。データベースの場合、インデックスは、インデックス付けされたカラムの値と、その値を含むデータのローが格納されている物理的なロケーションを関連付けます。

  インデックスは、パフォーマンスを高めるための重要な設計要素です。通常は明示的に作成しますが、プライマリ・キー、外部キー、ユニーク・カラムのインデックスは自動的に作成されます。作成されたインデックスは、ユーザに対して透過的に機能します。

- **ビュー**　ビューとは、計算から導き出されるテーブル、つまり仮想テーブルです。ビューはクライアント・アプリケーションからはテーブルのように見えますが、データは入っていません。ビューにアクセスすると、その中の情報は基本となるテーブルから計算されます。

  実際に情報が保持されるテーブルは、ビューと区別するために「**ベース・テーブル**」と呼ぶことがあります。ビューは、ベース・テーブルまたはその他のビューに対する SQL クエリによって定義付けられます。

- **ストアド・プロシージャとトリガ**　これらは、データベースそのものに保持され、データベース内の情報に作用するルーチンです。

  独自のストアド・プロシージャを作成して名前を付けると、特定のデータベース・クエリを実行したり、他のデータベース・タスクを実行したりすることができます。ストアド・プロシージャにはパラメータを指定できます。たとえば、プロシージャの呼び出しのパラメータに指定した値よりも多い金額を使った顧客名をすべて返すといったストアド・プロシージャを作成できます。

トリガとは、ユーザがデータを更新、削除または挿入すると自動的に呼び出される特別なストアド・プロシージャで、更新、削除、挿入のどの場合に起動するかは、トリガをどのように定義したかによって異なります。トリガは、テーブル自体またはテーブル内のカラムと対応させます。トリガは、データベース内で自動的にビジネス規則の管理を行う場合に有用です。

- **ユーザとグループ**　データベースの各ユーザは、ユーザ ID とパスワードを所有しています。ユーザごとにパーミッションを設定できるため、機密情報の漏洩や、ユーザに認可されていない変更の実行を防げます。パーミッションの管理を容易にするために、ユーザをグループに割り当てることができます。

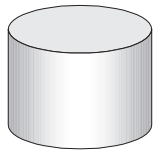
- **Java オブジェクト**　データベースに Java クラスをインストールできます。Java クラスは、ロジックをデータベースに組み込む強力な方法を提供でき、情報を保持するための特別なユーザ定義データ型のクラスを使用できます。

「リレーショナル・データベース管理システム」(RDBMS) は、これらのすべての項目から成り立っています。RDBMS はデータの格納と検索を行うシステムで、データは相互に関係するテーブルに格納されています。

データベース・システムの各部分

リレーショナル・データベース管理システムは、次の要素で構成されています。

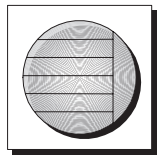
データベース　データはデータベース内に格納されます。このマニュアルでは、データベースは次のように示します。



Adaptive Server Anywhere データベースは、通常は拡張子 *.db* が付いた 1 つのファイルです。Adaptive Server Anywhere にはサンプル・データベースが含まれており、ユーザはこれを操作できます。このサンプル・データベースはファイル *asademo.db* で、Adaptive Server Anywhere のインストール・ディレクトリ内に入っています。

データベース・サーバ　データベース・サーバが、データベースを管理します。その他のアプリケーションは、直接データベース・ファイルを扱わずに、データベース・サーバとの通信を介してデータベースにアクセスします。

このマニュアルでは、データベース・サーバは次のように示します。



Adaptive Server Anywhere には、「**パーソナル・データベース・サーバ**」と「**ネットワーク・データベース・サーバ**」の 2 つのデータベース・サーバがあります。パーソナル・サーバの機能に加えて、ネットワーク・データベース・サーバは、ネットワークを介したクライアント／サーバ通信もサポートしています。一方、パーソナル・データベー

ス・サーバが扱うことができるのは、同じマシン上で稼働するアプリケーションからの接続のみです。2 つのサーバの要求処理エンジンはまったく同じです。

プログラミング・インタフェース　アプリケーションはプログラミング・インタフェースを通じてデータベース・サーバと通信します。ODBC、JDBC、OLE DB、Sybase Open Client、または Embedded SQL を使用できます。

多くのアプリケーション開発ツールは、下層部分のインタフェースをユーザが気にせずに使用できるように、それぞれ独自のプログラミング環境を提供しています。たとえば、Sybase PowerBuilder を使用してアプリケーションを開発する場合、ODBC 関数呼び出しをする必要はまったくありません。しかし、ユーザが認識していなくても、これらのツールはプログラミング・インタフェースの 1 つを使用しています。

プログラミング・インタフェースは、データベースと通信するための一連の関数呼び出しライブラリを提供します。ODBC と JDBC の場合、ライブラリは通常「**ドライバ**」と呼ばれます。通常、ライブラリは UNIX オペレーティング・システム上では共有ライブラリとして、また PC オペレーティング・システム上ではダイナミック・リンク・ライブラリ (DLL) として提供されます。JDBC インタフェースは、Sybase jConnect ドライバを使用します。このドライバはコンパイルされた Java クラスの zip ファイルです。

このマニュアルの図表では、プログラミング・インタフェースは次のように示します。

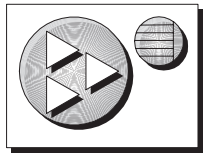
クライアント・アプリケーション　クライアント・アプリケーションは、プログラミング・インタフェースの 1 つを使用してデータベース・サーバと通信します。

Sybase PowerBuilder などの RAD (rapid application development) ツールを使用してアプリケーションを開発する場合、そのツールによってデータベース・サーバと通信するための独自の方法が提供されてい

て、言語インタフェースの細かい機能をユーザが意識することはありません。しかし、どのアプリケーションも、サポートされているインタフェースの 1 つを使用します。

このマニュアルの図では、クライアント・アプリケーションは次のアイコンで示されています。

Ultra Light データベース・サーバは、各 Ultra Light アプリケーションに適した形で生成され、アプリケーションの一部として組み込まれます。データベース・サーバと共に使用する Ultra Light アプリケーションは、次のように示されます。



各部分を組み合わせる方法

データベース・アプリケーションは、アプリケーションと同じマシン上にあるデータベース・サーバと接続できます。ネットワーク・データベース・サーバの場合は、別のマシン上にあるデータベース・サーバと接続できます。さらに、Adaptive Server Anywhere では、分散データベースを構築して、異なったマシン上にある物理的に別個のデータベース間でデータを共有できます。

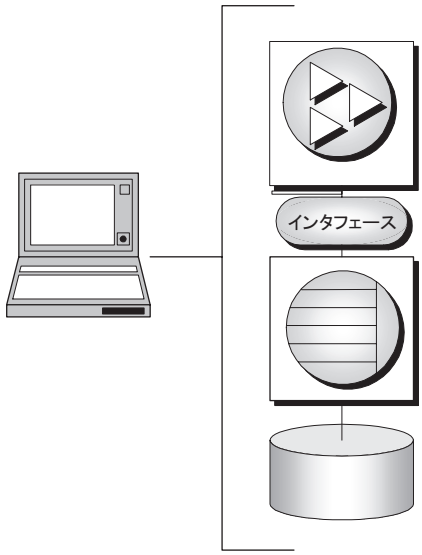
パーソナル・アプリケーションと組み込みデータベース

Adaptive Server Anywhere を使用して、1 台のコンピュータ上にアプリケーションとデータベースの完全なセットを構築できます。このセットの最も簡単な配置が「**スタンドアロン・アプリケーション**」、つまり「**パーソナル・アプリケーション**」です。このアプリケーションは独立型であり、ほかのデータベースには接続しません。この場合、データベース・サーバとデータベースをクライアント・アプリケーションから起動できます。一般に、このようなデータベースを「**組み込みデータベース**」と呼び、エンド・ユーザからはアプリケーションの一部のように見えます。データベース・サーバが、組み込みデータベースとして使用される場合、これを「**データベース・エンジン**」と呼ぶこともあります。

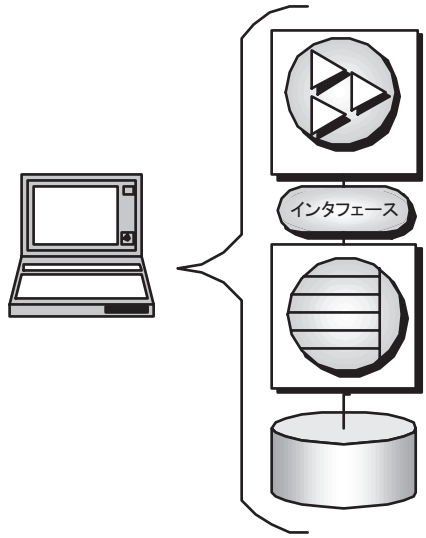
多くのリレーショナル・データベース管理システムでは、熟練したスタッフによる管理が求められます。組み込みデータベースの特徴は、管理をまったく必要とせずに運用できることです。

Adaptive Server Anywhere パーソナル・データベース・サーバは、通常スタンドアロン・アプリケーションとして使用します。クライアント・アプリケーションは、プログラミング・インタフェースを介して同じマシン上のデータベース・サーバと接続します。

スタンドアロン・アプリケーションのアーキテクチャを次に示します。クライアント・アプリケーションは、プログラミング・インタフェースを介して同じマシン上で動作するデータベース・サーバと接続しています。

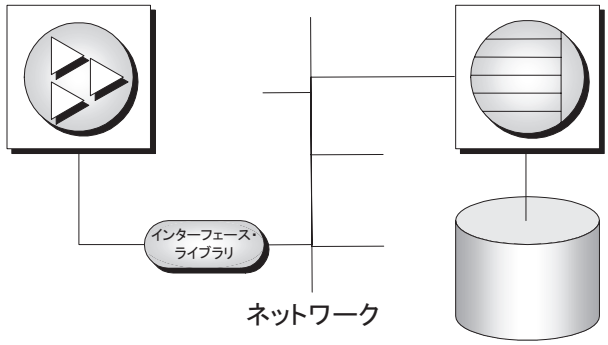


Adaptive Server Anywhere パーソナル・データベース・サーバは、通常スタンドアロン・アプリケーションに使用しますが、ネットワーク・サーバと同じマシン上で、アプリケーションを使用することもできます。



クライアント／サーバ・アプリケーションとマルチユーザ・データベース

Adaptive Server Anywhere を使用して、動作している多くのアプリケーションが組み込まれた別々のマシンをデータベース・サーバにネットワークを経由して接続する環境を構築できます。これを「**クライアント／サーバ**」または「**マルチユーザ・データベース**」環境と呼び、アーキテクチャは次のとおりです。インタフェース・ライブラリは各クライアント・マシン上にあります。



データベース・サーバとしては Adaptive Server Anywhere ネットワーク・データベース・サーバを使用し、ネットワーク通信をサポートしています。

クライアント・アプリケーションをクライアント／サーバ環境で使用するのに必要なのは、接続するサーバを指定することだけです。

多層コンピューティングのアーキテクチャ

多層コンピューティングでは、アプリケーション論理は、データベース・サーバとクライアント・アプリケーションの中間に位置するアプリケーション・サーバ (Sybase EAServer など) に格納されます。多くの場合、1 つのアプリケーション・サーバは、非リレーショナルなデータ・ストアのほかに、複数のデータベースにアクセスします。インターネットの場合、クライアント・アプリケーションはブラウザ・ベースであり、アプリケーション・サーバは、通常、Web サーバの拡張機能です。

Sybase EAServer は、アプリケーション論理をコンポーネントとして格納し、このコンポーネントをクライアント・アプリケーションから利用できるようにします。利用できるコンポーネントは、PowerBuilder コンポーネント、Java beans、または COM コンポーネントです。

アプリケーション・サーバは、クライアント・アプリケーションにトランザクション論理も提供し、一連のオペレーションが複数のデータベース上で確実にアトミックに実行されるようにします。Adaptive Server Anywhere は、多層コンピューティングに最適であり、Microsoft Distributed Transaction Coordinator (DTC) で管理されている分散トランザクションにサーバとして使用できます。Sybase Enterprise Application Server と Microsoft Transaction Server のどちらも、DTC を使用してクライアント・アプリケーションにトランザクション・サービスを提供します。

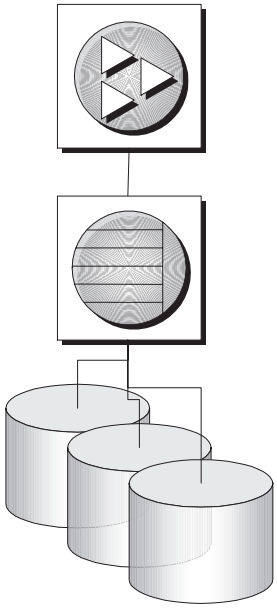
詳細については、『ASA プログラミング・ガイド』> 「3 層コンピューティングと分散トランザクション」を参照してください。

複数のデータベースの使用

この項では、2 つ以上のデータベースを使う場合に使用する、前述の Adaptive Server Anywhere アーキテクチャ拡張機能について説明します。

1 つのデータベース・サーバで複数のデータベースを実行する

Adaptive Server Anywhere のパーソナル・データベース・サーバとネットワーク・データベース・サーバの両方とも、複数のデータベースを同時に管理できます。アプリケーションからの 1 つの接続では 1 つのデータベースしか実行できません。しかし、1 つのデータベース・サーバを経由して、複数のアプリケーションが別々のデータベースに個別の接続をしたり、1 つのアプリケーション・セットから別々のデータベースに接続したりできます。

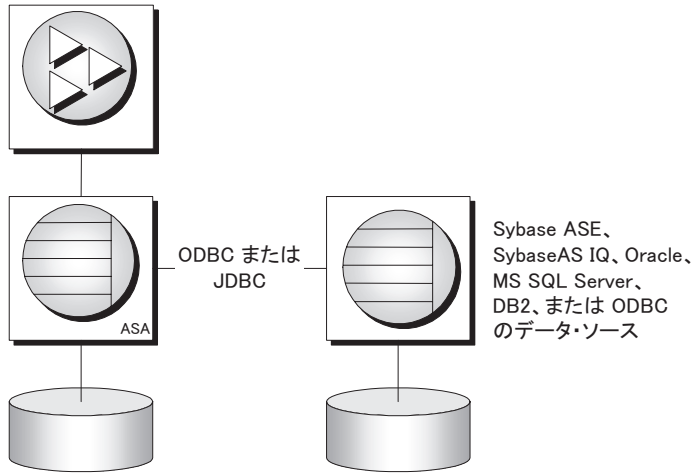


データベースは、データベース・サーバの起動時に開始できます。また、接続パラメータ DatabaseFile を使用してデータベースに接続しても開始できます。

詳細については、『ASA データベース管理ガイド』> 「データベース・サーバ」と 『ASA データベース管理ガイド』> 「DatabaseFile 接続パラメータ [DBF]」を参照してください。

ほかのデータベースのデータへのアクセス

Adaptive Server Anywhere Remote Data Access 機能を使用して、複数のデータベース・サーバ上にあるデータベースにアクセスしたり、また、同じサーバ上にあるほかの複数のデータベースにもアクセスできます。上記のアーキテクチャの図では、アプリケーションは 1 つのデータベースに接続していますが、リモート・サーバを定義すると、リモート・データベースにあるプロキシ・テーブルを、現在接続しているデータベースにあるのと同じように使用できます。



詳細については、『ASA SQL ユーザーズ・ガイド』> 「リモート・データへのアクセス」を参照してください。

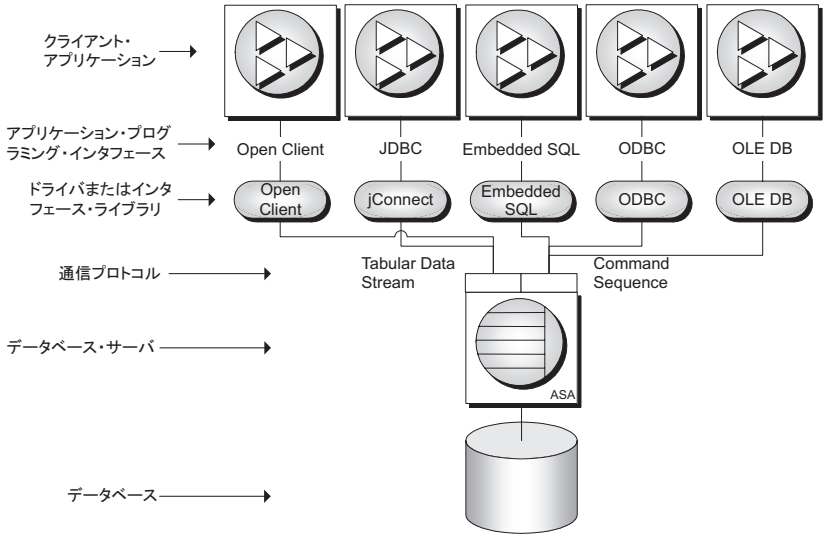
アプリケーション・プログラミング・インタフェース

この項では、アプリケーション・アーキテクチャについて詳しく説明します。データベース・アプリケーションのアーキテクチャの概要については、「各部分を組み合わせる方法」167 ページで説明しました。

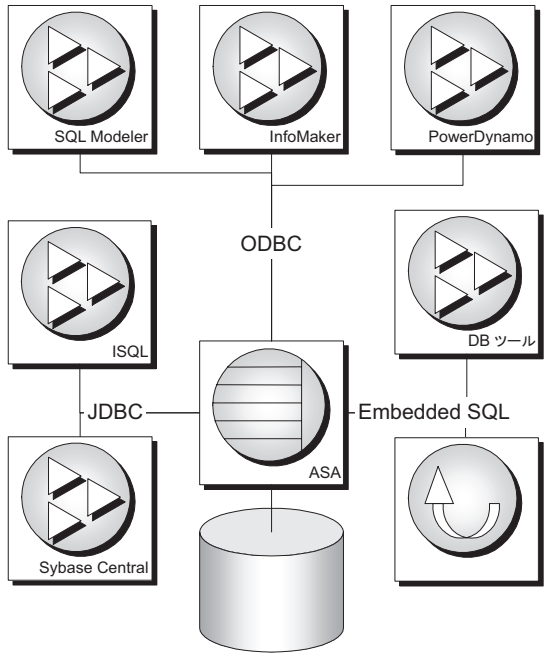
Adaptive Server Anywhere は、さまざまなプログラミング・インタフェースをサポートしており、使用するアプリケーションとアプリケーション開発環境を自由に選択できます。

サポートされるインタフェースとプロトコル

次の図はサポートされるインタフェースと使用されるインタフェース・ライブラリを表示しています。多くの場合、インタフェース・ライブラリとインタフェースの名前は同じです。



SQL Anywhere Studio に付属のアプリケーションは、これらのインタフェースのいくつかを使用します。



通信プロトコル

各インタフェース・ライブラリは、1 つの「**通信プロトコル**」を使用してデータベース・サーバと通信します。Adaptive Server Anywhere は、「**Tabular Data Stream（TDS)**」と「**Command Sequence**」という 2 つの通信プロトコルをサポートしています。これらのプロトコルは内部に組み込まれているので、ほとんどの場合どちらを使用しても変わりがありません。開発環境は、プロトコルではなくて利用できるツールによって異なります。

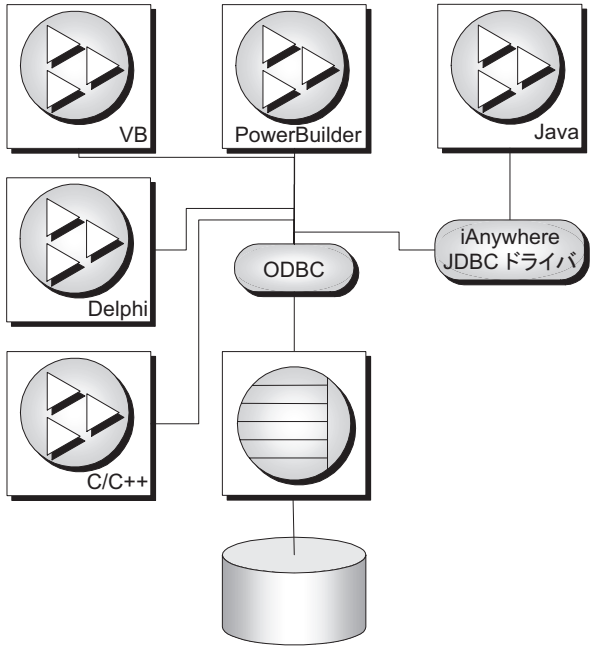
2 つのプロトコルの主な相違は、データベースと接続してみるとわかります。Command Sequence を使用するアプリケーションと TDS を使用するアプリケーションは、データベースとデータベース・サーバを識別する方法が異なるので、接続時の対話が異なります。

TDS　このプロトコルは、Sybase Adaptive Server Enterprise、Open Client アプリケーション、および TDS を使用して jConnect JDBC ドライバで接続する Java アプリケーションに使用されています。

Command Sequence　このプロトコルは、Adaptive Server Anywhere、Adaptive Server IQ で使用されています。また、Embedded SQL、ODBC、OLE DB の各アプリケーションでも使用されます。

ODBC アプリケーション

次の図に示すように、ODBC アプリケーションは、さまざまな開発ツールとプログラミング言語を使用して開発できます。

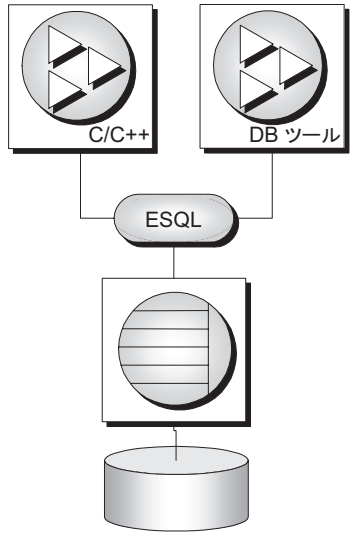


たとえば、SQL Anywhere Studio に付属のアプリケーションのうち、InfoMaker と SQL Modeler は ODBC を使用してデータベースと接続します。

Embedded SQL アプリケーション

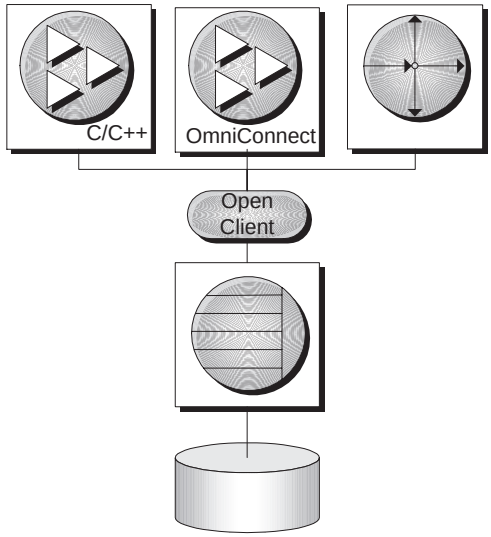
Embedded SQL プログラミング・インタフェースを使用して、C または C++ のアプリケーションを開発できます。たとえば、コマンド・ライン・データベース・ツールは、このような方法で開発されたアプリケーションです。

Embedded SQL は、Ultra Light アプリケーションのプログラミング・インタフェースとしても使用されます。



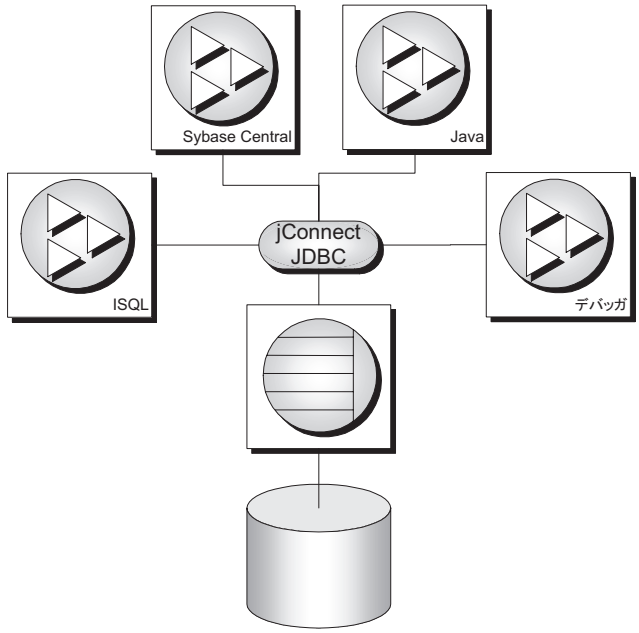
Open Client アプリケーション

Open Client も、Adaptive Server Enterprise でサポートされているインタフェースです。C または C++ で Open Client アプリケーションを開発できます。その他の Sybase アプリケーションの中にも OmniConnect や Replication Server のように Open Client を使用しているものがあります。



JDBC アプリケーション

Java を使って、Adaptive Server Anywhere に接続するために JDBC を使用してアプリケーションを開発できます。ストアド・プロシージャ・デバッガ、Sybase Central、Interactive SQL など、SQL Anywhere Studio に付属して提供されるアプリケーションのいくつかは JDBC を使用しています。



また、Java と JDBC は Ultra Light アプリケーションを開発するための重要なプログラミング言語です。

OLE DB アプリケーション

Adaptive Server Anywhere には OLE DB ドライバが組み込まれています。Microsoft の OLE DB インタフェースを直接使用するか、または ActiveX Data Objects (ADO) インタフェースを使用して、アプリケーションを開発できます。ADO インタフェースは、Visual Basic やその他の Microsoft プログラミング・ツールで提供されています。SQL Anywhere Studio には、Adaptive Server Anywhere 用の ADO.NET データ・プロバイダも含まれています。

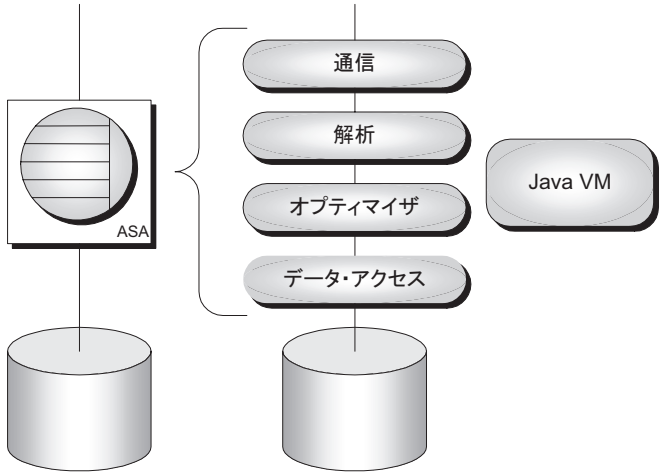
Adaptive Server Anywhere の内部構造

データベース・サーバの内部構造を変更する必要はありませんが、その概要を知っておくとそのプロセスをよく理解できます。

データベース・サーバの内部構造

Adaptive Server Anywhere データベース・サーバは、多くの要求を効率的に処理できる内部構造となっています。

- 通信レイヤが、クライアント・アプリケーションと実際のデータ交換を処理します。このレイヤがクライアント・アプリケーションから要求を受信して、その結果を返します。この動作のタイミングは、ネットワーク・トラフィックを最小に保ちながら、データができるだけ早くクライアント側に届くように、クライアントとサーバのネゴシエーションによって決定されます。
- パーサが、データベース・サーバに送信された各 SQL 文をチェックして、それを内部フォームに変換して処理します。
- 要求がクエリ、または更新や削除を行う文の場合は、データにアクセスするいろいろな方法があり、アクセスする時間もそれによってかなり異なる可能性があります。オプティマイザは、このように多くのアクセス方法から、必要なデータをすばやく取得する最良のルートを選択します。
- データベース・サーバには Java 仮想マシン (VM) が組み込まれていて、クライアント・アプリケーションから送信された Java オペレーション、またはデータベース・サーバによって内部で呼び出された Java オペレーションはすべて、Java VM で処理されます。
- データベース・サーバの最下位レベルがディスクからデータの読み込みと書き込みを行い、不必要なディスク・アクセスを避けるためにデータをメモリにキャッシュし、複数のユーザからの要求のバランスを調整します。



データベースの内部構造

Adaptive Server Anywhere データベースにあるすべての情報は、通常、単一のデータベース・ファイルに格納されていて、1 台のマシンから別のマシンにコピーできます。複数のファイルでデータベースを構成することもできますが、非常に大きいデータベースでなければ通常は必要ありません。

Adaptive Server Anywhere は、データベース・ファイルの他にデータベースの実行時に、2 つのファイルを使用します。この 2 つのファイルとは、トランザクション・ログとテンポラリ・ファイルです。

- **データベース・ファイル**　データベース・ファイルは、内部的には複数のページ、つまり一定サイズのディスク領域で構成されています。データ・アクセス・レイヤは、一度に 1 ページのデータの読み込みと書き込みをします。多くのページは、データベース・テーブルにあるデータを保持していますが、インデックス情報やデータベース内のデータの分散についての情報などを保持しているページもあります。

- **トランザクション・ログ**　トランザクション・ログは、データベースで実行されたすべての操作の記録を含む独立したファイルです。通常、このログ・ファイルの名前は、サフィックスが

.db ではなく *.log* で終わる点を除いてデータベース・ファイルと同じです。このファイルには、次のような 3 つの重要な機能があります。

- **リカバリを可能にするためにデータに対する操作を記録する** データベース・ファイルが破損した場合は、トランザクション・ログとバックアップからデータベースを再作成できます。
- **パフォーマンスを向上させる** トランザクション・ログに情報を書き込むことで、データベース・サーバは、データベース・ファイルに頻繁に書き込まなくても、安全にステートメントを処理できます。
- **データベースのレプリケーションを可能にする** SQL Remote と Mobile Link クライアント・ユーティリティは、このファイルを使用してネットワークに常時接続されているとは限らないポータブル・コンピュータ上に、データベースの変更をレプリケートできます。

- **テンポラリ・ファイル** テンポラリ・ファイルは、データベース・サーバが起動すると同時に開き、サーバが停止すると閉じます。その名前が示すように、テンポラリ・ファイルとはサーバの起動中に一時的に情報を保持するために使用されます。テンポラリ・ファイルには、セッション間にわたって維持する必要がある情報は格納されません。

  テンポラリ・ファイルはテンポラリ・ディレクトリに格納されます。このディレクトリのロケーションは、通常、TEMP 環境変数によって示されます。

第 7 章

データベース・テーブルからのデータの選択

この章の内容

この章では、データベースのデータを検索する SQL (Structured Query Language) とクエリについて説明します。また、1 つのテーブルのデータを検索する方法についても説明します。

このマニュアルのほかのパートは、この章で説明する概念を理解していることを前提に書かれています。

SQL とデータベース・コンピューティング

クライアント・アプリケーションは、クエリを使用して情報を検索したり、テーブルに新しいローを挿入したりするデータベース・タスクを実行する場合に、「**SQL**」(**Structured Query Language**) 文を使用します。SQL とは、規格制定機関である ANSI と ISO によって標準化されたリレーショナル・データベース言語です。

クライアント・アプリケーションの開発方法次第では、プログラミング言語からの関数呼び出しで SQL 文を使用することができます。アプリケーション開発ツールの中には、SQL 文を作成するためのユーザ・インタフェースを提供しているものもあります。

SQL 文は、プログラミング・インタフェースによってデータベース・サーバに配信されます。データベース・サーバは文を受信するとこれを実行し、必要な情報 (クエリの結果など) をアプリケーションに戻します。

クライアント／サーバの通信プロトコルはクライアント・アプリケーションとデータベース・サーバ間で情報を搬送し、プログラミング・インタフェースはアプリケーションの情報の送信方法を定義します。どのインタフェースを使用しても、また、どのネットワーク・プロトコルを使用しても、サーバに送信されるのは SQL 文で、クライアント・アプリケーションに返されるのも SQL 文の結果です。

例

次の SQL 文は、サンプル・データベースの employee テーブルからすべての従業員の姓を抽出します。

```
SELECT emp_lname
 FROM employee
```

このようなクエリを Interactive SQL を使用してデータベース・サーバに送信したり、独自のアプリケーションに構築したりできます。

例

次の SQL 文は、社員食堂の食品の種類と在庫量をリストするテーブル food を作成します。

```
CREATE TABLE food (
     food_id integer primary key,
     food_type char(20),
     quantity integer
 )
```

クエリ

"SQL" の "Q" は「**クエリ**」を表しています。SELECT 文を使用して、データベースからデータの問い合わせ (**検索**) をします。リレーショナル・システム内の基本的なクエリ・オペレーションは、射影、制限、ジョインです。これらのオペレーションは、すべて SELECT 文によって実行できます。

射影と制限

「**射影**」とは、テーブル内のカラムのサブセットです。「**制限**」(**選択**とも言う) は、いくつかの条件に基づいたテーブル内のローのサブセットとのことです。

たとえば、次の SELECT 文では、価格が $15 より高い製品すべての名前と価格が検索されます。

```
SELECT name, unit_price
 FROM product
 WHERE unit_price > 15
```

このクエリでは、射影 (`SELECT name, unit_price`) と制限 (`WHERE unit_price > 15`) を使用しています。

ジョイン

ジョインは、各テーブルのカラムの値を比較して、2 つ以上のテーブル内のローをリンクします。たとえば、1 ダースを超える数が出荷されたすべての注文項目について、注文項目 ID 番号と製品名を次のように選択できます。

```
SELECT sales_order_items.id, product.name
 FROM product JOIN sales_order_items
 WHERE sales_order_items.quantity > 12
```

product テーブルと sales_order_items テーブルは、この両テーブル間の外部キー関係に基づいてジョインされます。

その他の SQL 文

SQL を使用すると、クエリだけでなく、さらに多くのことを実行できます。SQL には、テーブル、ビュー、その他のデータベース・オブジェクトを作成する文があります。また、テーブルを修正する文 (INSERT、UPDATE、DELETE 文) や、このマニュアルで説明する他の数多くのデータベース・タスクを実行する文も含まれています。

システム・テーブル

すべての Adaptive Server Anywhere データベースには、システム・テーブルのセットがあります。このテーブルは、データとシステムを管理するためにシステムが使用する特別なテーブルです。このテーブルはまた、「**データ辞書**」または「**システム・カタログ**」とも呼ばれます。

システム・テーブルには、データベースに関する情報が入っています。システム・テーブルは、他のテーブルを変更するのと同じ方法で直接変更することは絶対にしないでください。システム・テーブルには、データベース内のテーブル、データベースのユーザ、各テーブル内のカラムなどに関する情報が入っています。この情報は、データに関するデータ、つまり「**メタデータ**」です。

Adaptive Server Anywhere システム・テーブルの詳細については、『ASA SQL リファレンス・マニュアル』> 「システム・テーブル」を参照してください。

SELECT 文

アプリケーション (クライアント) とデータベース・サーバ間のすべての対話は、データベース・サーバに SQL 文を送信し、データベース・サーバからクライアントに情報が返されることで行われます。

SELECT 文は、クライアント・アプリケーションで使用する情報をデータベースから取り出します。SELECT 文は「**クエリ**」とも呼ばれます。情報は、結果セットとしてクライアントに配信されます。クライアント・アプリケーションは、配信された結果セットを処理できます。たとえば、Interactive SQL では、結果セットが [結果] ウィンドウ枠に表示されます。結果セットは、データベースのテーブルと同様に一連のローで構成されます。

多数のテーブルから特定の情報を検索するアプリケーションでは、SELECT 文はかなり複雑になる場合があります。この章では、簡単な SELECT 文のみを使用しています。より高度なクエリについては、後のチュートリアルで説明します。

SELECT 文の完全な構文については、『ASA SQL リファレンス・マニュアル』> 「SELECT 文」を参照してください。

注意

この章のそれぞれの例は、コンピュータ上で Adaptive Server Anywhere ソフトウェアを実行しながら読み進めることをおすすめします。

個々の例では、Interactive SQL へのコマンド入力の指示と、コンピュータ画面に何が表示されるかについての説明があります。ソフトウェアが実行できない状況でチュートリアルを読んでも SQL について学ぶことはできます。ただし、さまざまな操作を自分なりに試みることはできません。

各例では、Interactive SQL を起動してサンプル・データベースに接続していることを前提としています。

テーブル全体の選択

最も単純な SELECT 文は、1 つのテーブルのデータをすべて取り出すものです。この SELECT 文の構文は、次のとおりです。

SELECT * FROM *table-name*

table-name は、問い合わせるテーブルの名前に置き換えてください。アスタリスク (*) は、すべてのカラムのリストを表す省略形です。

❖ 会社が販売しているすべての製品をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT * FROM Product
```

この SELECT 文は、product テーブルのローとカラムをすべて取り出して、[結果] ウィンドウ枠の [結果] タブに表示します。

id	name	description	size	color	quantity	unit_price
300	Tee Shirt	Tank Top	Small	White	28	9
301	Tee Shirt	V-neck	Medium	Orange	54	14
302	Tee Shirt	Crew Neck	One size fits all	Black	75	14
400	Baseball Cap	Cotton Cap	One size fits all	Black	112	9
…	…	…	…	…	…	…

Product テーブルには 7 つのカラムがあります。各カラムには色、ID などの名前があります。会社が販売している製品ごとに 1 つずつローがあり、各ローの各カラムに値が 1 つずつ入っています。たとえば、ID が 301 の製品は、T シャツです。これはサイズ M の V ネックで、色はオレンジです。

注意

- **テーブル名には大文字と小文字の区別がない**　テーブル名には大文字の P を使って Product と指定しましたが、実際のテーブル名はすべて小文字です。Adaptive Server Anywhere では、文字列の比較で大文字と小文字を区別するかどうかはデータベースの作成時に指定できますが、テーブル名やカラム名などの識別子を使用するときは大文字と小文字を区別しません。

  データベースの作成については、『ASA SQL ユーザーズ・ガイド』> 「データベースの作成」または『ASA データベース管理ガイド』> 「初期化ユーティリティ」を参照してください。

- **SQL キーワードは大文字と小文字が区別されない**　**SELECT** の代わりに **select** または **Select** と入力できます。このマニュアルでは、SQL キーワードには一般に大文字を使用しています。

- **改行は重要ではない**　複数の文すべてを 1 行に入力することも、各行の末尾で [Enter] キーを押して改行することもできます。SQL 文のなかには、SELECT 文などのように、「句」と呼ばれるいくつかの部分で構成されているものもあります。多くの例では、各句が読みやすいように 1 行ずつになっているので、通常、このマニュアルの例では文が次のように記述されています。

  ```
  SELECT *
   FROM product
  ```

- **結果セットのローの順序に意味はない**　データベースからローが返される順序に保証はなく、またその順序に意味はありません。ローを特定の順序で取り出したい場合は、クエリで順序を指定する必要があります。

演習

サンプル・データベース内の、employee、customer、contact、sales_order など、その他のテーブルを問い合わせてみてください。

テーブルからのカラムの選択

SELECT 文によって取り出されるカラムは、SELECT キーワードに続けて必要なカラムをリストすることで制限できます。この SELECT 文の構文は、次のとおりです。

SELECT *column-name-1*, *column-name-2*,…
FROM *table-name*

column-name-1、*column-name-2*、*table-name* は、問い合わせるカラムとテーブルの名前に置き換えてください。

結果セットのカラムのリストは、「**select リスト**」と呼ばれます。このリストはカンマで区切られています。リストの最終カラムの後や、リストにカラムが 1 つしかない場合、カンマはありません。この方法でカラムを制限することを、「**射影**」と呼ぶことがあります。

❖ 各製品の名前、説明、色をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT name, description, color
 FROM product
```

name	description	color
Tee Shirt	Tank Top	White
Tee Shirt	V-neck	Orange
Tee Shirt	Crew Neck	Black
Baseball Cap	Cotton Cap	Black
…	…	…

カラムの再配列

各カラムは、SELECT 文で入力した順に表示されます。カラムを再配列する場合は、SELECT 文でカラム名の順序を変更してください。たとえば、**description** カラムを左側に配列するには、次の文を使用します。

```
SELECT description, name, color
 FROM product
```

計算カラムの使用

結果セットのカラムは、テーブルのカラムでなくてもかまいません。基本となるデータから計算される式も使用できます。テーブルのカラムを結合して 1 つの結果セット・カラムにしたり、さまざまな関数と演算子を使用して、取得する結果を制御したりできます。

❖ 各製品の在庫額をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT id, ( unit_price * quantity ) AS value
 FROM product
```

id	value
300	252
301	756
302	1050
400	1008
…	…

注意

- **カラムのエイリアスを指定できる**　デフォルトでは、カラム名は select リストにリストされる式ですが、計算カラムの場合、式では煩雑でわかりにくくなります。ここでは、計算カラムの名前を select リスト内で value に変更しています。値はカラムの「エイリアス」です。

- **その他の演算子を使用できる**　上の例では、乗算演算子を使用してカラムを結合しています。標準的な算術演算子、論理演算子、文字列演算子など、その他の演算子も使用できます。

  たとえば、次のクエリは、すべての顧客のフル・ネームをリストします。

  ```
  SELECT id, (fname || ' ' || lname ) AS "Full name"
   FROM customer
  ```

  || 演算子は文字列を連結しています。このクエリの場合、カラムのエイリアスにはスペースが付いているので、二重引用符で囲みます。この規則は、カラムのエイリアスだけでなく、データベース内のテーブル名やその他の識別子にも適用されます。

  演算子の完全なリストについては、『ASA SQL リファレンス・マニュアル』> 「演算子」を参照してください。

- **関数を使用できる**　カラムを結合するだけでなく、さまざまな組み込み関数を使用して、必要な結果を得ることができます。

  たとえば、次のクエリは製品名を大文字でリストします。

  ```
  SELECT id, UCASE( name )
   FROM product
  ```

id	UCASE(product.name)
300	TEE SHIRT
301	TEE SHIRT
302	TEE SHIRT
400	BASEBALL CAP
…	…

  関数の完全なリストについては、『ASA SQL リファレンス・マニュアル』> 「アルファベット順の関数リスト」を参照してください。

クエリ結果の順序付け

特に指定しないかぎり、データベース・サーバはテーブルのローを特に意味のない順序で返します。テーブルのローは、多くの場合、意味のある順序にした方が便利です。たとえば、製品をアルファベット順に見たいとします。

SELECT 文の末尾に ORDER BY 句を使用して、結果セットのローの順序を指定します。この SELECT 文の構文は、次のとおりです。

SELECT *column-name-1*, *column-name-2*,…
　FROM *table-name*
　ORDER BY *order-by-column-name*

column-name-1、*column-name-2*、*table-name* を、問い合わせるカラムとテーブルの名前に置き換えてください。*order-by-column-name* はテーブルのカラムです。この場合も、テーブルのすべてのカラムを表す省略形としてアスタリスクを使用できます。

❖ 製品をアルファベット順にリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力します。

```
SELECT id, name, description
 FROM product
 ORDER BY name
```

id	name	description
400	Baseball Cap	Cotton Cap
401	Baseball Cap	Wool cap
700	Shorts	Cotton Shorts
600	Sweatshirt	Hooded Sweatshirt
…	…	…

注意

- **句の順序が重要** ORDER BY 句は、FROM 句と SELECT 句の後に指定してください。
- **昇順または降順を指定できる** デフォルトの順序は昇順です。次のクエリのように句の末尾にキーワード DESC を追加すると、降順を指定できます。

```
SELECT id, quantity
 FROM product
 ORDER BY quantity DESC
```

id	quantity
400	112
700	80
302	75
301	54
600	39
…	…

- **複数のカラム順に順序を設定できる** 次のクエリは、最初にサイズ順 (アルファベット順)、次に名前順にソートします。

```
SELECT id, name, size
 FROM product
 ORDER BY size, name
```

id	name	size
600	Sweatshirt	Large
601	Sweatshirt	Large
700	Shorts	Medium
301	Tee Shirt	Medium
…	…	…

- **ORDER BY カラムが select リストになくてもよい**　次のクエリは、価格が結果セットになくても、製品を単価順にソートします。

```
SELECT id, name, size
 FROM product
 ORDER BY unit_price
```

id	name	size
500	Visor	One size fits all
501	Visor	One size fits all
300	Tee Shirt	Small
400	Baseball Cap	One size fits all
…	…	…

- **ORDER BY 句を使用せず、クエリを複数回実行すると、異なる結果が表示される可能性がある**　この理由は、Adaptive Server Anywhere が同じ結果セットを異なる順序で返す可能性があるためです。ORDER BY 句がない場合は、Adaptive Server Anywhere が最も効率のよい順序でローを返します。これは、結果セットの提示が、最後にアクセスしたローとその他の要因によって変化する可能性があることを意味します。特定の順序でローが返されるようにする唯一の方法は ORDER BY を使用することです。

インデックスの使用による ORDER BY のパフォーマンス改善

Adaptive Server Anywhere データベース・サーバで ORDER BY 句を使用してクエリを実行するときに、複数の方法が考えられる場合があります。インデックスを使用すると、データベース・サーバがより効率的にテーブルを検索できるようになります。

WHERE 句と ORDER BY 句を使ったクエリ

複数の実行方法があるクエリの一例は、WHERE 句と ORDER BY 句の両方を含むクエリです。

```
SELECT *
 FROM customer
 WHERE id > 300
 ORDER BY company_name
```

この例で、Adaptive Server Anywhere は次の 2 つの方法のどちらを採用するか決定する必要があります。

1. customer テーブル全体を、company_name の順序で検索し、各ローの customer_id の値が 300 以上かどうかをチェックする。

2. id カラムのキーを使用して、300 を超える id を持つ会社だけを読み込む。結果を会社名順にソートする必要があります。

id 値が 300 を超える会社がほとんどない場合は、2 番目の方法の方が優れています。スキャンするローの数が少なく、ソートにも時間がかからないからです。大半の会社の id 値が 300 を超える場合は、ソートの必要がない最初の方法の方がはるかに優れています。

問題の解決

id と company_name の 2 つのカラムでインデックスを作成すれば、上の例を解決できます。Adaptive Server Anywhere は、このインデックスを使用してテーブルからローを適切な順序で選択できます。ただし、インデックスはデータベース・ファイルの領域を消費し、更新にオーバヘッドがかかることは覚えておく必要があります。インデックスの作成は慎重に行ってください。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「インデックスの使用」を参照してください。

テーブルからのローの選択

クエリに WHERE 句を追加すると、SELECT 文によって取り出されるローを制限できます。これは、結果セットに対する「**制限**」の適用と呼ばれることもあります。WHERE 句には、返されるローを指定する「**探索条件**」が含まれます。この SELECT 文の構文は、次のとおりです。

SELECT *column-name-1*, *column-name-2*,…
　FROM *table-name*
　WHERE *search-condition*

この場合も、*column-name-1*、*column-name-2*、*table-name* は、問い合わせるカラムとテーブルの名前に置き換えてください。*search-condition* の詳細については後述します。ORDER BY 句を使用する場合は、WHERE 句の後に指定してください。

❖ カラーが黒の製品をすべてリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力します。

```
SELECT *
 FROM product
 WHERE color = 'black'
```

id	name	description	size	color	quantity	unit_price
302	Tee Shirt	Crew Neck	One size fits all	Black	75	14
400	Baseball Cap	Cotton Cap	One size fits all	Black	112	9
501	Visor	Plastic Visor	One size fits all	Black	28	7
…	…	…	…	…	…	…

注意

- WHERE 句には、ローを選択するための探索条件が含まれます。この場合の探索条件は `color = 'black'` です。その他の探索条件については、以降の項を参照してください。

  検索条件の詳細については、『ASA SQL リファレンス・マニュアル』> 「探索条件」を参照してください。

- **black** を囲む一重引用符は必須です。この一重引用符は、**black** が文字列であることを示します。二重引用符は異なる意味を持ちます。二重引用符を使用すると、通常はカラム名や他の識別子に使用できない文字列を有効にできます。

  文字列の詳細については、『ASA SQL リファレンス・マニュアル』> 「文字列」を参照してください。

- サンプル・データベースでは大文字と小文字を区別しないので、**'BLACK'**、**'black'**、または **'Black'** のどれを検索しても同じ結果になります。

- 句の順序は重要です。select リストに続いて FROM 句、WHERE 句、ORDER BY 句の順に指定します。句を異なる順序で入力すると構文エラーになります。

演習

この章で説明した内容を組み合わせて、クエリをいくつか試してみます。次のクエリは、John という名前のすべての従業員について氏名と生年月日をリストします。

```
SELECT ( emp_fname || ' ' || emp_lname) AS Name,
         birth_date
 FROM employee
 WHERE emp_fname = 'John'
 ORDER BY birth_date
```

Name	birth_date
John Letiecq	4/27/1939
John Sheffield	9/25/1955

探索条件による日付の比較

等号以外の演算子を使用して、探索条件と一致するローのセットを選択できます。不等号演算子 (< と >) を使用すると、数値、日付、文字列を比較できます。

❖ 生年月日が 1964 年 3 月 13 日より前の従業員をすべてリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力します。

```
SELECT emp_lname, birth_date
 FROM employee
 WHERE birth_date < 'March 13, 1964'
 ORDER BY birth_date DESC
```

emp_lname	birth_date
Ahmed	12/12/1963
Dill	7/19/1963
Rebeiro	4/12/1963
Garcia	1/23/1963
Pastor	7/14/1962
…	…

注意

- **日付への自動変換**　Adaptive Server Anywhere データベース・サーバは birth_date カラムに日付が入っていることを認識して、`'March 13, 1964'` の文字列を自動的に日付に変換します。
- **日付の指定方法**　日付を指定するには、さまざまな方法があります。次の例は、すべて Adaptive Server Anywhere で使用できます。

```
'March 13, 1964'
 '1964/03/13'
 '1964-03-13'
```

データベース・オプションを設定して、クエリに含まれる日付の解釈をチューニングできます。*yyyy/mm/dd* または *yyyy-mm-dd* フォーマットの日付は、DATE_ORDER 設定がされているかどうかに関わらず、日付として常に正確に認識されます。

クエリで許可される日付フォーマットを制御する方法については、『ASA データベース管理ガイド』> 「DATE_ORDER オプション [互換性]」と『ASA データベース管理ガイド』> 「オプションの設定」を参照してください。

- **その他の比較演算子**　使用できる比較演算子の完全なリストについては、『ASA SQL リファレンス・マニュアル』> 「比較演算子」を参照してください。

探索条件でのパターン一致

パターン一致は、文字データを識別する便利な方法です。SQL では、パターンの検索には LIKE キーワードを使用します。パターン一致では、ワイルドカード文字を使用して、文字のさまざまな組み合わせに対応します。

❖ 姓が BR で始まるすべての従業員をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力します。

```
SELECT emp_lname, emp_fname
 FROM employee
 WHERE emp_lname LIKE 'br%'
```

emp_lname	emp_fname
Breault	Robert
Braun	Jane

探索条件にある % は、文字 BR の後ろに任意の数の文字が続くことを示します。

❖ **姓が BR で始まり、次に 0 文字以上の文字と T が続き、そのあとに 0 文字以上の文字が続くすべての従業員をリストするには、次の手順に従います。**

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力します。

```
SELECT emp_lname, emp_fname
 FROM employee
 WHERE emp_lname LIKE 'BR%T%'
```

emp_lname	emp_fname
Breault	Robert

最初の **%** 記号は文字列 **eaul** と対応しますが、2 番めの % 記号は空の文字列 (文字がない) と対応します。

LIKE とともに使用できるもう 1 つの特殊文字は **_** (アンダースコア) 文字です。この特殊文字は、厳密に 1 つの文字と対応します。たとえば、パターン `'BR_U%'` は、`BR` で始まり 4 番めの文字が `U` であるすべての名前と対応します。`Braun` では、_ は文字 `A` に対応し、`%` は `N` に対応します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「LIKE 条件」を参照してください。

音によるローの一致

SOUNDEX 関数を使用すると、ローを発音で対応させることができます。たとえば、"Ms. Brown" のように聞こえる名前を持つ従業員に対して電話メッセージが残されているとします。社内のどの従業員が Brown のような発音の名前を持っているでしょうか。

❖ **Brown のように聞こえる姓を持つ従業員をリストするには、次の手順に従います。**

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力します。

```
SELECT emp_lname, emp_fname
 FROM employee
 WHERE SOUNDEX( emp_lname ) = SOUNDEX( 'Brown' )
```

emp_lname	emp_fname
Braun	Jane

SOUNDEX によって使用されるアルゴリズムは、主に英語版データベースを対象としています。

詳細については、『ASA SQL リファレンス・マニュアル』> 「SOUNDEX 関数 [文字列]」を参照してください。

複合探索条件の使用

探索条件を論理演算子 AND と OR で結合して、複合探索条件にすることができます。探索条件の各部分は「述部」とも呼ばれます。

❖ **Whitney という名前の従業員を除いて、1964 年 3 月 13 日よりも前に生まれたすべての従業員をリストするには、次の手順に従います。**

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力します。

```
SELECT emp_lname, birth_date
 FROM employee
 WHERE birth_date < '1964-3-13'
 AND emp_lname <> 'whitney'
```

emp_lname	birth_date
Cobb	12/4/1960
Jordan	12/13/1951
Breault	5/13/1947
Espinoza	12/14/1939

emp_lname	birth_date
Dill	7/19/1963
Francis	9/12/1954

複合探索条件のショートカット

省略形 BETWEEN の使用

SQL では、複合探索条件に 2 つの省略形があります。BETWEEN は、ある範囲の値を検索する場合に使用されます。たとえば、次の 2 つのクエリは同じです。

```
SELECT emp_lname, birth_date
 FROM employee
 WHERE birth_date BETWEEN '1963-1-1' AND '1965-3-31'
```

および

```
SELECT emp_lname, birth_date
 FROM employee
 WHERE birth_date >= '1963-1-1'
 AND birth_date <= '1965-3-31'
```

省略形 IN の使用

もう 1 つの省略形 IN は、いくつかの値のうちの 1 つを検索する場合に使用されます。次の 2 つの文は同じです。

```
SELECT emp_lname, emp_id
 FROM employee
 WHERE emp_lname IN ('yeung','bucceri','charlton')
```

および

```
SELECT emp_lname, emp_id
 FROM employee
 WHERE emp_lname = 'yeung'
 OR emp_lname = 'bucceri'
 OR emp_lname = 'charlton'
```

第 8 章

複数のテーブルからのデータの選択

この章の内容

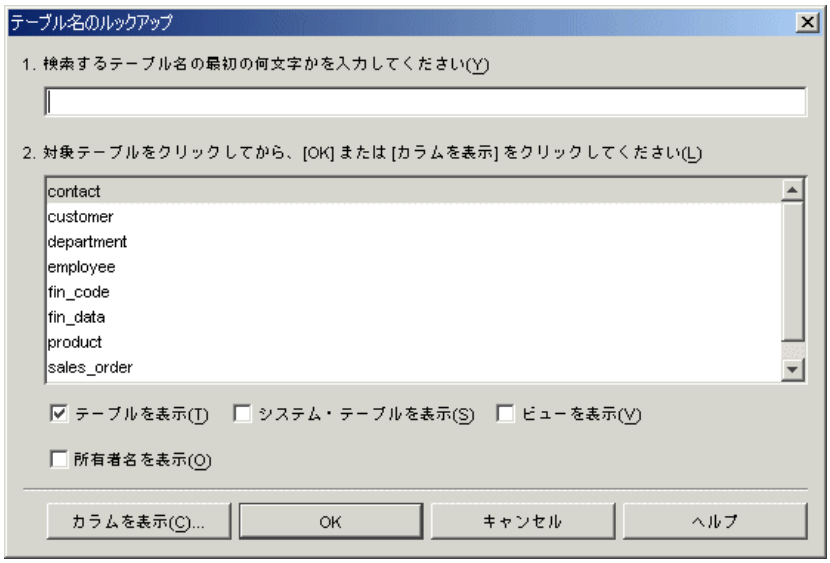
この章では、複数のテーブルにある情報を検索するデータベース・クエリについて説明します。このタスクを行うために、SQL では「**JOIN**」演算子が用意されています。クエリ内で複数のテーブルをジョインするには、何通りかの方法があります。この章では、そのうちのいくつか重要な方法について説明します。

概要

複数の関連性を持つテーブルからデータを表示したい場合があります。この章では、「ジョイン」を使用して役に立つ、意味を持ったデータを表示する方法について説明します。

テーブルのリストを表示する

Interactive SQL では、[F7] キーを押すとテーブルのリストを表示できます。



テーブルを選択してから [カラムを表示] をクリックすると、そのテーブルのカラムが表示されます。[ESC] キーを押すとテーブル・リストに戻り、もう一度 [Esc] キーを押すと [SQL 文] ウィンドウ枠に戻ります。[ESC] キーの代わりに [ENTER] キーを押すと、選択されているテーブルまたはカラム名が [SQL 文] ウィンドウ枠の現在カーソルが置かれている場所にコピーされます。

リストを終了するには、[ESC] キーを押します。

サンプル・データベース内のテーブルを示す図については、「サンプル・データベースについて」252 ページを参照してください。

直積を使用したテーブルのジョイン

サンプル・データベースにはテーブルの 1 つとして sales_order があります。このテーブルには、この会社が受けた注文がリストされています。注文ごとに、その注文を担当している営業担当者の従業員 ID を記録した sales_rep カラムがあります。sales_order テーブルには 648 のロー、employee テーブルには 75 のローがあります。

直積ジョインは、ジョインについて理解するための入口としては適していますが、直積ジョイン自体はそれほど便利なものではありません。

❖ **employee テーブルと sales_order テーブルのすべてのデータをリストするには、次の手順に従います。**

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT *
 FROM sales_order CROSS JOIN employee
```

  このクエリの結果が Interactive SQL の [結果] ウィンドウ枠の [結果] タブに表示されます。この結果では、employee テーブルの各ローは、sales_order テーブルの各ローと対応しています。employee テーブルには 75 のローがあり、sales_order テーブルには 648 のローがあるので、ジョインの結果には 75 × 648 = 48,600 のローが存在します。各ローは、sales_order テーブルからのすべてのカラムの後に、employee テーブルからのすべてのカラムが続く形で構成されています。このジョインは「**直積**」と呼ばれます。

この後の各項では、選択の幅が広いジョインを構成する方法について説明します。選択の幅が広いジョインは、直積テーブルに対する制限の適用とみなすことができます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「クロス・ジョイン」を参照してください。

ON 句を使用したジョインの制限

ON 句は、ジョインのローに制限を適用します。これは、WHERE 句がクエリのローに制限を適用するのと同様です。

ON 句を使用すると、CROSS JOIN よりも使用しやすいジョインを構成できます。たとえば、sales_order テーブルと employee テーブルのジョインに ON 句を適用できます。この場合、取得される結果のすべてのローで、sales_order テーブル内の sales_rep が employee テーブル内のものと同じになるように制限されます。各ローには、注文とその注文を担当する営業担当者についての情報が入っています。

❖ すべての注文とその日付、担当者をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT employee.emp_lname,
        sales_order.id,
        sales_order.order_date
 FROM sales_order JOIN employee
    ON sales_order.sales_rep = employee.emp_id
```

emp_lname	id	order_date
Chin	2008	4/2/2001
Chin	2020	3/4/2001
Chin	2032	7/5/2001
Chin	2044	7/15/2000
Chin	2056	4/15/2001
…	…	…

注意

カラムを識別するプレフィクスとしてテーブル名を使用します。プレフィクスとしてテーブル名を使用すると、文がわかりやすくなります。また、2 つのテーブルに同じ名前のカラムがある場合には必ず使用します。

このクエリの結果には、648 のローだけが (sales_order テーブル内のローごとに 1 つずつ) 入っています。直積における 48,600 のローのうち、2 つのテーブルで同じ従業員番号を持っているのは 648 のローだけだからです。

結果の順序に意味はありません。ORDER BY 句を追加すると、クエリに特定の順序を指定できます。

ON 句によって、最終的な結果セットには含まれないカラムが組み込まれます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「明示的なジョイン条件 (ON 句)」を参照してください。

キー・ジョインを使用したテーブルのジョイン

一般的なジョインの多くは 2 つのテーブル間で外部キーによって関連付けられます。最も一般的なジョインは、外部キー値がプライマリ・キー値と同じになるように制限します。

KEY JOIN 演算子は、外部キーの関係に基づいて 2 つのテーブルをジョインします。つまり、Adaptive Server Anywhere は、一方のテーブルのプライマリ・キー・カラムを他方のテーブルの外部キー・カラムと同等とする ON 句を生成します。

前項の例では、sales_order テーブル内の外部キー値を、employee テーブル内のプライマリ・キー値と同じになるように制限します。

```
SELECT employee.emp_lname,
        sales_order.id,
        sales_order.order_date
 FROM sales_order JOIN employee
    ON sales_order.sales_rep = employee.emp_id
```

KEY JOIN を使用すると、クエリをもっと簡単に表すことができます。

```
SELECT employee.emp_lname,
        sales_order.id,
        sales_order.order_date
 FROM sales_order KEY JOIN employee
```

KEY JOIN は ON 句を入力するための単なるショートカットで、この 2 つのクエリは同じです。JOIN を指定しても CROSS、NATURAL、KEY を指定しない場合、または ON 句を使用する場合のデフォルトは、キー・ジョインです。

従業員データベースの図では、テーブル間を結ぶラインは外部キーを表します。KEY JOIN 演算子は、図の中で 1 本の線によって 2 つのテーブルがジョインされているところならどこでも使用できます。

サンプル・データベースの図については、「サンプル・データベースについて」252 ページを参照してください。

3 つ以上のテーブルのジョイン

JOIN 演算子を使用すると、2 つ以上のテーブルでもジョインできます。次のクエリは、4 つのテーブルを使用して各顧客の注文の合計値をリストします。このクエリは、4 つのテーブル customer、sales_order、sales_order_items、product の各組の関係に単一の外部キーを使用して互いに接続します。

❖ **すべての会社とその注文の総額をリストするには、次の手順に従います。**

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT company_name,
        SUM( sales_order_items.quantity *
             product.unit_price) AS value
 FROM ( ( customer KEY JOIN sales_order )
    KEY JOIN sales_order_items )
    KEY JOIN product
 GROUP BY company_name
```

company_name	値
Bensoul's Boutique	1332
Bush Pro Shop	2940
Sterling & Co.	6804
Ocean Sports	3744
…	…

注意

結果セットが異なる順序で表示される場合があります。結果セット内のローの順序に意味はありません。

この例では、集合関数である SUM 演算子を使用しています。集合関数は GROUP BY 句とともに使用し各ロー・グループの値を返します。この例では、`sales_order_items.quantity * product.unit_price` の合計、つまり製品の種類ごとに支払われた合計金額が `company_name` ごとに計算され、その結果、会社ごとの売り上げが返されます。

FROM 句のカッコは、ジョインの順序を明確にするために使用されています。

複雑なキー・ジョインの詳細については、『ASA SQL ユーザーズ・ガイド』> 「キー・ジョイン」を参照してください。

ナチュラル・ジョインを使用したテーブルのジョイン

NATURAL JOIN 演算子は、共通のカラム名に基づいて 2 つのテーブルをジョインします。つまり、Adaptive Server Anywhere は各テーブルからの共通のカラムを同等とする ON 句を生成します。

❖ すべての従業員とその所属部門をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT emp_lname, dept_name
 FROM employee NATURAL JOIN department
```

emp_lname	dept_name
Whitney	R & D
Cobb	R & D
Breault	R & D
Shishov	R & D
Driscoll	R & D
…	…

Adaptive Server Anywhere は 2 つのテーブルを参照し、共通するカラム名が dept_id だけであると判断します。次の ON CLAUSE が内部的に生成され、ジョインの実行に使用されます。

```
FROM employee JOIN department
     ON employee.dept_id = department.dept_id
```

NATURAL JOIN は ON 句を入力するための単なるショートカットで、この 2 つのクエリは同じです。

NATURAL JOIN を使用した場合のエラー

この JOIN 演算子は、等価ではないカラムを等価と見なすと問題が発生する可能性があります。たとえば、次のクエリを実行すると、意図しない結果が生成されます。

```
SELECT *
 FROM sales_order NATURAL JOIN customer
```

このクエリを実行してもローは返されません。内部的に次の ON 句が生成されます。

```
FROM sales_order JOIN customer
     ON sales_order.id = customer.id
```

sales_order テーブル内の id カラムは*注文*の ID 番号です。customer テーブル内の id カラムは*顧客*の ID 番号です。これらはどれも一致しませんでした。もちろん、一致があったとしても意味がありません。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「ナチュラル・ジョイン」を参照してください。

外部ジョインを使用したテーブルのジョイン

前述の例では、ジョイン条件が満たされる場合にのみローを返すジョインを作成しました。この種のジョインは「**内部ジョイン**」と呼ばれ、デフォルトです。ただし、あるテーブルのすべてのローを保護したい場合があります。そのような場合は「**外部ジョイン**」を使用します。

右側のテーブルにあるローをすべて保護する「**右外部ジョイン**」、左側のテーブルを保護する「**左外部ジョイン**」、または両方のテーブルのローをすべて保護する「**全外部ジョイン**」を指定できます。

❖ 顧客とその注文の日付をすべてリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT lname, order_date, city
 FROM customer KEY LEFT OUTER JOIN sales_order
 WHERE customer.state = 'NY'
 ORDER BY order_date
```

lname	order_date	city
Thompson	(NULL)	Bancroft
Reiser	1993-01-22	Rockwood
Clarke	1993-01-27	Rockwood
Mentary	1993-01-30	Rockland
…	…	…

この文では、注文の有無にかかわらず、すべての顧客が入れられます。顧客が注文していない場合には、注文情報に対応する結果の各カラムには NULL が入ります。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「外部ジョイン」を参照してください。

第 9 章

サブクエリを使用したデータの選択

この章の内容

この章では、あるクエリの結果を別の SELECT 文の一部として使用する方法について説明します。これは、より複雑で情報量の多いクエリを構築するのに役立つツールです。

サブクエリの概要

リレーショナル・データベースを使用すると、複数のテーブルに関連するデータを保管できます。別のクエリの WHERE 句または HAVING 句に表示されるクエリである「**サブクエリ**」を使用して、関連するテーブルからデータを抽出できます。サブクエリを使用すると、一部のクエリをジョインより簡単に記述できます。また、サブクエリを使用しないと記述できないクエリがあります。

サブクエリは、あるクエリの結果を別のクエリの一部として使用します。この項では、サブクエリを使用できる場合について説明します。例として、在庫が少なくなっている製品の注文項目をリストするクエリを構築します。

このリストを生成するには、2 つのクエリが必要です。この項では、最初に個々のクエリについて説明してから、同じ結果を生成する 1 つのクエリについて説明します。

❖ 在庫数が 20 未満のすべての製品をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT id, description, quantity
 FROM product
 WHERE quantity < 20
```

id	description	quantity
401	Wool cap	12

このクエリは、在庫数が少ない製品はウールの帽子だけであることを示します。

❖ ウールの帽子についてすべての注文項目をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT *
 FROM sales_order_items
 WHERE prod_id = 401
 ORDER BY ship_date DESC
```

id	line_id	prod_id	quantity	ship_date
2082	1	401	48	7/9/2001
2053	1	401	60	6/30/2001
2125	2	401	36	6/28/2001
2027	1	401	12	6/17/2001
…	…	…	…	…

在庫数が少ない項目を調べて、これらの項目に対する受注を調べるというこの 2 段階の処理は、サブクエリを使用して単一のクエリに結合できます。

❖ 在庫数が少ない項目のすべての注文項目をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT *
 FROM sales_order_items
 WHERE prod_id IN
     (    SELECT id
          FROM product
          WHERE quantity < 20 )
 ORDER BY ship_date DESC
```

id	line_id	prod_id	quantity	ship_date
2082	1	401	48	7/9/2001
2053	1	401	60	6/30/2001
2125	2	401	36	6/28/2001
2027	1	401	12	6/17/2001
…	…	…	…	…

SQL 文中のサブクエリは、カッコで囲まれた次のフレーズです。

```
(     SELECT id
       FROM product
       WHERE quantity < 20 )
```

このサブクエリは、product テーブル内の id カラムにおいて WHERE 句の探索条件を満たすすべての値のリストを作成します。

一連のローが返されますが、返されるカラムは 1 つだけです。IN キーワードは、それぞれの値をセットのメンバとして扱い、メイン・クエリ内の各ローがセットのメンバかどうかをテストします。

単一ローのサブクエリと複数ローのサブクエリ

サブクエリが返すことのできるローとカラムの数には制約があります。IN、ANY、または ALL を使用すると、サブクエリは複数のローを返すことができますが、返されるカラムは 1 つだけです。その他の演算子を使用すると、サブクエリは 1 つの値を返します。

複数ローのサブクエリ

サンプルデータベース内にある 2 つのテーブルは、経理に関するデータです。fin_code テーブルは、経理データとこれらの意味についてのさまざまなコードが入っている小さなテーブルです。

fin_data テーブルから歳入項目をリストするには、次のように入力します。

```
SELECT *
 FROM fin_data
 WHERE fin_data.code IN
      (    SELECT fin_code.code
           FROM fin_code
           WHERE type = 'revenue' )
```

year	quarter	code	amount
1999	Q1	r1	1023
1999	Q2	r1	2033
1999	Q3	r1	2998
1999	Q4	r1	3014
2000	Q1	r1	3114

この例では、各参照の code カラムが属するテーブルを明確に識別する修飾子を使用しました。この例に関しては、修飾子を省略することもできます。

他の 2 つのキーワード、ANY と ALL を演算子の修飾子として使うと、その演算子は複数のローに対して機能します。

次のクエリは、正常終了する前述のクエリとまったく同じです。

```
SELECT *
 FROM fin_data
 WHERE fin_data.code = ANY
     (    SELECT fin_code.code
          FROM fin_code
          WHERE type = 'revenue' )
```

=ANY 条件は IN 条件とまったく同じですが、ANY を < や > などの不等号とともに使用するとサブクエリを柔軟に使用できます。

ALL キーワードは ANY に似ています。たとえば、次のクエリは歳入以外の経理データをリストします。

```
SELECT *
 FROM fin_data
 WHERE fin_data.code <> ALL
     (    SELECT fin_code.code
          FROM fin_code
          WHERE type = 'revenue' )
```

このクエリは、NOT IN を使用した場合の次のコマンドと同じです。

```
SELECT *
 FROM fin_data
 WHERE fin_data.code NOT IN
     (    SELECT fin_code.code
          FROM fin_code
          WHERE type = 'revenue' )
```

サブクエリ使用時の一般的なエラー

通常、サブクエリの結果セットは単一カラムに制限されています。次の例は、fin_code テーブルのどのカラムを fin_data.code カラムと比較するかを Adaptive Server Anywhere が認識しないので意味がありません。

```
-- this query is incorrect
 SELECT *
 FROM fin_data
 WHERE fin_data.code IN
     (    SELECT fin_code.code, fin_code.type
          FROM fin_code
          WHERE type = 'revenue' )
```

単一ローのサブクエリ

IN 条件とともに使用されるサブクエリがローのセットを返してもよいのに対して、比較演算子とともに使用されるサブクエリはローを 1 つだけ返す必要があります。たとえば、次のコマンドを実行すると、サブクエリは 2 つのローを返すのでエラーが発生します。

```
-- this query is incorrect
 SELECT *
 FROM fin_data
 WHERE fin_data.code =
     (    SELECT fin_code.code
          FROM fin_code
          WHERE type = 'revenue' )
```

ジョインに代わるサブクエリの使用

たとえば、受注と受注先の日付順リストが必要な場合に、顧客 ID ではなく会社名を知りたいとします。次のようなジョインを使用して、この結果を得ることができます。

ジョインの使用

2001 年 1 月 1 日以降の受注 ID、日付、各注文を行った会社名をリストするには、次のように入力します。

```
SELECT     sales_order.id,
               sales_order.order_date,
               customer.company_name
 FROM sales_order
     KEY JOIN customer
 WHERE order_date > '2001/01/01'
 ORDER BY order_date
```

id	order_date	company_name
2473	1/4/2001	Peachtree Active Wear
2474	1/4/2001	Sampson & Sons
2106	1/5/2001	Salt & Pepper's
2475	1/5/2001	Cinnamon Rainbow's
2036	1/5/2001	Hermanns

サブクエリの使用方法

次の SQL 文は、ジョインではなくサブクエリを使用して同じ結果を得ます。

```
SELECT sales_order.id,
     sales_order.order_date,
     (    SELECT company_name FROM customer
          WHERE customer.id = sales_order.cust_id )
 FROM sales_order
 WHERE order_date > '2001/01/01'
 ORDER BY order_date
```

sales_order テーブルがサブクエリの一部でない場合でも、サブクエリは sales_order テーブル内の cust_id カラムを参照します。一方、sales_order.cust_id カラムは SQL 文の本文にある sales_order テーブルを参照します。これは「**外部参照**」と呼ばれます。外部参照があるサブクエリは「**相関サブクエリ**」と呼ばれます。

他のテーブルから要求されるカラムが 1 つだけである場合は、ジョインの代わりにサブクエリを使用できます。サブクエリが返すことができるカラムは 1 つだけです。この例では company_name カラムだけを必要としていたので、ジョインをサブクエリに変更することができました。

サブクエリが結果を得ない可能性がある場合、この方法は外部ジョインと呼ばれます。このチュートリアルの前の項で示したジョインは、より正確に言うと内部ジョインと呼ばれます。

外部ジョインの使用

ワシントン州在住の顧客名すべてとその顧客の最も最近の受注 ID をリストするには、次のように入力します。

```
SELECT     company_name, state,
     (    SELECT MAX( id )
          FROM sales_order
          WHERE sales_order.cust_id = customer.id )
 FROM customer
 WHERE state = 'WA'
```

company_name	state	MAX(sales_order.id)
Custom Designs	WA	2547
It's a Hit!	WA	(NULL)

It's a Hit! という会社は何も注文しなかったので、サブクエリはこの顧客については NULL を返します。内部ジョインを使用した場合、発注しなかった会社はリストされません。

外部ジョインを明示的に指定することもできます。その場合は、次のように GROUP BY 句も指定する必要があります。

```
SELECT company_name, state,
     MAX( sales_order.id )
 FROM customer
     KEY LEFT OUTER JOIN sales_order
 WHERE state = 'WA'
 GROUP BY company_name, state
```

第 10 章

集合データの選択

この章の内容

この章では、集合情報をユーザに知らせるクエリを構築する方法について説明します。次に集合情報の例を示します。

- カラム内にあるすべての値の合計
- カラム内にある個別のエントリの数
- カラム内にあるエントリの平均値

データの要約

一部のクエリでは、テーブル内の、個々のローではなくロー・グループのプロパティを反映するデータの内容を検査します。たとえば、顧客が注文した総額の平均値や、各部門で何人の従業員が仕事をしているかなどです。この種のタスクには、「集合」関数と GROUP BY 句を使用します。

集合関数の概要

集合関数は、一連のローについて単一の値を返します。GROUP BY 句がないと、集合関数はクエリの他の条件を満たすすべてのローについて、単一の値を返します。

❖ 社内の従業員数をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT COUNT( * )
 FROM employee
```

COUNT(*)
75

この結果セットは、タイトル **COUNT(*)** を持つ 1 つのカラムと、従業員の合計数が入った 1 つのローで構成されています。

❖ 社内の従業員数と、最年長の従業員および最年少の従業員の生年月日をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT COUNT(*), MIN(birth_date), MAX(birth_date)
 FROM employee
```

COUNT(*)	MIN(employee.birth_date)	MAX(employee.birth_date)
75	1/2/1936	1/18/1973

関数 COUNT、MIN、MAX は「**集合関数**」と呼ばれます。この 3 つの関数は、それぞれ情報を要約します。その他の集合関数として、AVG、STDDEV、VARIANCE などの統計関数があります。COUNT 以外のすべての集合関数では、パラメータとしてカラムの名前を使用します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「集合関数」を参照してください。

グループ分けされたデータに対する集合関数の適用

集合関数はテーブル全体についての情報を提供するだけでなく、ローのグループについても使用できます。GROUP BY 句は、ローをグループ単位で配置し、集合関数はロー・グループごとに 1 つの値を返します。

例

❖ **営業担当者と各担当者が受注した件数をリストするには、次の手順に従います。**

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT sales_rep, count( * )
 FROM sales_order
 GROUP BY sales_rep
 ORDER BY sales_rep
```

sales_rep	count(*)
129	57
195	50
299	114
467	56
…	…

GROUP BY 句は Adaptive Server Anywhere に対して、他の条件の場合に返されるすべてのローのセットを分割するように指示します。各分割、つまりグループに含まれるすべてのローは、指定のカラムに同じ値を持ちます。ユニークな値ごと、または値セットごとに、1 つだけグループがあります。この場合、各グループに含まれるすべてのローの sales_rep 値が同じになります。

COUNT などの集合関数は、各グループのローに適用されます。したがって、この結果セットには各グループのローの合計数が表示されます。クエリの結果は、各営業担当者の ID 番号が入った 1 つのローで構成されています。各ローには、営業担当者 ID と、その担当者の受注の合計数が入ります。

GROUP BY を使用した場合には、結果テーブルには GROUP BY 句に指定したカラムまたはカラム・セットごとにローが 1 つずつあります。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「GROUP BY 句：クエリ結果のグループへの編成」を参照してください。

GROUP BY 句を使用する場合の一般的なエラー

GROUP BY 句を使用する場合の一般的なエラーは、1 つのグループにまとめることができない情報を得ようとすることです。次に例を示します。

```
-- This query is incorrect
 SELECT sales_rep, emp_lname, COUNT( * )
 FROM sales_order KEY JOIN employee
 GROUP BY sales_rep
```

これを実行すると、次のエラーが表示されます。

```
select リストの中の 'emp_lname' に対する関数またはカラムの参照は
GROUP BY 句の中になければなりません。
```

エラーがレポートされるのは、特定の ID を持つ従業員に関する各結果ローにすべて同じ姓が入っているかどうかを、Adaptive Server Anywhere が確認できないためです。

このエラーを解決するには、GROUP BY 句にカラムを追加します。

```
SELECT sales_rep, emp_lname, COUNT( * )
 FROM sales_order KEY JOIN employee
 GROUP BY sales_rep, emp_lname
 ORDER BY sales_rep
```

この方法が適切でない場合は、代わりに集合関数を使用して、次のように値を 1 つだけ選択できます。

```
SELECT sales_rep, MAX( emp_lname ), COUNT( * )
 FROM sales_order KEY JOIN employee
 GROUP BY sales_rep
 ORDER BY sales_rep
```

MAX 関数は、各グループのディテール・ローから最大 (アルファベット順の最後) の姓を選択します。最大値は 1 つしかないので、この文は有効です。この場合は、グループ内のすべてのディテール・ローに同じ姓が表示されます。

グループの制限

WHERE 句を使用して結果セット内のローを制限する方法については、すでに説明しました。グループ内のローを制限するには、HAVING 句を使用します。

❖ 受注数が 55 を超えるすべての営業担当者をリストするには、次の手順に従います。

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
SELECT sales_rep, count( * ) AS orders
 FROM sales_order KEY JOIN employee
 GROUP BY sales_rep
 HAVING count( * ) > 55
 ORDER BY orders DESC
```

sales_rep	orders
299	114
129	57
1142	57
467	56

句の順序
GROUP BY 句は、常に HAVING 句より前に置いてください。両方の句を使用する場合は、WHERE 句を GROUP BY 句より前に置きます。

HAVING 句と WHERE 句の両方を 1 つのクエリに使用できます。条件が HAVING 句に置かれている場合は、グループが構成されたあとでのみ結果のローを論理的に制限します。条件が WHERE 句に置かれている場合は、グループが構成される前に論理が評価されるので、時間が節約されます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「HAVING 句 : データ・グループの選択」を参照してください。

WHERE 句と HAVING 句の組み合わせ

WHERE 句または HAVING 句を使用して、同じロー・セットを指定できる場合があります。このような場合に、効率的な方法とそうでない方法があります。オプティマイザは、入力された各文を常に自動的に分析し、効率的な実行方法を選択します。必要な結果を最も明確に記述する構文を使用するのが最善です。通常は、前にある句の不要なローが削除されます。

例

受注数が 55 を超え、かつ ID が 1000 よりも大きいすべての営業担当者をリストするには、次の文を入力します。

```
SELECT sales_rep, count( * )
 FROM sales_order KEY JOIN employee
 WHERE sales_rep > 1000
 GROUP BY sales_rep
 HAVING count( * ) > 55
 ORDER BY sales_rep
```

次の SQL 文も同じ結果になります。

```
SELECT sales_rep, count( * )
 FROM sales_order KEY JOIN employee
 GROUP BY sales_rep
 HAVING count( * ) > 55 AND sales_rep > 1000
 ORDER BY sales_rep
```

Adaptive Server Anywhere は、両方の文で同じ結果セットが記述されていることを検出するため、それぞれの文が効率的に実行されます。

第 11 章

データベースの更新

この章の内容

この章では、データベースの内容を変更する方法について説明します。これには、ローの追加、ローの削除、ローの内容の変更だけでなく、変更を永続的なものにしたり、すでに行った変更を修正したりする方法も含まれます。

概要

この章では、データベース内のデータを変更する方法について説明します。次の 3 つの基本操作があります。

- **挿入**　テーブルにローを追加して、新しいデータを組み込むことができます。
- **削除**　テーブルのローを削除して、データを削除できます。
- **更新**　テーブルの既存のローを修正できます。

各操作を実行するには、SQL 文を使用します。

テーブルへのローの追加

たとえば、企業が新しい製品、茶色のアクリル製ベストの販売を決定したと仮定します。これにより、サンプル・データベースの product テーブルにデータを追加する必要が生じます。

❖ **product テーブルに茶色のアクリル製ベストを追加するには、次の手順に従います。**

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
INSERT
 INTO product ( id, name, description, size, color,
                quantity, unit_price )
 VALUES ( 800, 'Vest', 'Acrylic Vest', 'Extra
Large',
          'Brown', 42, 20.00 )
```

  誤っていずれかのカラムを指定するのを忘れると、Adaptive Server Anywhere はエラーをレポートします。

Interactive SQL で、結果セットからデータベース・テーブルに新しいローを追加することもできます。

詳細については、「Interactive SQL でのテーブル値の編集」282 ページを参照してください。

カラム内の NULL 値

NULL 値は、未定であるかまたは該当する値がないかのどちらかを示すのに使用される特別な値です。NULL 値が許可されるカラムもあれば許可されないカラムもあります。

テーブル内のローの変更

多くのデータベースでは、すでに保管されているレコードを更新する必要があります。たとえば、従業員が部署間で異動になると、部署 ID だけでなくマネージャ ID も変わります。

❖ **Interactive SQL で従業員 #195 を部門 400 に移動するには、次の手順に従います。**

- Interactive SQL で、[SQL 文] ウィンドウ枠に次のように入力し、[F5] キーを押して文を実行します。

```
UPDATE employee
 SET dept_id = 400, manager_id = 1576
 WHERE emp_id = 195
```

この SQL 文は、従業員 Marc Dill (従業員 ID 195) について 2 種類の更新を同時に実行します。

Adaptive Server Anywhere では、WHERE 句の条件を満たすすべてのローを更新するため、一度に 2 つ以上のローが更新されることがあります。たとえば、営業部員のグループがマーケティング部に異動し、それに伴って彼らの dept_id カラムが更新される場合、次の文によってマーケティング部に所属するすべての従業員の manager_id が 1576 に設定されます。

```
UPDATE employee
 SET manager_id = 1576
 WHERE dept_id = 400
```

すでにマーケティング部に所属していた従業員については、変更はありません。

また、Interactive SQL で結果セットからのローを修正することもできます。

詳細については、「Interactive SQL でのテーブル値の編集」282 ページを参照してください。

ローの削除

テーブルからローを削除する必要がある場合があります。たとえば、Rodrigo Guevara (従業員 ID 249) が会社を退社するとします。次の SQL 文は、employee テーブルから Rodrigo Guevara を削除します。

```
DELETE
 FROM employee
 WHERE emp_id = 249
```

UPDATE 文と DELETE 文を使用して、複雑な検索条件を実行することもできます。たとえば employee テーブルを再編成する場合に次の SQL 文を使用すると、1988 年 3 月 3 日から 1989 年 3 月 3 日の間に入社したすべての男性従業員を employee テーブルから削除できます。

```
DELETE
 FROM employee
 WHERE sex = 'm'
     AND start_date BETWEEN '1988-03-03'
     AND '1989-03-03'
```

また、Interactive SQL の結果セットから、データベース・テーブルのローを削除することもできます。

詳細については、「Interactive SQL でのテーブル値の編集」282 ページを参照してください。

ここでは保持する必要がない変更をデータベースに対して行ったので、次のコマンドを使用して変更を取り消してください。

```
ROLLBACK
```

トランザクションによる変更のグループ分け

Adaptive Server Anywhere では、コマンドはトランザクションにまとめます。トランザクションをコミットすると、データベースの変更が永続的なものになります。データを変更しても、その変更がすぐに永続的なものになるわけではありません。変更は「**トランザクション・ログ**」に格納され、COMMIT コマンドを入力すると永続的なものになります。

トランザクションを開始または終了するコマンドや動作を理解すると、トランザクションを有効に使用できます。

トランザクションの開始

トランザクションは次のいずれかのイベントで開始します。

- データベースへの接続後、最初の文
- トランザクションの終了後、最初の文

トランザクションの完了

トランザクションは次のいずれかのイベントで完了します。

- データベースの変更を確定する COMMIT 文
- トランザクションで行われたすべての変更を取り消す ROLLBACK 文
- オートコミットが実行される文。データベース定義コマンドの ALTER、CREATE、COMMENT、DROP はすべて自動的にコミットを実行します。
- データベースへの接続を解除すると、暗黙的なロールバックが実行されます。

Interactive SQL のオプション

Interactive SQL は、いつ、どのようにトランザクションを終了するかを指定する次の 2 つのオプションを提供します。

- AUTO_COMMIT オプションを ON に設定すると、Interactive SQL では、文が正常に実行された場合は結果が自動的にコミットされ、失敗した場合は ROLLBACK が自動的に実行される。

- COMMIT_ON_EXIT オプションは、Interactive SQL を終了したときにコミットされていない変更をどうするかを決定する。このオプションを ON (デフォルト) に設定すると、Interactive SQL は COMMIT を実行し、それ以外の場合はコミットされていない変更を ROLLBACK 文で取り消します。

データ・ソースを使用している場合
デフォルトでは、ODBC はオートコミット・モードで動作します。Interactive SQL で AUTO_COMMIT オプションを OFF に設定しても、Interactive SQL の設定は ODBC の設定によって上書きされます。ODBC の設定は、SQL_ATTR_AUTOCOMMIT 接続属性を使用して変更できます。ODBC オートコミットは CHAINED オプションから独立しています。

Adaptive Server Anywhere は、BEGIN TRANSACTION のような Transact-SQL コマンドもサポートし、Sybase Adaptive Server Enterprise との互換性を保ちます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「Transact-SQL のサポートの概要」を参照してください。

変更の確定

COMMIT 文は、すべての変更を永続的なものにします。

COMMIT 文は、まとまった意味を持つ文のグループの後で使用してください。たとえば、ある顧客の口座から別の顧客の口座に金銭を振り込む場合、振り込み前と後の合計金額が同一である必要があるため、振り込まれる側の口座にその金額を加え、その後で振り込む側の口座からその金額を削除して、最後にコミットします。

AUTO_COMMIT オプションを ON に設定すると、Adaptive Server Anywhere に対して変更を自動的にコミットするように指定できます。これは Interactive SQL のオプションです。AUTO_COMMIT を ON に設定すると、データベースに対して挿入文、更新文、削除文を実行するたびに Adaptive Server Anywhere がそのデータベースを更新します。

このため、パフォーマンスが大幅に低下することがあります。このような場合には、AUTO_COMMIT オプションを OFF に設定することをおすすめします。

注意して COMMIT 文を使用すること
このチュートリアルにある例を試してみる場合、データベースの変更を確定しても問題がないと確信するまでは、どのような変更に対しても COMMIT 文を実行しないように注意してください。

Interactive SQL オプションの詳細については、『ASA データベース管理ガイド』> 「Interactive SQL オプション」を参照してください。

変更のキャンセル

コミットされていない変更はすべてキャンセルできます。SQL では、ROLLBACK 文を使用することによって最後のコミット以降に加えた変更をすべて取り消せます。

ROLLBACK 文は、最後に変更を確定した後にデータベースに対して行われたすべての変更を取り消します。

Interactive SQL オプションの詳細については、『ASA データベース管理ガイド』> 「Interactive SQL オプション」を参照してください。

トランザクションとデータ・リカバリ

システム障害や停電によって突然データベース・サーバがダウンしたと仮定します。Adaptive Server Anywhere は、そのような状況でもデータベースの整合性を保護するように入念に設計されています。データベースをリストアする独自の方法がいくつも用意されています。たとえば、「**ログ・ファイル**」を別のドライブに保管し、1 つのドライブがシステム障害を起こしても、データをリストアできます。また、ログ・ファイルを使用すると、Adaptive Server Anywhere は頻繁にデータベースを更新する必要がなくなるため、データベースのパフォーマンスが向上します。

トランザクション処理により、データベース・サーバはデータが一貫性を保っていることを識別できます。トランザクション処理は、なんらかの理由でトランザクションが正常に完了しなかった場合に、トランザクション全体が取り消されるか、ロールバックしたことを確認します。トランザクションが失敗しても、データベースには影響ありません。

Adaptive Server Anywhere のトランザクション処理は、トランザクションの途中でシステムがダウンした場合でも、トランザクションの内容は確実に処理されることを保証します。

データ・リカバリ・メカニズムの詳細については、『ASA データベース管理ガイド』> 「バックアップとデータ・リカバリ」を参照してください。

整合性の検査

Adaptive Server Anywhere はデータ内の一般的なエラーについて自動的に検査を行います。

重複データの挿入

たとえば、部署を新設しようとする場合に、すでに使用されている dept_id 値を指定すると仮定します。

これを行うには、次のコマンドを入力します。

```
INSERT
 INTO department ( dept_id, dept_name, dept_head_id )
 VALUES ( 200, 'Eastern Sales', 902 )
```

この INSERT 文は、テーブルのプライマリ・キーをユニークでないものにする可能性があるため拒否されます。dept_id フィールドはプライマリ・キーなので、重複する値は許可されません。

関係に違反する値の挿入

次の文は sales_order テーブルに新しいローを挿入しますが、employee テーブル内には存在しない sales_rep ID を誤って指定します。

```
INSERT
 INTO sales_order ( id, cust_id, order_date,
  sales_rep)
 VALUES ( 2700, 186, '1995-10-19', 284 )
```

employee テーブルと sales_order テーブルには 1 対多の関係があり、sales_order テーブルの sales_rep フィールドと employee テーブルの emp_id フィールドでジョインしています。ジョインのプライマリ・キーを含むテーブル (employee テーブル) にレコードが入力されていないかぎり、外部キーを含むテーブル (sales_order テーブル) の中に対応するレコードを挿入できません。

外部キー

employee テーブルのプライマリ・キーは従業員 ID 番号です。sales_rep テーブル内の営業担当者 ID 番号は、employee テーブルの「**外部キー**」です。これは、sales_order テーブル内にあるそれぞれの営業担当者 ID 番号が、employee テーブル内にある従業員の従業員 ID 番号と必ず一致することを意味します。

営業担当者 ID 番号 284 に対して受注を追加しようとすると、次のようなエラー・メッセージが表示されます。

```
テーブル 'sales_order' の外部キー 'ky_so_employee_id' に対応するプライマリ・キーの値がありません。
```

employee テーブルには、その ID 番号を持つ従業員は存在しません。このエラーによって、有効な営業担当者 ID を持たない受注の挿入が禁止されます。このタイプの妥当性検査は、データベースに含まれるテーブル間での参照の整合性が維持されていることを検査するので、「**参照整合性**」検査と呼ばれます。

プライマリ・キーと外部キーの詳細については、「テーブル間の関係」160 ページを参照してください。

DELETE 文または UPDATE 文でのエラー

更新オペレーションまたは削除オペレーションを行う場合にも、外部キー・エラーが発生する可能性があります。たとえば、department テーブルから R&D 部を削除するとします。department テーブルのプライマリ・キーである dept_id フィールドは 1 対多の関係の「1」の側を構成します (employee テーブルの dept_id フィールドは対応する外部キーであり、1 対多の関係の「多」の側を構成します)。1 対多の関係の「1」のレコードは、対応する「多」のレコードすべてが削除されるまで削除できません。

```
DELETE
 FROM department
 WHERE dept_id = 100
```

例 : DELETE エラー

R&D 部を参照するその他のレコードがデータベース内にあることを示す次のようなエラーがレポートされて、削除オペレーションが実行されません。

テーブル 'department' のプライマリ・キーは他のテーブルから参照されています。

R&D 部を削除するには、次のコマンドを使用して、その部署に所属しているすべての従業員を削除する必要があります。

```
DELETE
 FROM employee
 WHERE dept_id = 100
```

これで、R&D 部の削除を実行できます。

次の ROLLBACK 文を入力して、(今後の使用のために) データベースのこれらの変更をキャンセルしてください。

```
ROLLBACK WORK
```

最後に実行した COMMIT WORK 文以降に行った変更は、すべて取り消されます。COMMIT 文を実行していない場合は、Interactive SQL の起動以降に行った変更がすべて取り消されます。

例 : UPDATE エラー

たとえば、employee テーブルの dept_id フィールドを変更するとします。dept_id フィールドは employee テーブルの外部キーなので、1 対多の関係の「多」の側を構成します (対応するプライマリ・キーは department テーブルの dept_id フィールドで、「1」の側を構成します)。1 対多の関係の「多」の側のレコードは、「1」の側のレコードに対応しない限り、つまり 参照するプライマリ・キーがなければ変更できません。

たとえば次の UPDATE 文を実行すると、整合性エラーが発生します。

```
UPDATE employee
 SET dept_id = 600
 WHERE dept_id = 100
```

「テーブル 'employee' の外部キー 'ky_dept_id' に対応するプライマリ・キーの値がありません」というエラーメッセージが表示されるのは、600 のプライマリ・キーを持つ部署が department テーブルにないからです。

employee テーブルの dept_id フィールドの値を変更するには、department テーブルの既存の値に対応する必要があります。例 :

```
UPDATE employee
 SET dept_id = 300
 WHERE dept_id = 100
```

この文は、dept_id 300 は既存の Finance 部に対応するので実行できます。

次の ROLLBACK 文を入力して、データベースのこれらの変更をキャンセルしてください。

```
ROLLBACK WORK
```

COMMIT WORK 文の実行完了後の整合性検査

前述のすべての例で、それぞれのコマンドの実行時にデータベースの整合性が検査されています。データベースの整合性を失わせる可能性があるオペレーションは実行されません。

COMMIT WORK 文が実行されるまでは、整合性が検査されないようにデータベースを設定できます。この設定は、参照されるプライマリ・キーの値を変更する (たとえば、department テーブルと employee テーブルで R&D 部 を削除する) 場合に重要です。このような削除を実行するには、最初のテーブルの変更から次のテーブルの変更までの間、データベースが不整合になります。この場合、コミット時にのみ検査するようにデータベースを設定してください。

詳細については、『ASA データベース管理ガイド』> 「WAIT_FOR_COMMIT オプション [データベース]」を参照してください。

また、プライマリ・キーに行われた変更に合わせて外部キーが自動的に修正されるように定義することもできます。上の例では、employee テーブルから department テーブルへの外部キーが ON DELETE CASCADE を使用して定義された場合、部署 ID を削除すると、employee テーブルの対応するエントリが自動的に削除されます。

ここまでの例では、整合性のないデータベースが確定的なものとしてコミットされることはありません。変更することによってデータベースの整合性が失われる可能性がある場合、Adaptive Server Anywhere では代替の動作もサポートされています。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「データ整合性の確保」の章を参照してください。

第 4 部　SQL Anywhere Studio のチュートリアル

第 4 部は、SQL Anywhere Studio のすべてのコンポーネントを使用する上で役に立つチュートリアルで構成されています。各章は、テーマごとに独立した構成になっています。

第 12 章

サンプル・データベース

この章の内容

この章では、データベース・サーバとサンプル・データベースの基本情報について説明します。サンプル・データベースのコピーの作成、サンプル・データベースを実行するデータベース・サーバの起動、データベース・サーバ・ウィンドウの表示、サーバの停止の方法について説明します。

サンプル・データベースについて

Adaptive Server Anywhere データベース・サーバは、ソフトウェアの中でデータベースを管理する部分です。データベース・サーバを使用すると、クライアント・アプリケーションのデータベースへのアクセスや、コマンドの処理を安全に効率良く実行できます。ある 1 つのデータベースを管理できるのは、一度に 1 つのデータベース・サーバだけです。

データベースへのアクセスは、すべてデータベース・サーバを通じて行う必要があります。

2 種類のデータベース・サーバ

Adaptive Server Anywhere データベース・サーバには、2 つのバージョンがあります。「**パーソナル・サーバ**」は、同じマシン上で稼働するアプリケーションやユーザからの接続だけを受け入れることができます。このサーバは、同一マシンを使用するシングルユーザ用です。

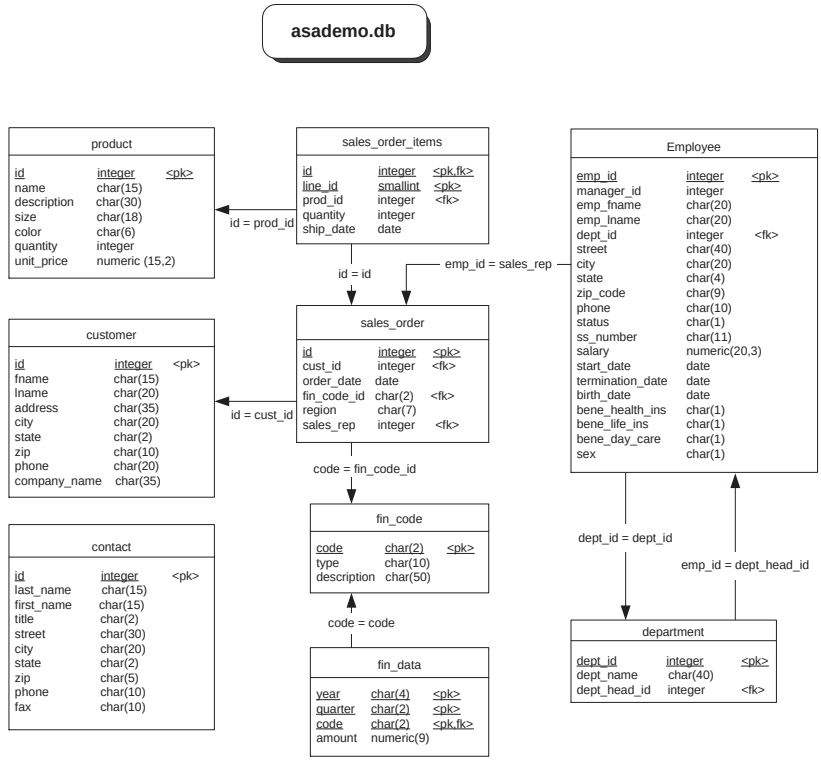
これに対して、「**ネットワーク・サーバ**」は、ネットワーク経由のクライアント／サーバ通信をサポートしているマルチユーザ用です。

この 2 つのデータベース・サーバのクエリ処理やその他の内部オペレーションはまったく同じです。パーソナル・データベース・サーバとネットワーク・データベース・サーバの性能は同じです。

サンプル・データベース

このチュートリアルでは、サンプル・データベースに重点を置きます。サンプル・データベースは、限られた種類のスポーツ衣料品を製造する小企業を想定して作られています。データベースには、この企業の内部情報 (従業員、部署、財務データ) とともに、製品情報 (製品) や販売情報 (受注、顧客、連絡先) が入っています。サンプル・データベースに入っている情報は、すべて架空のものです。

次の図は、サンプル・データベース内のテーブルと各テーブル間の関係を示します。四角はテーブル、矢印は外部キーの関係を表します。プライマリ・キー・カラムにはアンダーラインが引かれています。


asademo.db
product
id integer <pk>
name char(15)
description char(30)
size char(18)
color char(6)
quantity integer
unit_price numeric (15,2)
sales_order_items
id integer <pk,fk>
line_id smallint <pk>
prod_id integer <fk>
quantity integer
ship_date date
Employee
emp_id integer <pk>
manager_id integer
emp_fname char(20)
emp_lname char(20)
dept_id integer <fk>
street char(40)
city char(20)
state char(4)
zip_code char(9)
phone char(10)
status char(1)
ss_number char(11)
salary numeric(20,3)
start_date date
termination_date date
birth_date date
bene_health_ins char(1)
bene_life_ins char(1)
bene_day_care char(1)
sex char(1)
id = prod_id
id = id
emp_id = sales_rep
customer
id integer <pk>
fname char(15)
lname char(20)
address char(35)
city char(20)
state char(2)
zip char(10)
phone char(20)
company_name char(35)
sales_order
id integer <pk>
cust_id integer <fk>
order_date date
fin_code_id char(2) <fk>
region char(7)
sales_rep integer <fk>
id = cust_id
code = fin_code_id
fin_code
code char(2) <pk>
type char(10)
description char(50)
contact
id integer <pk>
last_name char(15)
first_name char(15)
title char(2)
street char(30)
city char(20)
state char(2)
zip char(5)
phone char(10)
fax char(10)
code = code
fin_data
year char(4) <pk>
quarter char(2) <pk>
code char(2) <pk,fk>
amount numeric(9)
dept_id = dept_id
emp_id = dept_head_id
department
dept_id integer <pk>
dept_name char(40)
dept_head_id integer <fk>

レッスン 1 ：サンプル・データベースのコピーの作成

始める前に、サンプル・データベースのコピーを作成し、変更後にリストアできるようにしておきます。

❖ サンプル・データベースのコピーを作成するには、次の手順に従います。

1. サンプル・データベース・ファイルの *asademo.db* が含まれているディレクトリに移動します。

   デフォルトのインストール・ディレクトリは、*Program Files¥Sybase¥Adaptive Server Anywhere 9* です。

2. *asademo.db* のコピーを保存するサブディレクトリを作成します。

   ここでは、名前を *demoback* とします。

3. バックアップ・ディレクトリに *asademo.db* のコピーを作成します。

   オリジナル・バージョンを使用します。パックアップ・コピーは変更しないでください。

レッスン 2 ：Adaptive Server Anywhere データベース・サーバの起動

この項では、サンプル・データベースを実行する Adaptive Server Anywhere データベース・サーバを起動する方法について説明します。サンプル・データベースは、ファイル *asademo.db* に保管され、SQL Anywhere インストール・ディレクトリに入っています。

サンプル・データベースを実行するデータベース・サーバの起動方法は、使用するオペレーティング・システムに応じて選択します。

❖ サンプル・データベースを実行するパーソナル・データベース・サーバを起動するには、次の手順に従います (Windows の場合)。

- [スタート] メニューから､[プログラム] － [SQL Anywhere 9] － [Adaptive Server Anywhere] － [パーソナル・サーバのサンプル] を選択します。

  これによって、サンプル・データベースを実行するパーソナル・サーバが起動します。データベース・サーバ・ウィンドウが表示され、すぐに閉じます。このサーバは、タスク バーの [スタート] ボタンとは反対側にあるシステム・トレー内にアイコンで表示されます。

❖ サンプル・データベースを実行するデータベース・サーバを起動するには、次の手順に従います (コマンド・プロンプトの場合)。

1. コマンド・プロンプトを開きます。
2. SQL Anywhere インストール・ディレクトリに移動します。

   Windows のオペレーティング・システムでは、デフォルトのインストール・ディレクトリは *C:¥Program Files¥Sybase¥SQL Anywhere 9* です。

3 サンプル・データベースを実行するデータベース・サーバを起動します。

データベース・サーバの起動方法は、使用するオペレーティング・システムや、ネットワーク上の他のマシンからデータベースに接続するかどうかによって異なります。

- Windows または UNIX オペレーティング・システム上で同じマシンからのみ接続する場合は、次のコマンドを入力してパーソナル・データベース・サーバを起動します。

```
dbeng9 -n asademo9 asademo.db
```

- Windows または UNIX のオペレーティング・システムを使用し、ネットワーク上の他のマシンからデータベース・サーバに接続する場合は、次のコマンドを入力してネットワーク・データベース・サーバを起動します。

```
dbsrv9 -n asademo9 asademo.db
```

- NetWare の場合は、ネットワーク・データベース・サーバしかありません。次のコマンドを入力します。

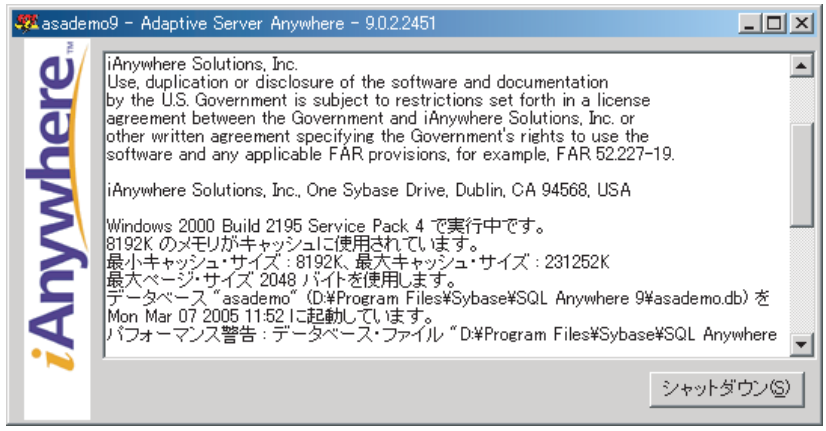
```
load dbsrv9.nlm -n asademo9 asademo.db
```

このサーバは、タスク バーの [スタート] ボタンとは反対側にあるシステム・トレー内にアイコンで表示されます。

レッスン 3 ：データベース・サーバ・ウィンドウの表示

この時点では、サンプル・データベースを実行するデータベース・サーバはすでに起動しています。ただし、まだデータベース内のデータを表示したり操作したりはできません。

実際、Adaptive Server Anywhere のアイコンは単に表示されているだけです。システム・トレー内の [Adaptive Server Anywhere] アイコンをダブルクリックすると、Windows のデータベース・サーバ・ウィンドウが表示されます。



サーバ・ウィンドウには次の情報が表示されます。

- **サーバ名**　タイトル・バー内の名前 (この例では、**asademo9**) が「**サーバ名**」です。データベース・サーバが起動すると、その度にサーバ名が割り当てられます。この名前は、アプリケーションがデータベースに接続するときに使用します。
- **バージョンとビルド番号**　サーバ名に続く数字 (この場合は **9.0.2.2451**) は、「**バージョン番号とビルド番号**」です。バージョン番号は SQL Anywhere Studio の特定のリリースを表し、ビルド番号はソフトウェアがコンパイルされた特定のインスタンスに関連します。

- **起動情報**　データベース・サーバは、起動時に、データベース要求を処理するときに使用するメモリを別に設定します。これを「**キャッシュ**」と呼びます。キャッシュ・メモリの量は、このウィンドウに表示されます。キャッシュは固定サイズの「**ページ**」で構成されていますが、このページのサイズもウィンドウ内に表示されます。

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

ここでは、起動時のキャッシュ・サイズとページのサイズはデフォルト値です。このチュートリアルの場合も含めて、多くの場合、デフォルトの起動オプションが適しています。

レッスン 4 ：データベース・サーバの停止

起動したデータベース・サーバを停止できます。

❖ **サンプル・データベースを実行しているデータベース・サーバを停止するには、次の手順に従います (Windows の場合)。**

1. Windows タスク バーにある Adaptive Server Anywhere のアイコンをダブルクリックします。

   データベース・サーバ・ウィンドウが表示されます。

2. [シャットダウン] をクリックします。

まとめ

このチュートリアルでは、サンプル・データベースのコピーの作成方法、サンプル・データベースを実行するデータベース・サーバの起動方法、データベース・サーバ・ウィンドウの表示方法について学習しました。また、データベース・サーバの停止方法についても学習しました。

第 13 章

接続の確立

この章の内容

この章では、ODBC データ・ソースの作成方法と、そのデータ・ソースを使用してデータベースとアプリケーション間の接続を確立する方法について説明します。この場合、アプリケーションは Sybase Central です。

接続について

データベースを使用するクライアント・アプリケーションでは、なんらかの作業を行う前に必ずそのデータベースへの接続を確立する必要があります。「接続」は、チャネルを形成します。クライアント・アプリケーションからのアクティビティは、すべてそのチャネルを介して行われます。たとえば、データベースではユーザ ID に応じてアクションを実行するパーミッションが決定しますが、データベース・サーバがこのユーザ ID を認識するのは、ユーザ ID が接続確立要求の一部として送信されるからです。

ODBC データ・ソースについて

「**ODBC**（**Open Database Connectivity**)」インタフェースは Microsoft によって定義されたもので、Windows の環境下でクライアント・アプリケーションからデータベース管理システムに接続するときに使用する標準インタフェースです。接続は、接続パラメータを指定して行います。通常は、一連の接続パラメータをひとまとめにして、それらを「**ODBC データ・ソース**」内に格納しておくと便利です。ODBC データ・ソースを使用すると、繰り返し使用する接続パラメータを保存できるので便利です。

データ・ソースがあれば、接続文字列では使用するデータ・ソースを次のように指定するだけで済みます。

```
DSN=my data source
```

ODBC データ・ソースを使用して Adaptive Server Anywhere データベースに接続できるアプリケーションには、次のものがあります。

- Sybase Central と Interactive SQL
- すべての Adaptive Server Anywhere ユーティリティ
- PowerDesigner と InfoMaker
- Microsoft Visual Basic、Sybase PowerBuilder、Borland Delphi など、ODBC をサポートしているアプリケーション開発環境
- UNIX 上の Adaptive Server Anywhere クライアント・アプリケーション UNIX では、データ・ソースはファイルとして保存されます。

レッスン 1 ：ODBC データ・ソースの作成

この項では、Windows で単純な ODBC データ・ソースを作成する方法について説明します。

❖ 単純な ODBC データ・ソースを作成するには、次の手順に従います。

1 ODBC アドミニストレータを起動します。

- [スタート] メニューから、[プログラム] ー [SQL Anywhere 9] ー [Adaptive Server Anywhere] ー [ODBC アドミニストレータ] の順に選択します。

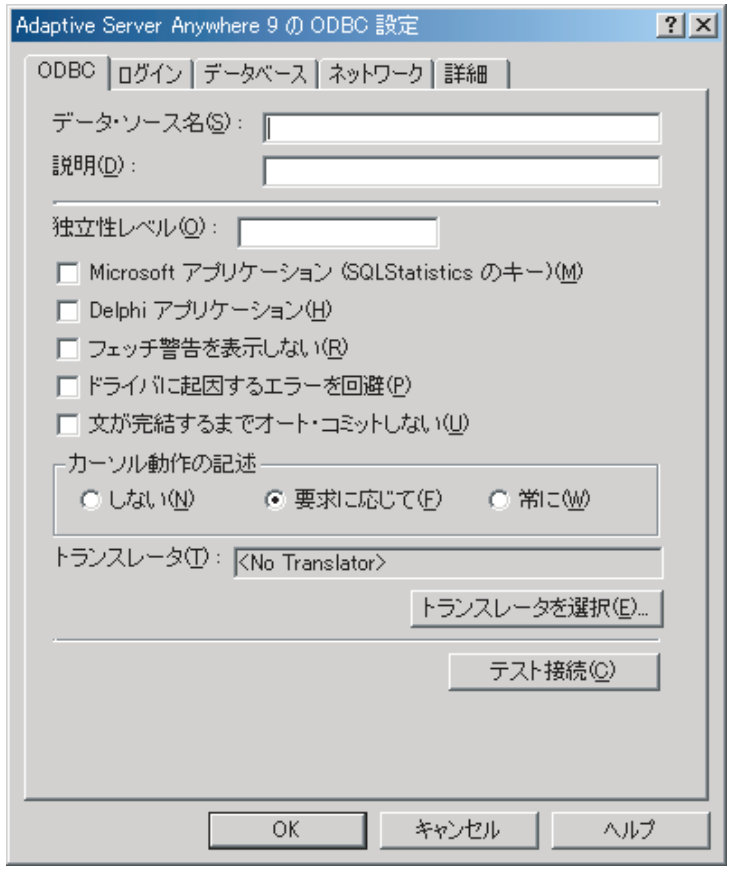
現在コンピュータにインストールされているデータ・ソースのリストが [ODBC データ ソース アドミニストレータ] ウィンドウに表示されます。

2 [ユーザー DSN] タブで [追加] をクリックします。

[データ ソースの新規作成] ウィザードが表示されます。

3　ドライバのリストから [**Adaptive Server Anywhere 9.0**] を選択し、[完了] をクリックします。

[設定] ダイアログが表示されます。



このダイアログのフィールドの多くは、オプションです。個々のタブの下にある [ヘルプ] ボタンをクリックすると、そのタブ上のすべてのフィールドの説明が表示されます。多

くの用途に使用するため、使用するパラメータは少しだけにしておきます。Adaptive Server Anywhere のサンプル・データベースには、次のデータ・ソース・パラメータで十分です。

- **データ・ソース名 ([ODBC] タブ)** サンプルのデータ・ソース名を入力します。[接続] ダイアログに表示される名前です。この名前にはスペースを入れられますが、短い名前にする必要があります。
- **ユーザ ID ([ログイン] タブ)** デフォルトのユーザ ID に「**DBA**」と入力します。接続に使用するデータベース・ユーザ ID です。
- **パスワード ([ログイン] タブ)** 「**SQL**」と入力します。DBA ユーザ ID のデフォルト・パスワードです。
- **データベース・ファイル ([データベース] タブ)** 使用中の SQL Anywhere インストール・ディレクトリを検索して *asademo.db* ファイルを選択します。このファイルが Adaptive Server Anywhere のサンプル・データベースです。

4 データ・ソースをテストします。

[ODBC] タブで [テスト接続] をクリックします。

データソースが正しく作成された場合は、「接続に成功しました」というメッセージが表示されます。

「接続に失敗しました」というメッセージが表示された場合は、データ・ソースの設定を確認し、必要な修正を行ってください。

5 [OK] をクリックしてデータ・ソースを作成します。

レッスン 2 ：Sample ODBC データ・ソースを使用した接続

この例では、Sybase Central からサンプル・データベースに接続します。Sybase Central は、データベースを管理するための Adaptive Server Anywhere のグラフィカル・ツールです。

❖ Sybase Central からサンプル・データベースに接続するには、次の手順に従います。

1. [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [Sybase Central] の順に選択して、Sybase Central を起動します。

2. [ツール] メニューから [接続] を選択して、[接続] ダイアログを開きます。

   プロンプトでプラグインを選択するように表示されたら、[Adaptive Server Anywhere 9] を選択し、[OK] をクリックします。

   [接続] ダイアログが表示されます。

3. [ODBC データ・ソース名] オプションを選択します。

4. [ODBC データ・ソース名] ボックスに「**ASA 9.0 Sample**」と入力して、[OK] をクリックします。

   または、[参照] をクリックし、[データ・ソース名] ダイアログの [ASA 9.0 Sample] を選択します。

5. [OK] をクリックして、サンプル・データベースに接続します。

   ダイアログが閉じ、Sybase Central ウィンドウにデータベースが開きます。

レッスン 3 ：他の接続方法

ODBC アプリケーションからだけでなく、管理ユーティリティなどの Embedded SQL アプリケーションや ODBC を使用しない Interactive SQL や Sybase Central からも、ODBC データ・ソースを使用して Adaptive Server Anywhere に接続できます。ODBC データ・ソースの処理機能は、Embedded SQL クライアント・ライブラリ、Sybase Central、Interactive SQL に組み込まれています。

Adaptive Server Anywhere は、ODBC のほかにもいくつかのプログラミング・インタフェースをサポートしています。詳細については、『ASA データベース管理ガイド』> 「接続の概要」を参照してください。

Sybase Central と Interactive SQL の [接続] ダイアログには、ODBC データ・ソース名または ODBC データ・ソース・ファイルを入力するフィールドがあります。

レッスン 4 ：サンプル・データベースからの切断

このチュートリアルを終了するには、データベース・サーバを停止して、Sybase Central を終了します。

❖ データベース・サーバを停止するには、次の手順に従います。

- [ツール] メニューから、[切断] を選択します。

  データベース・サーバがサンプル・データベースから切断され、停止します。

❖ Sybase Central を終了するには、次の手順に従います。

- [ファイル] メニューから [終了] を選択します。

  Sybase Central が停止します。

まとめ

このチュートリアルでは、ODBC データ・ソースの作成方法と、そのデータ・ソースを使用してデータベースと Sybase Central 間の接続を確立する方法を学習しました。

第 14 章

Interactive SQL の使用

この章の内容

この章では、Interactive SQL を起動し、使用する方法について説明します。Interactive SQL は Adaptive Server Anywhere 付属のユーティリティで、SQL 文の実行、スクリプトの構築、データベースのデータの表示に使用します。

Interactive SQL について

Interactive SQL は、次の目的で使用できます。

- SQL 文のデータベース・サーバへの送信
- データベース内の情報のブラウズ
- アプリケーションで使用する予定のある SQL 文のテスト
- データベースへのデータの読み込み
- 管理タスクの実行

また、Interactive SQL は「**コマンド・ファイル**」または「**スクリプト・ファイル**」を実行できます。たとえば、データベースに対して繰り返し実行可能なスクリプトを作成し、Interactive SQL を使用してこれらのスクリプトをバッチとして実行できます。

レッスン 1 ：Interactive SQL の起動

この項では、Interactive SQL ユーティリティの起動、サンプル・データベースへの接続、コマンドの入力方法について説明します。

❖ Interactive SQL を起動して Interactive SQL からサンプル・データベースに接続するには、次の手順に従います (Windows の場合)。

1. [スタート] メニューから、[プログラム] － [Sybase SQL Anywhere 9] － [Adaptive Server Anywhere] － [Interactive SQL] の順に選択します。

   [接続] ダイアログ・ボックスが表示されます。

2. [ODBC データ・ソース名] オプションを選択します。

3. [参照] をクリックし、リストから [**ASA 9.0 Sample**] を選択します。

4. [OK] をクリックします。

5. [OK] をクリックして、サンプル・データベースに接続します。

❖ Interactive SQL を起動してサンプル・データベースに接続するには、次の手順に従います (コマンド・プロンプトの場合)。

1. Windows または UNIX オペレーティング・システムを使用している場合は、次のコマンドを入力して、パーソナル・データベース・サーバを起動します。

   ```
   dbeng9 -n asademo9
   ```

2. コマンド・プロンプトで次のように入力して、Interactive SQL を起動しサンプル・データベースに接続します。

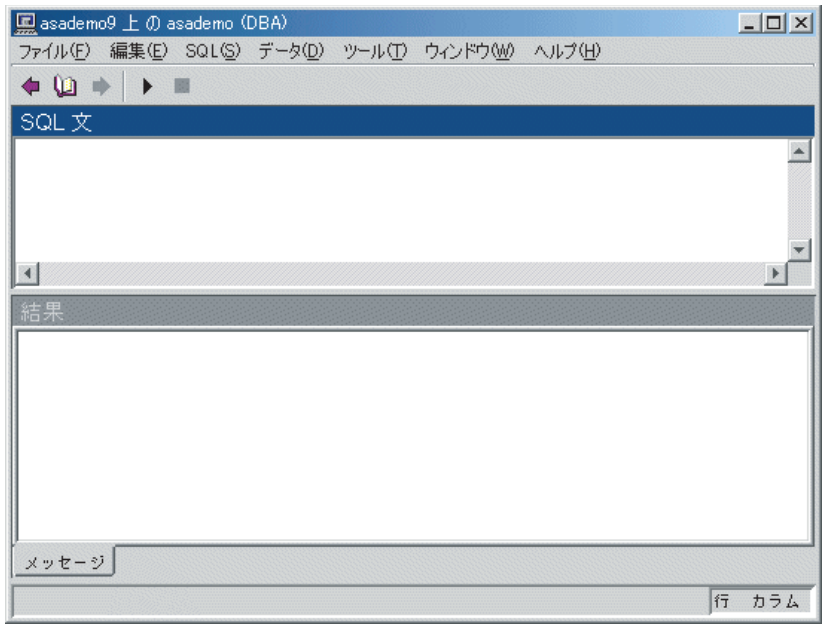
   ```
   dbisql -c "DSN=ASA 9.0 Sample"
   ```

レッスン 2 ：Interactive SQL のインタフェース

この項では、Interactive SQL メイン・ウィンドウの構成要素、Interactive SQL のキーボード・ショートカット、Interactive SQL で使用できるツールについて学習します。

Interactive SQL メイン・ウィンドウの説明

データベースに接続すると、[Interactive SQL] ウィンドウが表示されます。



ウィンドウの最上部にあるタイトル・バーには、次のように接続情報が表示されます。

server-name 上の *database-name* （*userid*）

たとえば、ASA 9.0 Sample ODBC を使用してサンプル・データベースに接続すると、タイトル・バーは次のようになります。

```
asademo9 è„ÇÃ asademo (DBA)
```

Interactive SQL のウィンドウ枠

Interactive SQL には、次のウィンドウ枠があります。

- **[SQL 文]**　このウィンドウ枠には、データのアクセスや変更に使用する SQL 文を入力します。
- **[結果]**　このタブには、実行したコマンドの結果が表示されます。たとえば、データベースから特定のデータを検索する SQL 文を使用すると、その上のウィンドウ枠の [結果] タブに、検索条件と一致するカラムとローが表示されます。表示される情報がウィンドウ枠よりも長い場合は、スクロール・バーが自動的に表示されます。[結果] タブに表示された結果セットは編集できます。

  結果セットの編集については、「Interactive SQL でのテーブル値の編集」282 ページを参照してください。
- **メッセージ**　このタブには、Interactive SQL で実行する SQL 文に関するデータベース・サーバのメッセージが表示されます。
- **プラン**　このタブと [Ultra Light プラン] タブには、SQL 文に関するクエリ・オプティマイザの実行プランが表示されます。

Interactive SQL メイン・ウィンドウのタブとウィンドウ枠のオプションは、[ツール] メニューの [オプション] ダイアログで設定できます。

Interactive SQL ツールバーの使用

Interactive SQL ツールバーは、Interactive SQL ウィンドウの一番上に表示されます。ここには、一般コマンドを実行するためのボタンがあります。このツールバーのボタンを使用して、次の作業を実行できます。

- 履歴リストに示されている実行した SQL 文の中で、現在の位置の直前の文を呼び戻します。
- これまでに実行した最大 50 までの SQL 文をリストにして表示します。

- 履歴リストに示されている実行した SQL 文の中で、現在の位置の直後の文を呼び戻します。
- 現在 [SQL 文] ウィンドウ枠に表示されている SQL 文を実行します。
- 現在の SQL 文の実行を中断します。

各ボタンの上にカーソルを移動するとポップアップ・ヒントが表示され、簡単にボタンの機能を確認できます。

複数のウィンドウを開く

複数の Interactive SQL ウィンドウを開くことができます。各ウィンドウは、接続するデータベース別になっています。異なるサーバ上にある 2 つ (またはそれ以上) のデータベースに同時に接続したり、単一のデータベースへの同時接続を開始したりできます。

❖ **新しい Interactive SQL ウィンドウを開くには、次の手順に従います。**

1. [ウィンドウ] メニューから、[新しいウィンドウ] を選択します。

   [接続] ダイアログ・ボックスが表示されます。

ヒント
SQLCONNECT 環境変数が設定されている場合や、すでにデータベースに接続している場合は、情報の入力を求める前に、サーバがこの情報を使ってデータベースに接続しようとします。それに失敗した場合、またはデータベースにまだ接続していない場合に、[接続] ダイアログが表示されます。

---

2. [接続] ダイアログに接続オプションを入力し、[OK] をクリックして接続します。

   接続情報 (データベース名、ユーザ ID、データベース・サーバ名など) が [SQL 文] ウィンドウ枠の上のタイトル・バーに表示されます。

データベースとの接続を開始または解除するには、[SQL] メニューの接続コマンドと切断コマンドを使用する方法、または CONNECT 文または DISCONNECT 文を実行する方法もあります。

Interactive SQL キーボード・ショートカット

Interactive SQL には、以下のキーボード・ショートカットがあります。

ファンクション・キー	説明
[ALT + F4]	Interactive SQL を終了する
[ALT + 左矢印]	履歴リストにある、前の SQL 文を表示する
[ALT + 右矢印]	履歴リストにある、次の SQL 文を表示する
[CTRL + BREAK]	実行中の SQL 文を中断する
[CTRL + C]	[結果] ウィンドウ枠では、選択したローとカラム見出しをクリップボードにコピーする。 [SQL 文] ウィンドウ枠では、選択したテキストをクリップボードにコピーする。
[CTRL + END]	現在のウィンドウ枠の一番下に移動する
[CTRL + H]	実行した SQL の履歴を表示する
[CTRL + HOME]	現在のウィンドウ枠の一番上に移動する
[CTRL + N]	Interactive SQL ウィンドウの内容をクリアする
CTRL + P	[SQL 文] ウィンドウ枠の内容を印刷する。印刷テキストの外観は、Interactive SQL の [オプション] ダイアログで設定できます。 印刷オプションの設定については、『SQL Anywhere Studio ヘルプ』> 「[印刷] タブ」を参照してください。

ファンクション・キー	説明
[CTRL + Q]	クエリ・エディタを表示する。 クエリ・エディタを使用すると、SQL クエリを作成できます。クエリを構築し終わったら、[OK] をクリックして、[SQL 文] ウィンドウ枠にエクスポートします。
[CTRL + S]	[SQL 文] ウィンドウ枠の内容を保存する
[ESC]	[SQL 文] ウィンドウ枠をクリアする
[F1]	ヘルプを表示する
[F2]	結果セット内の選択した値を編集する。ローのカラム間でタブ移動できます。
[F5]	[SQL 文] ウィンドウ枠にあるすべてのテキストを実行する。 この操作は、ツールバーで [SQL 文の実行] ボタンをクリックしても実行できます。
[F7]	[テーブル名のルックアップ] ダイアログを表示する。 このダイアログでテーブルを検索して選択し、[ENTER] を押せば、そのテーブル名が [SQL 文] ウィンドウ枠のカーソル位置に入力されます。また、リストでテーブルを選択し、もう一度 [F7] を押すとそのテーブルのカラムが表示されます。この後カラムを選択して [ENTER] キーを押せば、そのカラム名が [SQL 文] ウィンドウ枠のカーソル位置に入力されます。
[F8]	[プロシージャ名のルックアップ] ダイアログを表示する。 このダイアログでプロシージャを検索して選択し、[ENTER] キーを押せば、そのプロシージャ名が [SQL 文] ウィンドウ枠のカーソル位置に入力されます。

ファンクション・キー	説明
[F9]	[SQL 文] ウィンドウ枠で選択されているテキストを実行する。 テキストが選択されていない場合は、文全体が実行されます。
[PAGE DOWN]	現在のウィンドウ枠を 1 ページ下にスクロールする
[PAGE UP]	現在のウィンドウ枠を 1 ページ上にスクロールする
[SHIFT + F5]	[SQL 文] ウィンドウ枠内の文を実行せずに、その文のプランを表示する
[SHIFT + F10]	フォーカスのある領域のコンテキスト・メニューを表示する。 このキーボード・ショートカットは、領域を右クリックする代わりに使用します。

[SQL 文] ウィンドウ枠にフォーカスがあるときは、以下のキーボード・ショートカットを使用できます。

ファンクション・キー	説明
[CTRL +]]	カーソルを閉じカッコまで移動する。カッコの照合はカッコ ()、大カッコ {}、角カッコ []、山カッコ <> に一致します。
[CTRL + BACKSPACE]	カーソルの左の 1 ワードを削除する
[CTRL + DEL]	カーソルの右の 1 ワードを削除する
[CTRL + G]	[ジャンプ] ダイアログが開き、ジャンプ先の行を指定できる
[CTRL + L]	現在の行を [SQL 文] ウィンドウ枠から削除してクリップボードに入れる
[CTRL + SHIFT +]]	選択範囲を閉じカッコまで拡張する。カッコの照合はカッコ ()、大カッコ {}、角カッコ []、山カッコ <> に一致します。

ファンクション・キー	説明
[CTRL + SHIFT + L]	現在の行を削除する
[CTRL + SHIFT + U]	選択している内容を大文字に変換する
[CTRL + U]	選択している内容を小文字に変換する
[F3]	選択したテキストの次の出現箇所を検索する
[HOME]	カーソルを現在の行の行頭、または現在の行の最初の単語に移動する
[SHIFT + F3]	選択したテキストの 1 つ前の出現箇所を検索する
[SHIFT + HOME]	選択範囲を現在の行のテキストの先頭まで拡張する

Interactive SQL の詳細

Interactive SQL では、表示と動作を設定するための一連のウィンドウが表示されます。

詳細については、『SQL Anywhere Studio ヘルプ』> 「Interactive SQL のヘルプ」を参照してください。

Interactive SQL からインデックス・コンサルタントにアクセスできます。インデックス・コンサルタントは Sybase Central からも使用できます。使用することでクエリのパフォーマンスが向上します。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「インデックス・コンサルタントの概要」を参照してください。

Interactive SQL からクエリ・エディタにもアクセスできます。クエリ・エディタを使用すると、あらゆる種類のクエリを設計、分析、テストできます。

詳細については、『SQL Anywhere Studio ヘルプ』> 「クエリ・エディタのヘルプ」を参照してください。

レッスン 3 ：Interactive SQL を使用したデータの表示

Interactive SQL の重要な使用目的の 1 つは、テーブル・データをブラウズすることです。この項では、サンプル・データベースの情報を問い合わせる方法について説明します。クエリは SQL 文の形式です。

データベース情報は、Interactive SQL で SELECT 文を使って表示できます。次の例は、[SQL 文] ウィンドウ枠に入力するコマンドを示しています。コマンドを入力したら、ツールバーにある [SQL 文の実行] ボタンをクリックしてコマンドを実行します。

文を実行すると、データ (結果セットと呼びます) が [結果] ウィンドウ枠の [結果] タブに表示されます。現在の枠内に表示されていない部分のデータを表示するには、スクロール・バーを使って、画面をスクロールしてください。デフォルトでは、ロー番号が結果セットの左に表示されます。

❖ 従業員テーブルのカラムとローをすべてリストするには、次の手順に従います。

1. まだサンプル・データベースに接続していない場合は、接続します。

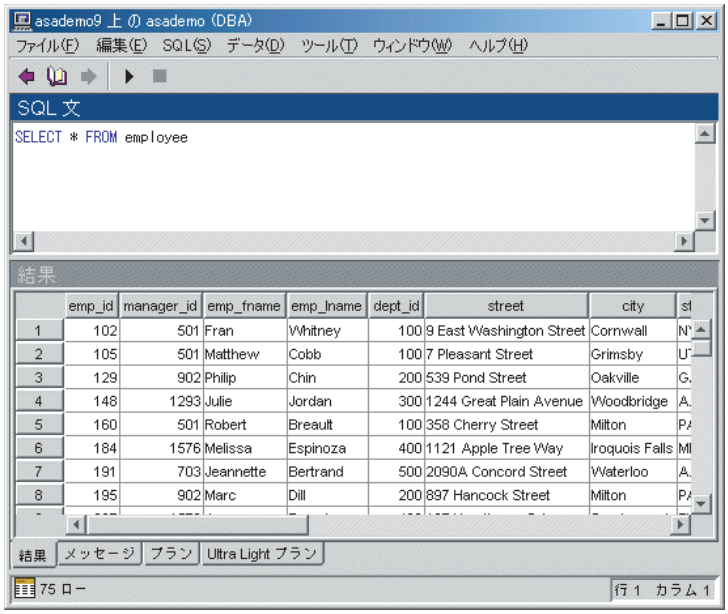
   サンプル・データベースへの接続方法については、「レッスン 1 ：Interactive SQL の起動」273 ページを参照してください。

2. [SQL 文] ウィンドウ枠で、次の SQL クエリを入力します。

```
SELECT * FROM employee
```

3. [F5] を押して、上記の文を実行します。

   このクエリを実行すると、employee という名前のテーブルのカラムがすべて表示されます。このテーブルには、仮想の商品販売会社の従業員についての情報が入っています。次のように表示されます。



Interactive SQL では、データベース・サーバに要求を送信することによって情報を検索します。一方、データベースサーバは、従業員テーブル内の情報を調べ、それを Interactive SQL に返します。

この時点で、サンプル・データベースにある他のテーブル (product、department、sales_order など) のデータを参照します。

結果セット内のローを追加、削除、更新できます。

結果セットの編集については、「Interactive SQL でのテーブル値の編集」282 ページを参照してください。

Interactive SQL でのテーブル値の編集

Interactive SQL でクエリを実行すると、結果セットを編集してデータベースを修正できます。また、結果セットからローを選択し、他のアプリケーションで使用できるようにコピーすることもできます。

Interactive SQL は、ローの編集、挿入、削除をサポートしています。これらの操作の影響は、UPDATE、INSERT、DELETE の各文を実行した場合と同じです。

結果セット内のローまたは値を編集するには、値を修正するテーブルまたはカラムに対する適切なパーミッションが必要です。たとえば、ローを削除する場合は、そのローが属しているテーブルに対する DELETE パーミッションが必要です。

次の場合には、結果セットを編集できません。

- プライマリ・キーを持つテーブルからカラムを選択したが、一部のプライマリ・キー・カラムを選択していない。
- JOIN の結果セットを編集しようとした (たとえば、結果セットに複数のテーブルのデータがある場合)。

次の場合には、結果セットの編集に失敗することがあります。

- パーミッションのないローやカラムを編集しようとした。
- 無効な値を入力した (たとえば、数値カラムに文字列を入力したり、NULL を使用できないカラムに NULL を入力した場合)。

編集に失敗すると、エラーを説明する Interactive SQL エラー・メッセージが表示されます。データベース・テーブルの値は変更されません。

テーブルの値を変更したら、COMMIT 文を実行して変更を永続的なものにします。変更を取り消す場合は、ROLLBACK 文を実行します。

詳細については、『ASA SQL リファレンス・マニュアル』>「COMMIT 文」と 『ASA SQL リファレンス・マニュアル』>「ROLLBACK 文」を参照してください。

Interactive SQL の結果セットからテーブル値を編集する

Interactive SQL から、データベース・テーブルに存在するローの一部またはすべての値を変更できます。修正するカラムに対する UPDATE パーミッションが必要です。結果セットを編集する場合、一度に変更できるのは 1 つのローの値だけです。

❖ 結果セット内のローを編集するには、次の手順に従います。

1. 変更する値をクリックします。
2. 値を右クリックし、ポップアップ・メニューから [編集] を選択します。

   [F2] キーを押して結果セットを編集することもできます。

   その値を含むテーブル・セルに点滅するカーソルが表示されます。
3. 新しい値を入力します。

   カラムには、無効なデータ型は入力できません。たとえば、INT データ型を受け入れるカラムには、文字列は入力できません。

   ローの値の編集が終了したら、[Enter] キーを押してデータベースを更新します。そのローの他の値を変更する場合は、[TAB] または [SHIFT + TAB] キーを押して他の値に移動します。

   [ESC] キーを押すと、選択した値に対して行った変更をキャンセルできます。
4. COMMIT 文を実行し、テーブルに対する変更を永続的なものにします。

計算カラムの編集

結果セット内の値を編集すると、データベースは変更した値で更新されます。計算カラムは、結果セットの一部であるかどうかにかかわらず、変更した値に基づいて再計算されます。結果セット内に計算カラムがある場合に計算カラムの値を変更すると、変更した値でデータベースが更新されます。

Interactive SQL の結果セットからデータベースにローを挿入する

Interactive SQL では、結果セットに新しいローを追加できます。結果セット内のカラム間をタブで移動し、ローに新しい値を追加します。テーブルに値を追加すると、入力した文字は大文字と小文字を区別して格納されます。新しいローを追加するには、テーブルに対する INSERT パーミッションが必要です。

❖ 結果セットに新しいローを挿入するには、次の手順に従います。

1. 結果セットを右クリックし、ポップアップ・メニューで [追加] を選択します。

   または、[CTRL+F10] を押して、コンテキスト・メニューから [追加] を選択することもできます。

   新しい空白のローが表示され、そのローの最初の値に点滅するカーソルが表示されます。

   [TAB] キーを押して、カーソルをローのカラム間で移動します。選択したローの値をクリックしても、値を挿入できます。

2. 新しい値を入力します。

   カラムには、無効なデータ型は入力できません。たとえば、INT データ型を受け入れるカラムには、文字列は入力できません。

3. [TAB] キーを押して次のカラムに進みます。

4. すべてのカラム値が追加されるまで手順 2 と 3 を繰り返します。

5. [ENTER] キーを押してデータベースを更新します。

6. COMMIT 文を実行し、テーブルに対する変更を永続的なものにします。

計算カラムへの値の挿入

結果セット内に計算カラムがあり、計算カラムに値を指定しない場合、値はデータベースが更新されたときに計算されます。しかし、計算カラムに値を指定した場合は、指定した値でデータベースが更新され、値は計算カラムに対して計算されません。

INPUT 文を使用した新しいローの挿入

Interactive SQL の結果セットから新しいローを挿入するには、INPUT 文で PROMPT 句を使用してローを追加するという方法もあります。PROMPT 句を指定すると、Interactive SQL によって、テーブルの各カラムの値を入力するよう要求されます。たとえば、product テーブルに新しいローを追加し、各カラムの値を入力するよう求めるメッセージを表示するには、Interactive SQL で次の文を実行します。

```
INPUT INTO product
 PROMPT
```

Interactive SQL を使用してデータベースからローを削除する

また、Interactive SQL では、データベース・テーブルからローを削除できます。ローを削除するには、テーブルに対する DELETE パーミッションが必要です。結果セット内で選択できるのは、連続するローだけです。

❖ 結果セットからローを削除するには、次の手順に従います。

1. 削除するローを選択します。ローを選択するには、次の手順に従います。
   - [SHIFT] キーを押しながらローをクリックします。
   - [SHIFT + 上矢印] または [SHIFT + 下矢印] キーを押します。

   ローが連続していない場合は、それぞれ個別に削除します。
2. 結果セットを右クリックし、ポップアップ・メニューで [削除] を選択します。[DELETE] キーでも、選択したローを削除できます。

   選択したローがデータベース・テーブルから削除されます。
3. COMMIT 文を実行し、テーブルに対する変更を永続的なものにします。

Interactive SQL の結果セットからローをコピーする

Interactive SQL に表示される結果セットからローを直接コピーし、他のアプリケーションにペーストできます。ローをコピーすると、カラム見出しもコピーされます。コピーしたデータはカンマで区切られ、Microsoft Excel などのアプリケーションで適切にフォーマットできます。コピーしたデータは ASCII フォーマットになり、すべての文字列は一重引用符で囲まれます。結果セット内で選択できるのは、連続するローだけです。

❖ Interactive SQL の結果セットからローをコピーするには、次の手順に従います。

1. コピーするローを選択します。ローを選択するには、次の手順に従います。
   - [SHIFT] キーを押しながらローをクリックします。
   - [SHIFT + 上矢印] または [SHIFT + 下矢印] キーを押します。
2. 結果セットを右クリックし、ポップアップ・メニューで [コピー] を選択します。[CTRL + C] キーを押して、選択したローをコピーすることもできます。

   選択したローが、カラム見出しを含めてクリップボードにコピーされます。それを、他のアプリケーションにペーストできます。

結果セットからの個々の値のコピー

結果セットから単一の値をコピーするには、値を選択し、結果セットを右クリックし、ポップアップ・メニューで [セルのコピー] を選択します。この操作を行った場合、カラム見出しはコピーされず、データのみがコピーされます。

レッスン 4 ：SQL 文の使用

以下の各項では、Interactive SQL で使用できるコマンドのいくつかについて説明します。この項では、Interactive SQL でコマンドを使用するために必要な一般的なタスクについて説明します。

すべての SQL 文はコマンドとして、Interactive SQL ウィンドウの一番上の枠に入力できます。入力が終わったらその文を実行して結果を表示します。

Interactive SQL からは、データベースに対してどのような SQL 文でも入力できます。SQL 文の完全なリストについては、『ASA SQL リファレンス・マニュアル』> 「 SQL 文」を参照してください。

❖ **SQL 文を実行するには、次の手順に従います。**

- [SQL 文の実行] ボタンをクリック*するか*、[SQL] － [実行] を選択するか、*または* ［F5] キーを押します。

❖ **[SQL 文] ウィンドウ枠をクリアするには、次の手順に従います。**

- [編集] － [SQL のクリア] を選択するか、*または* ［Esc] キーを押します。

Interactive SQL コマンドをキャンセルする

Interactive SQL ツールバーで [中断] ボタンをクリックすると、コマンドをキャンセルできます。

キャンセル操作によって現在の処理が停止し、次のコマンドのプロンプトが表示されます。コマンド・ファイルが処理中の場合、または [SQL 文] ウィンドウ枠に複数の文がある場合は、必要なアクション (コマンド・ファイルを停止するか、継続するか、Interactive SQL を終了する) を指定するためのプロンプトが表示されます。これらのアクションは、Interactive SQL の ON_ERROR オプションを使用して制御できます。

ON_ERROR オプションの詳細については、『ASA データベース管理ガイド』> 「ON_ERROR オプション [Interactive SQL]」を参照してください。

レポートされるメッセージ

中断が検出されると、検出されたタイミングによって、3 種類のメッセージのうち 1 つがレポートされます。

1. データベース・サーバではなく Interactive SQL が要求を処理しているときに中断が検出されると、次のメッセージが表示されます。

   ```
   ユーザが ISQL コマンドを終了しました
   ```

   Interactive SQL はすぐに処理を停止し、現在のデータベース・トランザクションだけはそのまま残ります。

2. データベース・サーバがデータ操作コマンド (SELECT、INSERT、DELETE、または UPDATE) を処理しているときに中断が検出されると、次のメッセージが表示されます。

   ```
   文の実行がユーザによって中断させられました。
   ```

   現在のコマンドの効果は取り消されますが、トランザクションの残りの部分はそのまま残ります。

3. データベース・サーバがデータ定義コマンド (CREATE、DROP、ALTER など) を処理している間に割り込みが検出された場合は、次のメッセージが表示されます。

   ```
   ユーザによって中断させられました。-- トランザクションはロールバックされました。
   ```

   データ定義コマンドはすべて、コマンドが起動する前に自動的に COMMIT を実行するため、ROLLBACK の影響は、現在のコマンドをキャンセルするだけになります。

   また、このメッセージは、データベース・サーバがデータベース修正コマンド (INSERT、UPDATE、または DELETE) を実行するバルク・オペレーション・モードで実行されているときにも発生します。この場合、ROLLBACK は、現在のコマンドだけでなく、

最後の COMMIT 以降に実行されたすべての作業をキャンセルします。データベース・サーバが自動 ROLLBACK を実行するには、かなりの時間がかかる場合があります。

複数の文の実行

Interactive SQL 環境では、複数の文を一度に入力できます。文を連続して入力するには、各文の末尾にコマンド・デリミタを付けます。コマンド・デリミタは、Interactive SQL で設定できるオプションであり、COMMAND_DELIMITER オプションを使用して変更できます。デフォルトでは、セミコロン (;) です。

詳細については、『ASA データベース管理ガイド』> 「COMMAND_DELIMITER オプション [Interactive SQL]」を参照してください。

❖ [SQL 文] ウィンドウ枠で複数の文を入力するには、次の手順に従います。

1 SQL 文のウィンドウ枠に次の 3 つのコマンドを入力します。

```
UPDATE employee
 SET dept_id = 400,
     manager_id = 1576
 WHERE emp_id = 467;

 UPDATE employee
 SET dept_id = 400,
     manager_id = 1576
 WHERE emp_id = 195;

 SELECT *
 FROM employee
 WHERE emp_id IN ( 195, 467 );
```

2 ツールバーにある [SQL 文の実行] ボタンをクリックします。3 つの文がすべて実行されます。実行された後も、コマンドは [SQL 文] ウィンドウ枠に残ります。

3 [SQL 文] ウィンドウ枠に「ROLLBACK」と入力して文を実行し、変更をロールバックします。

go の代用

セミコロンを使用する代わりに、行頭に「go」を単独で入力できます。

```
UPDATE employee
 SET dept_id = 400,
     manager_id = 1576
 WHERE emp_id = 467
 go

 UPDATE employee
 SET dept_id = 400,
     manager_id = 1576
 WHERE emp_id = 195
 go

 SELECT *
 FROM employee
 WHERE emp_id IN ( 195, 467 )
 go
```

ヒント
[F9] キーを押すと、[SQL 文] ウィンドウ枠で選択したテキストだけを実行できます。

テーブル、カラム、プロシージャの検索

Interactive SQL にコマンドを入力する際現在のデータベースに格納されているテーブル名、カラム名、またはプロシージャ名を検索し、それをカーソル位置に挿入することができます。

❖ データベース内のテーブル名を検索するには、次の手順に従います。

1. [ツール] － [テーブル名のルックアップ] を選択するか、[F7] キーを押します。
2. テーブルを検索し、選択します。
3. [OK] をクリックして、テーブル名を [SQL 文] ウィンドウ枠の現在のカーソル位置に挿入します。

❖ データベース内のカラム名を検索するには、次の手順に従います。

1 [ツール] ー [テーブル名のルックアップ] を選択するか、[F7] キーを押します。

2 カラムを含むテーブルを検索し選択します。

3 [カラムを表示] をクリックします。

4 カラムを選択し [OK] をクリックして、カラム名を [SQL 文] ウィンドウ枠の現在のカーソル位置に挿入します。

❖ データベース内のプロシージャ名を検索するには、次の手順に従います。

1 [ツール] ー [プロシージャ名のルックアップ] を選択するか、[F8] キーを押します。

2 プロシージャを検索し、選択します。

3 [OK] をクリックして、プロシージャ名を [SQL 文] ウィンドウ枠の現在のカーソル位置に挿入します。

[テーブル名のルックアップ] と [プロシージャ名のルックアップ] の各ダイアログで、検索するテーブルまたはプロシージャの最初の数文字を入力します。フィールドに数文字を入力すると、その文字で始まる項目だけを含むようにリストが限定されます。

SQL のワイルドカード文字 '%' (パーセント記号) と '_' (アンダースコア) を使用すると、検索対象を絞り込むことができます。'%' は、0 文字以上の任意の文字列を表し、'_' は、任意の 1 文字を表します。

たとえば、profile という語を含むすべてのテーブルをリストするには、「**%profile%**」と入力します。

テーブル名に含まれるパーセント記号またはアンダースコアを検索する場合は、パーセント記号またはアンダースコアの前にエスケープ文字を付ける必要があります。エスケープ文字は、使用している JDBC

ドライバによって異なります。jConnect を使用して接続している場合は '\' (円記号)、iAnywhere JDBC ドライバを使用している場合は '~' (チルダ) をそれぞれエスケープ文字として使用します。

SQL 文の印刷

[SQL 文] ウィンドウ枠の内容を印刷するには、[CTRL + P] キーを押すか、[ファイル] メニューで [印刷] を選択します。Interactive SQL の [オプション] ダイアログで、ヘッダまたはフッタの追加、その他のフォーマット・オプションの設定ができます。

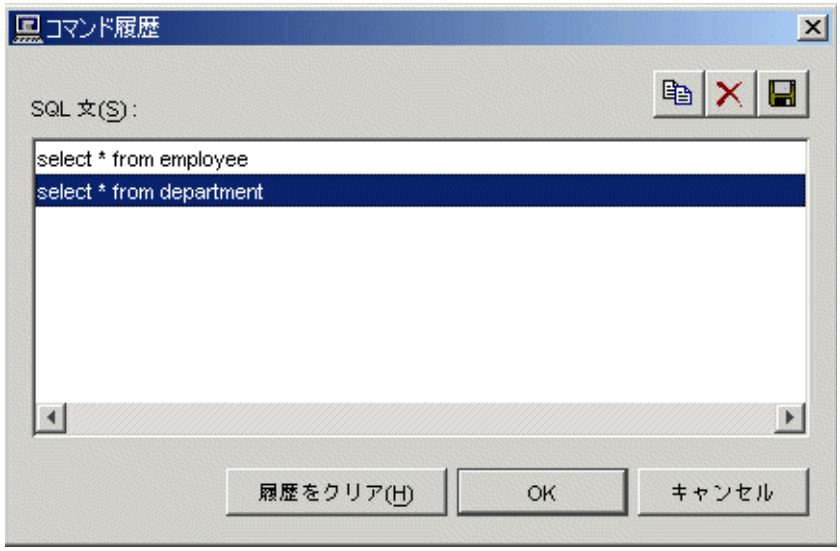
印刷オプションの設定については、『SQL Anywhere Studio ヘルプ』> 「[印刷] タブ」を参照してください。

Interactive SQL でのグラフィカル・プランの印刷については、『ASA SQL ユーザーズ・ガイド』> 「グラフィカルなプラン」を参照してください。

コマンドの再呼び出し

コマンドを実行すると、Interactive SQL は自動的にそのコマンドを履歴リストに保存し、次の Interactive SQL セッションまで保持します。Interactive SQL は、最近使用したコマンドを 50 個まで記録します。

[コマンド履歴] ダイアログで、コマンド・リスト全体を表示できます。[コマンド履歴] ダイアログにアクセスするには、[CTRL + H] キーを押すか、ツールバーのブック・アイコンをクリックします。



最後に使用したコマンドはリストの一番下に表示されます。コマンドを再度呼び出すには、そのコマンドを選択して [OK] をクリックします。コマンドが Interactive SQL の [SQL 文] ウィンドウ枠に表示されます。

また、[コマンド履歴] ダイアログを使用しないでコマンドを再度呼び出すこともできます。ツールバーの矢印を使用してコマンドを前後にスクロールするか、または [ALT + 右矢印] キーと [ALT + 左矢印] キーを押します。

注意
パスワード情報を含む SQL 文 (GRANT CONNECT、GRANT REMOTE DBA、CONNECT、または CREATE EXTERNLOGIN) を実行する場合は、現在の Interactive SQL セッションの間、[コマンド履歴] ダイアログにパスワード情報が表示されます。

コマンド履歴がその後の Interactive SQL セッションで表示された場合は、パスワード情報を含むこれらの文でパスワードが「...」に置換されます。たとえば、Interactive SQL で次の文を実行したと仮定します。

```
GRANT CONNECT TO testuser
  IDENTIFIED BY testpassword
```

この場合は、後続の Interactive SQL セッションで [コマンド履歴] ダイアログに次のように表示されます。

```
GRANT CONNECT TO testuser
  IDENTIFIED BY ...
```

Interactive SQL がセッション間でコマンド履歴を保存する場合は、上にリストした文からパスワード情報が削除されます。

[コマンド履歴] ダイアログからのコマンドのコピー

コマンドをクリップボードにコピーするには、[コマンド履歴] ダイアログでそのコマンドを選択し、[CTRL + C] キーを押すか、ダイアログにフォーカスがあるときに [コピー] ボタンをクリックします。選択したコマンドを [SQL 文] ウィンドウ枠にコピーする場合は、[OK] をクリックします。複数のコマンドをコピーする場合は、それらのコマンドがコマンド・デリミタ (デフォルトではセミコロン) で区切られます。

[コマンド履歴] ダイアログからのコマンドの削除

[コマンド履歴] ダイアログの内容は、Interactive SQL セッション間で保持されます。コマンドは、次の 2 つの方法のいずれかでダイアログから削除できます。

- 1 つまたは複数のコマンドを選択し、[削除] ボタンをクリックするか [DELETE] キーを押して、選択したコマンドをダイアログから削除します。この操作は取り消せません。
- [履歴をクリア] ボタンをクリックすると、すべてのコマンドがダイアログから削除されます。この操作は取り消せません。

コマンドは、別の Interactive SQL セッションで使用できるように、テキスト・ファイルに保存することも可能です。

❖ コマンド履歴をファイルに保存するには、次の手順に従います。

1. [コマンド履歴] ダイアログを開きます。
2. [保存] ボタンをクリックするか、[CTRL + S] キーを押します。
3. [Save As] ダイアログで、ファイルの保存場所と名前を指定します。

コマンド履歴ファイルの拡張子は、*.SQL* にします。

4 設定を終了したら、[保存] をクリックします。

コマンドのロギング

Interactive SQL のロギング機能を使うとコマンドの実行を記録できます。Interactive SQL は、ロギング・プロセスを停止するまでまたは現在のセッションを終了するまで記録を続けます。記録されたコマンドは、ログ・ファイルに保存されます。

❖ Interactive SQL コマンドのロギングを開始するには、次の手順に従います。

1 [SQL] － [ロギングの開始] を選択します。

2 [名前を付けて保存] ダイアログで、ログ・ファイルの保存場所と名前を指定します。

3 設定を終了したら、[保存] をクリックします。

❖ Interactive SQL コマンドのロギングを停止するには、次の手順に従います。

- [SQL] － [ロギングの停止] を選択します。

ヒント

ロギングの開始と停止には、[SQL 文] ウィンドウ枠にコマンドを入力する方法もあります。ロギングを開始するには、「**START LOGGING** *'c:\filename.sql'*」と入力して実行します。*c:¥filename.sql* には、ログ・ファイルのパス、名前、拡張子を指定します。一重引用符が必要なのはパスにスペースが含まれる場合だけです。ロギングを停止するには、「**STOP LOGGING**」と入力して実行します。

ロギングを開始すると、正常に動作しなかったものも含めて実行したコマンドがすべて記録されます。

まとめ

この章では、Interactive SQL の起動方法、SQL 文の実行方法、データベースのデータの表示方法について学習しました。

第 15 章

Sybase Central を使用したデータベースの管理

この章の内容

この章では、Sybase Central という Adaptive Server Anywhere グラフィカル・データベース管理ツールについて説明します。また、Sybase Central を使用してデータベース・プロパティを管理する方法についても説明します。

Sybase Central の起動とサンプル・データベースへの接続、サンプル・データベースの表示と編集、テーブルの編集、ユーザとグループの管理、ストアド・プロシージャの操作、データベースのバックアップなどを行う方法について学習します。

始める前に、必ずサンプル・データベースのコピーを作成して、終了時に元のフォームに戻せるようにしておいてください。

サンプル・データベースのコピーについては、「レッスン 1 ：サンプル・データベースのコピーの作成」254 ページを参照してください。

Sybase Central について

Sybase Central は、Adaptive Server Anywhere データベースのさまざまな設定、プロパティ、ユーティリティをグラフィカル・ユーザ・インタフェースを介して使用するデータベース管理ツールです。Sybase Central は、iAnywhere のその他の製品を管理する場合にも使用できます。この章では、Adaptive Server Anywhere のサンプル・データベースで Sybase Central を使用する方法について説明します。

Sybase Central は、この両方のタスクを実行するための使いやすいインタフェースを提供します。

- SQL 文をサーバに送信することによって実行されるタスク。
- Adaptive Server Anywhere ユーティリティによって実行されるタスク。

レッスン 1 ：Sybase Central の起動と接続

この項では、Sybase Central を起動し、ASA 9.0 Sample データ・ソースを使用してサンプル・データベースに接続する方法について説明します。

❖ Sybase Central を起動するには、次の手順に従います。

- [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [Sybase Central] の順に選択します。

  [Sybase Central] ウィンドウが開きます。

❖ サンプル・データベースに接続するには、次の手順に従います。

1. [ツール] メニューから、[接続] を選択します。
2. プラグインの選択を求められたら、[Adaptive Server Anywhere 9] を選択し、[OK] をクリックします。

   [接続] ダイアログが表示されます。
3. [ODBC データ・ソース名] オプションを選択します。
4. [ODBC データ・ソース名] ボックスに、「**ASA 9.0 Sample**」と入力します。ドロップダウン・リストにこのエントリがあれば、それを選択します。

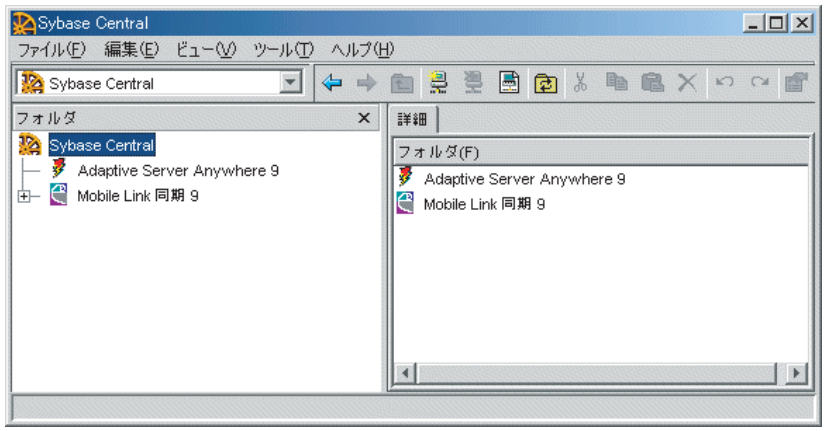
   または、[参照] をクリックし、[データ・ソース名] ダイアログの [ASA 9.0 Sample] を選択して、[OK] をクリックします。
5. [OK] をクリックして、サンプル・データベースに接続します。

レッスン 2 ：Sybase Central のインタフェース

この項では、Sybase Central のユーザ・インタフェースのナビゲーション方法について説明します。

Sybase Central のメイン・ウィンドウ

Sybase Central のメイン・ウィンドウ



ウィンドウ枠

Sybase Central のメイン・ウィンドウは左右に並んだ 2 つのウィンドウ枠に分割されています。

左側のウィンドウ枠には、データベース・オブジェクト、または「**コンテナ**」の階層が、ツリーのような構造で表示されます。コンテナとは、他のコンテナも含めた他のデータベース・オブジェクトを保持できるデータベース・オブジェクトのことです。

右側のウィンドウ枠には、左側のウィンドウ枠で選択されたコンテナの内容が表示されます。両方のウィンドウ枠の表示は、[オプション] ダイアログ ([ツール] メニューからアクセス) で変更できます。

ツールバー

Sybase Central のツールバーには、一般的なコマンドのボタンがあります。ツールバーを表示または非表示にするには、[ビュー] － [ツールバー] を選択します。

メイン・ツールバーを使用すると、次の作業ができます。

- オブジェクト・ツリーのナビゲーション
- データベース、サーバ、製品モジュールへの接続または接続の解除
- [接続プロファイル] ダイアログにアクセス ([ツール] メニューからも可能)
- 現在のフォルダのビューの再表示。[F5] キーを押しても、フォルダを再表示できます。
- オブジェクトのカット、コピー、貼り付け、または削除
- 選択したオブジェクトのプロパティ・シートの表示

各ボタンの上にカーソルを移動するとポップアップ・ヒントが表示され、簡単にボタンの機能を確認できます。

ステータス・バー

ステータス・バーはメイン・ウィンドウの下部に表示されます。メニュー間をナビゲーションすると、このステータス・バーにメニュー・コマンドの簡単な説明が表示されます。ステータス・バーを表示または非表示にするには、[ビュー] － [ステータス・バー] を選択します。

レッスン 3 ：サンプル・データベースの表示

Sybase Central を使用すると、テーブル、カラム、プロシージャの作成、削除、表示、ユーザとユーザ・グループの管理、データベースのバックアップ、データベース・テーブル内のデータの修正など、さまざまなデータベース管理タスクを実行できます。この項を参照すると、サンプル・データベース内のテーブルやその他のオブジェクトにアクセスできます。

データベース・スキーマの表示

データベース「**スキーマ**」とは、データベース内のすべてのオブジェクトの集合のことです。Sybase Central では、データベース・スキーマをコンテナとその内容の階層で表示します。この項では、データベースのスキーマの表示方法について説明します。

データベース・コンテナの展開

Sybase Central では、次の方法でデータベース内のオブジェクトを表示できます。

- 左側のウィンドウ枠で、コンテナをクリックして選択します。選択したコンテナの内容が右側のウィンドウ枠に表示されます。
- 左側のウィンドウ枠で、コンテナの横のプラス記号またはマイナス記号を 1 回クリックすると、コンテナが展開するか、折りたたまれます。コンテナの隣にプラス記号またはマイナス記号が表示されない場合は、そのコンテナには、現在のレベル以下にオブジェクトが含まれている階層はありません。
- 左右どちらかのウィンドウ枠にあるコンテナをダブルクリックします。これによって、左側のウィンドウ枠にあるツリーが展開され、右側のウィンドウ枠にコンテナの内容が表示されます。

左側のウィンドウ枠には、コンテナ・オブジェクトだけが表示されます。右側のウィンドウ枠には、左側のウィンドウ枠で選択されたコンテナ・オブジェクトの内容が表示されます。たとえば、左側のウィンドウ枠で [テーブル] フォルダを選択すると、そのフォルダにあるすべてのテーブルが右側のウィンドウ枠に表示されます。

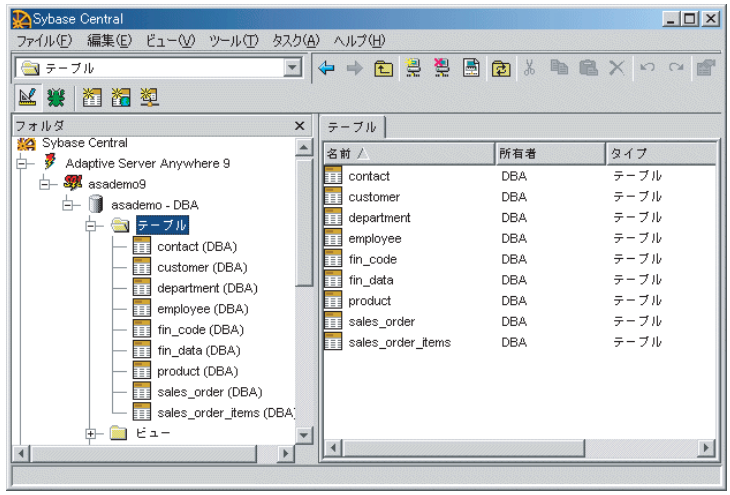
データベース内のテーブルの表示

サンプル・データベースに接続したら、左ウィンドウ枠のフォルダを開けば、データベースを構成するテーブルとその他のオブジェクトを表示できます。

各テーブルの「所有者」が、左ウィンドウ枠にあるテーブル名の横のカッコ内に表示されます。下の図よりも多くのテーブルが表示される場合は、asademo データベースを右クリックし、ポップアップ・メニューから [所有者別にオブジェクトをフィルタ] を選択します。リスト内の [DBA] をオンにして、その他のユーザ ID をオフにします。[OK] をクリックし、表示するオブジェクトを DBA が所有するものだけに制限します。

❖ サンプル・データベース内のテーブルを調べるには、次の手順に従います。

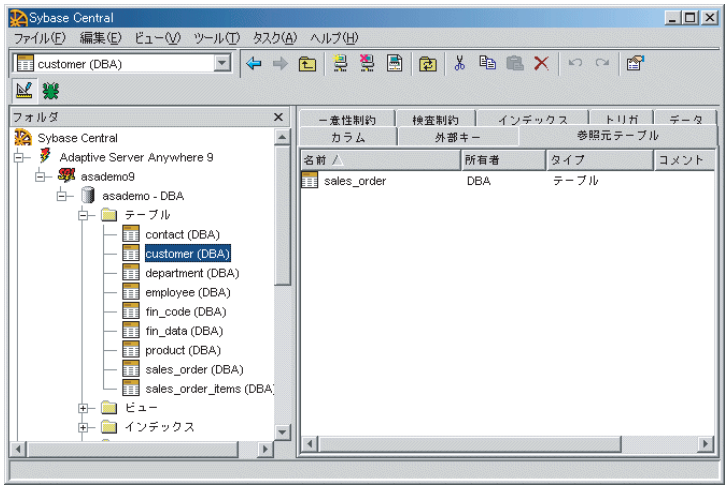
1 [テーブル] フォルダを展開します。最初に、サンプル・データベース・ファイルの asademo を展開しなければならないことがあります。



2 [テーブル] フォルダ内の各テーブルは、それ自体がコンテナです。左側のウィンドウ枠でテーブルを選択します。

3　右側のウィンドウ枠にあるさまざまなタブをクリックして、選択したテーブルに対応するタブの内容を確認します。

データベース・オブジェクトのプロパティの表示



データベースまたはストアド・プロシージャなどのデータベース・オブジェクトのプロパティは、次のいずれかの方法を使用して表示できます。

- 左側のウィンドウ枠内のデータベース・オブジェクトを右クリックして、ポップアップ・メニューで [プロパティ] を選択します。
- データベース・オブジェクトを選択して、[ファイル] ー [プロパティ] を選択します。

項目をクリックまたはダブルクリックすると、データベース内をナビゲーションできます。

データベース・フォルダ

どの Adaptive Server Anywhere データベースにも、次のようなフォルダがあります。

- **テーブル**　データベース内に格納されているベース・テーブルとグローバル・テンポラリ・テーブル。
- **ビュー**　データベース内にクエリとして格納されていて、アクセスされたときに評価される計算テーブル。

- **インデックス**　一部のカラムまたはすべてのカラム内の値に基づいてテーブルのローの順序を指定するデータベース内のインデックス。
- **トリガ**　データを変更するクエリが実行されると、自動的に実行される特別な形式のストアド・プロシージャ。
- **システム・トリガ**　データベース内で参照整合性機能を実装するトリガ。
- **プロシージャとファンクション**　SQL プロシージャで構成されたモジュールベースの言語を使う場合に使用する。
- **イベント**　イベントの作成と編集に使用する。
- **ドメイン**　非標準のデータ型を作成する場合に使用する。
- **ユーザとグループ**　データベースの使用を許可されたユーザを管理する場合に使用する。
- **統合化ログイン**　ユーザが Windows のユーザ名とパスワードを使用してデータベースに接続できるようにする。
- **SQL Remote ユーザ**　データベース内のデータの SQL Remote レプリケーションを管理する場合に使用する。
- **Mobile Link ユーザ**　Mobile Link ユーザと Mobile Link データ同期を管理する場合に使用する。

  パブリケーション　レプリケートするデータを指定するデータベース・オブジェクト。
- **Ultra Light プロジェクト**　Ultra Light アプリケーションで使われる SQL 文とテーブル定義を収集する場合に使用する。
- **DB 領域**　データベースで使う *.db* ファイルを複数作成する場合に使用する。
- **リモート・サーバ**　リモート・サーバを識別するために使用する。識別すると、ローカル・ユーザは、リモート・プロシージャ・コールを実行したり、リモート・サーバから情報を取り出したりできる。

Web サービス　データベース内で現在使用できる Web サービスのリスト。

Sybase Central のメイン・ウィンドウ内でデータベース・オブジェクトを見つけるのに慣れるまで、サンプル・データベースを試してみてください。

テーブル内のカラムの表示

各テーブルの右ウィンドウ枠のタブには、テーブルのカラム、外部キー、インデックス、トリガ、データに関する情報が表示されます。ここでは、サンプル・データベースの product テーブルにあるカラムのみに注目します。

❖ 製品テーブルのカラムに関する情報を表示するには、次の手順に従います。

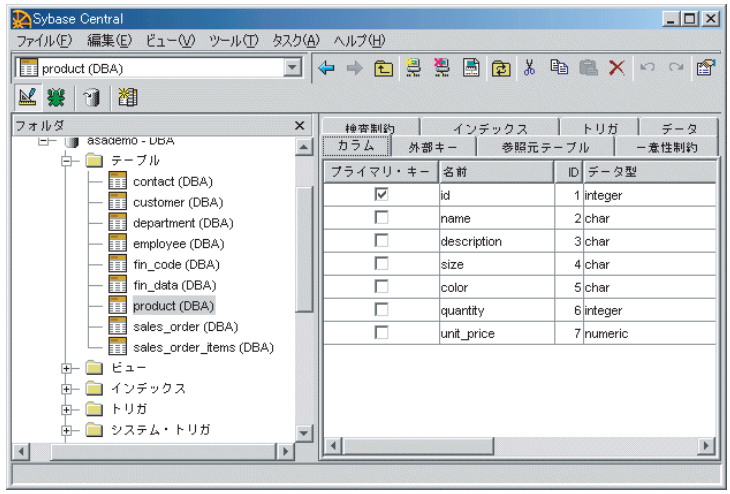
1. product テーブルを開きます。

   左ウィンドウ枠で [テーブル] フォルダを展開してから、左ウィンドウ枠で product テーブルを選択します。

   product テーブルについての情報が、右ウィンドウ枠のタブに表示されます。

2. 右ウィンドウ枠で、[カラム] タブをクリックします。

   各カラムの名前、データ型、その他の情報が右ウィンドウ枠に表示されます。



3　color カラムのプロパティを表示します。

color カラムを選択し、[ファイル] メニューから [プロパティ] を選択します。

color カラムのプロパティ・シートが表示されます。このプロパティ・シートには color カラムの詳細情報が記載されています。プロパティの中には、高度な設定をするときにのみ必要なものがありますが、それらについては、このマニュアルでは取り扱いません。

データベース内のデータの表示

Sybase Central を使用すると、データベース・スキーマを表示するほか、データベース・テーブルにあるデータを表示できます。

❖ データベース・テーブルの内容を表示するには、次の手順に従います。

1　[テーブル] フォルダを展開します。

2　左側のウィンドウ枠で、contact テーブルをクリックします。

3　右側のウィンドウ枠で、[データ] タブをクリックします。

テーブル内のデータが表示されます。

レッスン 4 ：テーブルの作成と編集

このレッスンでは、サンプル・データベースにテーブルを追加する手順を説明します。既存のテーブルへのカラムの追加も行います。

Sybase Central でテーブルを作成する

このレッスンでは、サンプル・データベースに「office」という名前のテーブルを作成します。

❖ office テーブルを作成するには、次の手順に従います。

1. Sybase Central の左側のウィンドウ枠にある [テーブル] フォルダをクリックします。
2. [ファイル] メニューから、[新規] ー [テーブル] を選択します。

   [テーブル作成] ウィザードが表示されます。
3. ウィザードの説明に従って、テーブル名を「**office**」とし、テーブルにプライマリ・キーの制約を割り当てないように選択します。ページの残りの項目については、デフォルト設定をそのまま使用します。
4. 先頭のローを「**office_id**」という名前にし、「smallint」というデータ型を割り当てます。
5. [ファイル] メニューから、[テーブルの保存] を選択します。

Office テーブルの編集

この項では、既存のテーブルの編集方法について説明します。次の手順では、office_id という名前のカラムをプライマリ・キーにして、前の項で作成したテーブルにプライマリ・キーを追加する方法について説明します。

❖ Office テーブルを編集するには、次の手順に従います。

1 [テーブル] フォルダを開いて、office テーブルを選択します。

2 [プライマリ・キー] カラムにチェックマークを付けて、office_id カラムがテーブルのプライマリ・キーであることを示します。

プライマリ・キーの条件
プライマリ・キーの制約カラムに Null を含めることはできません。そのため、[プライマリ・キー] チェックボックスを選択すると、[NULL] チェックボックスのチェックマークがその列から自動的に削除されます。

同様に、[プライマリ・キー] チェックボックスを選択すると、その列からプライマリ・キーの制約が自動的に削除されます。

---

3 [ファイル] メニューから、[テーブルの保存] を選択します。

4 [ファイル] メニューから、[新しいカラム] を選択します。

5 カラム名が office_name、データ型が char、サイズが 20 になるように新しいカラムを編集します。[NULL] を選択して office_name をブランクにできるようにします。

6 [ファイル] メニューから、[テーブルの保存] を選択します。

これでデータベース内にカラムが入りましたが、このカラムには、データは入っていません。

注意

- テーブル、ユーザ、ストアド・プロシージャなど、Sybase Central のすべてのオブジェクトにはプロパティ・シートがあります。
- Sybase Central でオブジェクトを選択すると、そのオブジェクトに関係するコマンドが [ファイル] メニュー、ポップアップ・メニュー、ツールバーのボタンなど、さまざまな場所に表示されます。ポップアップ・メニューには、オブジェクトを右クリックするとアクセスできます。

- フォルダをクリックすると、多くの場合、ツールバーに「*新しい項目*」のボタンが表示されます。これらのボタンをクリックすると、データベースの新規オブジェクトが作成されます。
- コンテナを開くと、パフォーマンスを向上させるためにその内容がキャッシュされます。[ビュー] メニューにある 2 つのリフレッシュ・コマンドを使って、現在のコンテナのビューまたはウィンドウ全体のビューを更新できます。

office テーブルの削除

テーブルをデータベースから削除できます。たとえば、office テーブルを削除して、サンプル・データベースを元の状態に戻せます。

❖ サンプル・データベースから office テーブルを削除するには、次の手順に従います。

1. [テーブル] フォルダを開きます。
2. office テーブルを選択し、[編集] メニューから [削除] を選択します。

   [削除の確認] ダイアログが表示されます。
3. [はい] をクリックして、データベースから office テーブルを削除します。

レッスン 5 ：ユーザとグループの管理

Adaptive Server Anywhere では、ユーザもグループもデータベース内のオブジェクトです。しかし、グループはまたコンテナでもあり、ユーザやその他のグループを含むことができます。ユーザがグループ内に含まれている場合は、そのユーザはグループのメンバです。

この構造では、グループに付与されたパーミッションは、その中に含まれるすべてのユーザとグループに継承されます。Adaptive Server Anywhere では、データベースの使用を許可されたユーザとグループを作成したり、他のグループにメンバシップを付与したりできます。ユーザとグループは複数のグループのメンバになることができます。

このセクションでは、データベースに対してグループを作成する方法、個別のユーザを作成する方法、ユーザをグループのメンバにする方法を説明します。

グループのサンプル・データベースへの追加

この項では、サンプル・データベースにグループを追加する方法を説明します。

❖ グループをサンプル・データベースに追加するには、次の手順に従います。

1. サンプル・データベースに接続します。
2. 左側のウィンドウ枠で、[ユーザとグループ] フォルダを選択します。
3. [ファイル] メニューから、[新規] － [グループ] の順に選択します。

   [グループ作成] ウィザードが表示されます。
4. ウィザードの指示に従います。新しいグループ名は **Sales** とし、他のオプションはデフォルト設定のままにします。

Sales グループが左右のウィンドウ枠に表示されます。

データベースへのユーザの追加

この項では、サンプル・データベースにユーザを追加する方法を説明します。

❖ ユーザをサンプル・データベースに追加するには、次の手順に従います。

1. サンプル・データベースに接続します。
2. 左側のウィンドウ枠で、[ユーザとグループ] フォルダを選択します。
3. [ファイル] メニューから、[新規] － [ユーザ] の順に選択します。
4. [ユーザ作成] ウィザードの指示に従います。新しいユーザ名は **Sandy** とし、パスワードとして **Sandy** を使用します。残りの設定はデフォルト値のままにします。

新しいユーザのアイコンが両方のウィンドウ枠に表示されます。

ユーザのグループへの追加

この項では、コピーと貼り付け機能を使用して 1 つのグループに 2 人のユーザを追加する方法を説明します。

❖ ユーザをグループに追加するには、次の手順に従います。

1. 左側のウィンドウ枠で [ユーザとグループ] フォルダを開きます。
2. Sales グループをクリックします。
3. 右側のウィンドウ枠で、[メンバ] タブをクリックします。
4. [ファイル] メニューから、[新規] － [メンバ] の順に選択します。

   [新しいメンバ] ダイアログが表示されます。

5　ユーザ Sandy を選択し、[OK] をクリックします。

レッスン 6 ：ストアド・プロシージャの表示と編集

[プロシージャとファンクション] フォルダには、データベースのストアド・プロシージャが格納されています。このレッスンでは、プロシージャの内容を表示、変更する方法と、プロシージャのプロパティを変更する方法を説明します。

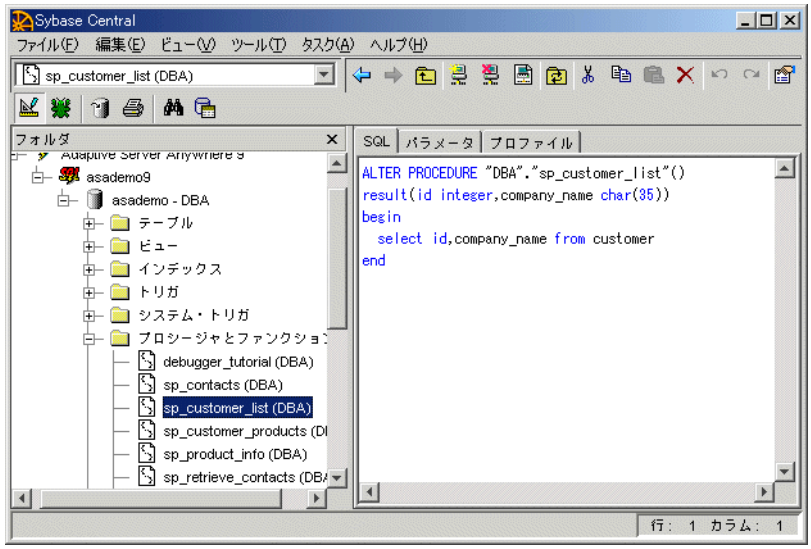
ストアド・プロシージャの表示

サンプル・データベースには、いくつかのストアド・プロシージャがあります。次の手順では、ストアド・プロシージャの表示および編集方法について説明します。

❖ ストアド・プロシージャの内容を表示するには、次の手順に従います。

1. Sybase Central の左側のウィンドウ枠で、サンプル・データベースの [プロシージャとファンクション] フォルダを開きます。
2. sp_customer_list ストアド・プロシージャをクリックします。

   sp_customer_list というプロシージャのテキストが、右側のウィンドウ枠にある [SQL] タブに表示されます。



このプロシージャは、customer テーブルから顧客 ID と会社名のセットを返すように設計されています。

ストアド・プロシージャ・プロパティの編集

ストアド・プロシージャのプロパティを編集するには、そのプロパティ・シートを使用します。この項では、ストアド・プロシージャのパーミッションを変更します。変更後、前のレッスンで作成したユーザ (Sandy) など、すべての営業員が、顧客リストを入手するプロシージャを実行できます。

ストアド・プロシージャには、それに関連したパーミッションがあります。プロシージャを実行するには、そのプロシージャを実行するためのパーミッションが付与されているか、またはそのプロシージャを実行するためのパーミッションを持つユーザ・グループのメンバである必要があります。

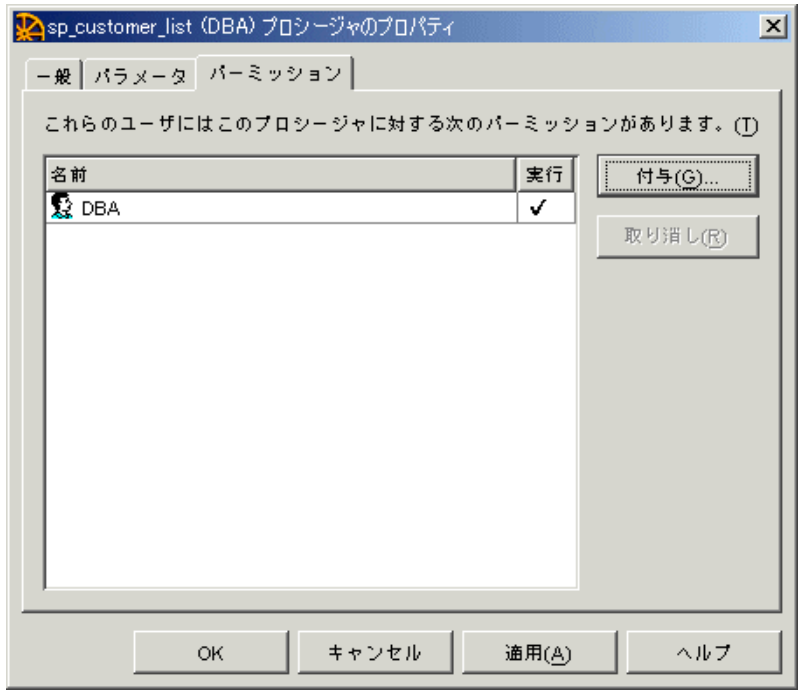
❖ sp_customer_list プロシージャのパーミッションを表示して変更するには、次の手順に従います。

1. [プロシージャとファンクション] フォルダで、sp_customer_list を選択します。
2. [ファイル] － [プロパティ] を選択します。
3. プロパティ・シートの [パーミッション] タブをクリックして、このプロシージャに対するパーミッションが付与されたユーザ ID を表示します。現在、サンプル・データベースのユーザは DBA だけで、他にパーミッションを付与されたユーザはいません。DBA はこのプロシージャの所有者であり、自動的にプロシージャの実行パーミッションが付与されています。
4. [付与] をクリックします。

   [パーミッションの付与] ダイアログが表示されます。

5 [パーミッションの付与] ダイアログで、Sales を選択して [OK] をクリックし、このグループに sp_customer_list プロシージャを実行するパーミッションを付与します。

[プロシージャ] プロパティ・シートの [実行] カラムにチェックマークが入った状態で、[パーミッション] タブに Sales が追加されます。



このチェックマークをはずすか、グループを選択して [取り消し] をクリックすると、グループに付与されたパーミッションを破棄できます。

6　[OK] をクリックして、sp_customer_list パーミッションに加えた変更を受け入れます。

レッスン 7 ：データベースのバックアップ

Sybase Central には、一般的なデータベース管理タスクを実行するための、一連のデータベース・ユーティリティが含まれています。ウィザードを実行することで、必要な操作を順を追って実行できます。

ユーティリティ

実行中のデータベースで使用できるユーティリティのリストを表示するには、サンプル・データベース・コンテナを選択し、[ファイル] メニューをクリックします。[データベースのバックアップ]、[データベースのアップグレード]、[データベースの検証] など、複数のオプションが表示されます。これらの各メニュー項目は、ユーティリティに対応しています。

この項では、サンプル・データベースのバックアップにウィザードを使用します。これは、データベースを実行したまま行えます。

❖ 実行中のデータベースをバックアップするには、次の手順に従います。

1. 左側のウィンドウ枠で、[asademo - DBA] アイコンを右クリックして、ポップアップ・メニューで [データベースのバックアップ] を選択します。[データベース・バックアップ] ウィザードが表示されます。
2. この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。
3. バックアップできるデータベースのリストから、asademo を選択します。[次へ] をクリックします。
4. [ディスク上の次のファイル] を選択します。

   データベースのバックアップ先のファイル名をテキスト・ボックスに入力します。これはチュートリアルなので、*c:¥temp¥backup* などのテンポラリ・ディレクトリにあるファイルを選択してください。
5. [完了] をクリックし、データベースをバックアップします。

注意

この種類のバックアップは「**アーカイブ・バックアップ**」と呼ばれます。イメージ・バックアップというバックアップも作成できます。**イメージ・バックアップ**は、データベース・ファイルとそれに関連付けられたファイルの物理的なコピーです。

[データベース・バックアップ] ウィザードで指定するファイル名には、拡張子 *.1* が追加されます。

ウィザードで実行できる管理タスクは、ほかにもいくつかあります。左側のウィンドウ枠でサーバを選択し、[ファイル] メニューから [データベースの作成] を選択して、データベースを作成することもできます。

サンプル・データベースのリストア

このチュートリアルは終了したので、サンプル・データベースを元に戻して、元のフォームでまた使用できるようにします。変更を加えたバージョンの *asademo.db* を、チュートリアルを始める前に作成したバックアップ・コピーに置き換えます。*asademo.log* は削除してください。

まとめ

この章では、Sybase Central を起動して、サンプル・データベースに接続する方法を学習しました。また、サンプル・データベースの表示と編集、テーブルの編集、ユーザとグループの管理、ストアド・プロシージャの操作、データベースのバックアップについても学習しました。

第 16 章

データベースの設計と構築

この章の内容

この章では、データベース設計の原理をいくつか紹介し、Sybase Central を使ってデータベースを作成する方法について説明します。Adaptive Server Anywhere のサンプル・データベースを使用して、データベースの仕組みと働きについて説明します。また、サンプル・データベースを使用して、前の各章で習得したスキルをデータベースの新規作成に応用します。サンプル・データベースの product、sales_order_items、sales_order、customer の各テーブルを使用し、モデル化した非常に簡単なデータベースを設計し、構築します。

新しいデータベースの設計について

ごく単純なデータベースを設計する場合でも、作業を開始する前に時間を取り、設定するテーブル、テーブルに関連付けるキーなどを決めておくことが重要です。

小中規模のデータベースでは、設計はそれほど複雑ではありませんが、それでも非常に重要な作業です。設計が良くないと、データベース・システムは非効率的で信頼性の低いものになってしまいます。データベース・アプリケーションはデータベースの特定の部分に対して処理を行うように構築されており、データベースの設計に依存します。このため、設計のよくないデータベースをあとから修正するのは困難です。

大規模データベースの設計は複雑な作業です。正式な手順としては、効率的な設計が可能な概念データ・モデリングなどがあります。Sybase PowerDesigner や DataArchitect などの強力なツールを使用すると、これらの技術を設計に適用し、大規模なデータベース設計を管理できます。

この章では、大規模データベースを設計する際の問題については取り扱いません。この章では、1 つのテーブルにグループ分けする情報の種類と、テーブル間の関係を検討して、分類する方法を決定するために考慮すべき事項を説明します。

データベース設計の基本原理については、『ASA SQL ユーザーズ・ガイド』> 「データベースの設計」を参照してください。高度な利用法については、Sybase PowerDesigner のマニュアルまたはデータベース設計の専門書をお読みください。

単純なデータベースの設計と構築

データベースを設計するときは、どの項目についての情報を格納するか、それぞれの項目にどの情報を保持するかを計画します。また、これらの項目がどのように関連しているかを調べます。データベース内のものを各エンティティとして分類しエンティティ間のリンクを「**関係**」と呼びます。

レッスン 1 ：データベース・ファイルの作成

このレッスンでは、データベースを格納するデータベース・ファイルを作成します。

詳細については、「データベース・システムの各部分」164 ページを参照してください。

概念

データベース・ファイルがコンテナであり、データベースの格納場所となります。データベース・ファイルには、システム・テーブルと、すべてのデータベースに共通するその他のシステム・オブジェクトが入っていますがそこに格納するテーブルとデータを追加する必要があります。

データベース内のテーブル、インデックスなどの集合とそれらすべての関係をデータベースの「**スキーマ**」と呼びます。スキーマは、まだデータが入っていないデータベースです。このチュートリアルでは、非常に簡単なスキーマを設定し、作成する方法について説明します。

各部分の基本概念については、「リレーショナル・データベースの概念」158 ページを参照してください。

テーブル、カラム、インデックスを含むデータベース内の各オブジェクトの名前は「**識別子**」として使われます。何を識別子として使用できるかを定める規則があります。英字、数字、記号を自由に組み合わせて使用できます。ただし、カラム名に英字、数字、アンダースコア以外のものが含まれていたり、カラム名が英字で始まっていなかったり、それがキーワードと同じであったりする場合、カラム名を二重引用符 (" ") で囲みます。

QUOTED_IDENTIFIER データベース・オプションが OFF に設定されている場合には、SQL 文字列を二重引用符で区切る必要があります。この SQL 文字列は識別子としては使用できません。ただし角カッコは、QUOTED_IDENTIFIER の設定に関わらず、どのような場合でも識別子の区切りとして使用できます。

識別子の詳細については、『ASA SQL リファレンス・マニュアル』>「識別子」を参照してください。

演習

❖ **新しいデータベース・ファイルを作成するには、次の手順に従います。**

1 Sybase Central を起動します。

2 左ウィンドウ枠で Adaptive Server Anywhere プラグインを選択してから、右ウィンドウ枠で [ユーティリティ] タブをクリックします。

3 右ウィンドウ枠で、[データベースの作成] をダブルクリックします。

[データベースの作成] ウィザードが開きます。

4 概要ページに表示される情報を読んで、[次へ] をクリックします。

5 [このコンピュータにデータベースを作成] を選択し、[次へ] をクリックします。

6 データベース・ファイルを格納する場所と名前を選択します。

ファイル名「*c:¥temp¥mysample*」を入力します。テンポラリ・ディレクトリとして *c:¥temp* 以外のディレクトリを使用する場合は、正しいパスを指定します。

7 [完了] をクリックするとデータベースを作成します。

データベース作成時に指定できるその他のオプションは、[完了] ではなく [次へ] をクリックすると表示できますが、通常はデフォルトの設定で問題ありません。

8 [データベースの作成中] ウィンドウに、タスクの進行状況が表示されます。ファイルの作成後に、[OK] をクリックしてウィンドウを閉じます。

レッスン 2 ：データベースへの接続

このレッスンでは、作成したデータベース・ファイルに接続します。

詳細については、「各部分を組み合わせる方法」167 ページを参照してください。

演習

データベースを作成したら、次にそのデータベースに接続してテーブルやその他のデータベース・オブジェクトを作成できるようにします。

❖ データベースに接続するには、次の手順に従います。

1. Sybase Central を起動します。
2. [ツール] メニューから [接続] を選択して、[接続] ダイアログを開きます。

   複数のプラグインがロードされている場合は、Adaptive Server Anywhere プラグインを選択してから、[接続] ダイアログを開きます。
3. ユーザ ID とパスワードを指定します。

   [ID] タブで、ユーザ ID を「**DBA**」、パスワードを「**SQL**」と入力します。これらは、新しいデータベースのデフォルト値です。

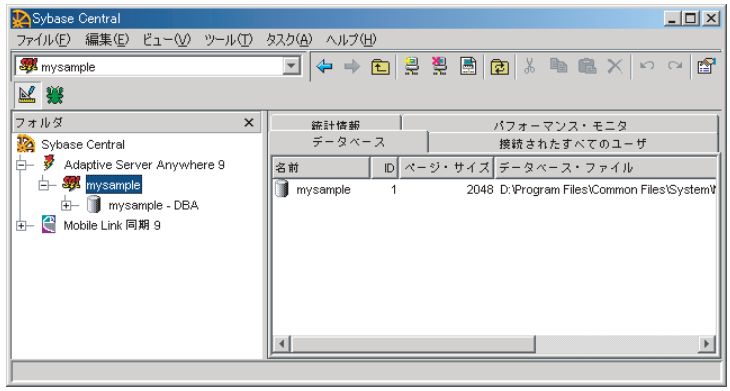
   タブの一番下にある [プロファイル] オプションで [**なし**] を選択します。
4. 使用するデータベース・ファイルを指定します。

   [データベース] タブをクリックします。[データベース・ファイル] フィールドに使用するデータベース・ファイルのフル・パスを入力します。たとえば、前のレッスンで指示のとおりにデータベース・ファイルを作成した場合は、次のように入力します。

   ```
   c:¥temp¥mysample.db
   ```

5　データベースに接続します。

[OK] をクリックします。Sybase Central がデータベースに接続します。



6　左ウィンドウ枠でデータベース・サーバ・コンテナを開き、mysample データベースを表示します。

レッスン 3 ：テーブルの設計と作成

このレッスンでは、データベースにテーブルを追加します。

概念

データベースの各テーブルには特定のサブジェクトに関する情報が含まれている必要があります。データベースの設計用語では、各テーブルの「**エンティティ**」を決定すると呼びます。たとえば、サンプル・データベースの 1 つのテーブルには従業員の情報、別のテーブルには製品に関する情報が保存されていますが、この「従業員」と「製品」をデータベース設計でエンティティと呼びます。

テーブルの概要については、「データベース・テーブル」158 ページを参照してください。

テーブルの各カラムには、情報を格納する対象のエンティティの個々の特性が記述されています。たとえば、サンプル・データベースの従業員テーブルには、従業員 ID 番号、姓と名、住所、その他の各従業員に関する特定の情報を格納するカラムがあります。

データベース図では、各テーブルは長方形のボックスで表わされます。ボックスの一番上にあるのがテーブル名で、ボックス内にリストされているのがカラム名です。

Product	
ID	Integer
name	char(15)
description	char(30)
size	char(18)
color	char(6)
quantity	integer
unit_price	numeric(15,12)

サンプル・データベースから取った上記の product テーブルでは、各製品がスポーツ衣料品目となります。

演習

識別子 (id) と name カラムのみを含む非常に簡単な product テーブルを作成します。

❖ product テーブルを作成するには、次の手順に従います。

1. Sybase Central で、まだ接続していない場合はデータベースに接続します。
2. データベース内の [テーブル] フォルダを開きます。

   最初に [サーバ] と [データベース] コンテナを開き、次に [テーブル] フォルダを開きます。
3. [ファイル] メニューから、[新規] － [テーブル] の順に選択します。

   [テーブル作成] ウィザードが表示されます。
4. テーブルに **product** という名前を付けます。
5. [完了] をクリックします。
6. 右側のウィンドウ枠に、テーブルの最初のカラムが自動的に表示されます。テーブルの最初のカラムを次の値で定義します。
   - **プライマリ・キー**　カラム名の横にあるチェックボックスをオンにして、カラムが**プライマリ・キー**であることを示します。
   - **名前**　カラムに **id** 名を指定します。
   - **データ型**　カラムを **integer** データ型と指定します。
7. 追加のカラムを作成します。

   [ファイル] メニューから [新しいカラム] を選択し、次のプロパティを持つカラムを追加します。
   - **名前**　カラム名を **name** と指定します。このカラムには製品名が入ります。
   - **データ型**　カラムを **char** データ型と指定します。これは文字列を格納します。

- **サイズ** [サイズ] カラムに最大長を「**15**」と入力します。

8 [ファイル] メニューから、[テーブルの保存] を選択します。

これで、データベースにテーブルが作成されました。テーブル・データはデータベース・ファイルに格納されます。この段階ではテーブルは空です。

次の 2 つのレッスンでは、カラムとデータ型についてもう少し詳しく説明します。

レッスン 4 ：プライマリ・キーの識別と作成

このレッスンでは、テーブルのプライマリ・キーを設定する方法についてさらに詳しく説明します。このレッスンには演習はありません。

詳細については、「テーブルにはプライマリ・キーがある」160 ページを参照してください。

概念

「**プライマリ・キー**」とは、テーブル内の 1 つのローをユニークに識別する 1 つまたは複数の特殊なカラムです。product テーブルでは、id カラムで各製品をユニークに識別します。

Product	
ID	Integer
name	char(15)
description	char(30)
size	char(18)
color	char(6)
quantity	integer
unit_price	numeric(15,12)

各ローの id カラムにはユニークな値が入り、各ローの値は id 値によって、1 つの製品識別子だけに関連付けられています。つまり、2 つの製品が同じ名前または同じサイズだったとしても、id 番号は必ず異なります。図では id カラムに下線が引かれプライマリ・キーであることがわかります。

わざわざ識別子のみを格納するためのカラムを作成するのは、データベース設計では普通のことです。したがって、銀行、公共料金、クレジット・カードの明細書でも、必ずユニークな ID 番号が使用されています。

AUTOINCREMENT プライマリ・キーを使用する

プライマリ・キーの入力を簡単にするには、プライマリ・キーのカラムに AUTOINCREMENT のデフォルト値を代入します。このカラムの値は新しいローが追加されるたびに直前の行のフィールド値に 1 を足した値が自動的に入力されます。

❖ AUTOINCREMENT プライマリ・キーを作成するには、次の手順に従います。

1 左側のウィンドウ枠で product テーブルを選択し、右側のウィンドウ枠で [カラム] タブを選択します。

2 プライマリ・キー・カラムを選択します。[ファイル] メニューから [プロパティ] を選択して、そのカラムのプロパティ・シートを開きます。

3 [値] タブをクリックします。

4 [デフォルト値] オプションを選択します。

5 [システム定義] をクリックし、ドロップダウン・リストから [autoincriment] を選択します。

6 [OK] をクリックして、カラムのプロパティ・シートを閉じます。

7 [ファイル] メニューから、[テーブルの保存] を選択します。

レッスン 5 ：カラムのプロパティ設計

このレッスンでは、テーブルのカラムのデータ型とその他の属性を設定する方法について詳しく説明します。

概念

各カラムにはデータ型が関連付けられています。「**データ型**」により、カラムに入る情報の種類が定義されます。そのカラムのデータに適したデータ型をカラムに選択します。たとえば、識別子カラムは通常は整数データ型、名前や住所を入力するカラムは文字データ型にします。

データ型は以下のカテゴリに分類されています。

- **数値データ型**　数値データ型にはいくつかの種類があり、正確なもの (処理中に丸め誤差が発生しないもの) と近似のものに分けられます。

  カラムのデータ型により、そのカラムの最大サイズが決まります。たとえば、SMALLINT を指定した場合、カラムには最大 32,767 までの値を入力できます。INTEGER を指定すると、最大 2,147,483,647 までの値を入力できます。

  完全なリストについては、『ASA SQL リファレンス・マニュアル』> 「数値データ型」を参照してください。

- **文字データ型**　名前や住所などのテキスト文字列を格納するのに使用する型です。それぞれの文字データ型で、格納できる文字列の最大長が決まっています。

  リストについては、『ASA SQL リファレンス・マニュアル』> 「文字データ型」を参照してください。

- **バイナリ・データ型**　この型は、アプリケーションで必要となるバイナリ・フォーマットにエンコードされた情報を格納するのに使用します。

  リストについては、『ASA SQL リファレンス・マニュアル』> 「バイナリ・データ型」を参照してください。

- **日付／時刻データ型**　日付と時刻を格納するのに使用します。

リストについては、『ASA SQL リファレンス・マニュアル』> 「日付と時刻データ型」を参照してください。

- **Long データ型** blob (バイナリ・ラージ・オブジェクト) とも呼ばれます。この型は、長いテキスト文字列 (データベースによっては、メモ・フィールドと呼ばれる)、画像、その他のバイナリ情報を格納するのに使用します。

  詳細については、『ASA SQL リファレンス・マニュアル』> 「LONG BINARY データ型 [バイナリ]」と 『ASA SQL リファレンス・マニュアル』> 「LONG VARCHAR データ型 [文字]」を参照してください。

このほか、Adaptive Server Anywhere はユーザ定義のデータ型と特殊な Java データ型をサポートしています。これらの型については、この入門書では取り扱いません。

NULL と非 NULL

このカラム値が各ローに対して必須の場合は、そのカラムを非 NULL に定義します。カラムには NULL 値が許可されるので、値を指定しないこともできます。デフォルトは NULL 値を許容しますが、NULL 値を許可する明確な理由がないかぎり、カラムには非 NULL を明示的に宣言してください。

NULL 値の詳細については、『ASA SQL リファレンス・マニュアル』> 「NULL 値」を参照してください。NULL 値を比較で使用する方法については、『ASA SQL リファレンス・マニュアル』> 「探索条件」を参照してください。

❖ カラムのデータ型を指定するには、次の手順に従います。

1. 左側のウィンドウ枠で product テーブルを選択し、右側のウィンドウ枠で [カラム] タブを選択します。
2. ID カラムを選択します。
3. [ファイル] メニューから [プロパティ] を選択して、そのカラムのプロパティ・シートを開きます。

   カラムのプロパティ・シートが開きます。

4　[データ型] タブで、[組み込みタイプ] ドロップダウン・リストから Integer を選択します。

5　[OK] をクリックします。

演習

このレッスンと 1 つ前のレッスンでは、データベース・テーブルを作成する際に知っておく必要のある基本概念について紹介しました。これらの知識を生かして、データベースにさらにいくつかのテーブルを追加してみましょう。これらのテーブルは、この章の後のレッスンでも使用します。

次のテーブルをデータベースに追加します。

- **customer**　テーブル customer を追加し、テーブル内に以下のカラムを作成します。
  - **id**　各顧客の ID 番号です。このカラムは **integer** データ型であり、**プライマリ・キー**です。これをオートインクリメント・キーにします。
  - **company_name**　会社名です。このカラムは **character** データ型であり、最大長は **35** 文字です。
- **sales_order**　テーブル sales_order を追加し、テーブル内には以下のカラムを作成します。
  - **id**　各注文書の ID 番号です。このカラムは **integer** データ型であり、**プライマリ・キー**です。これをオートインクリメント・キーにします。
  - **order_date**　注文日です。このカラムは **date** データ型です。
  - **cust_id**　発注した顧客の ID 番号です。このカラムは **integer** データ型です。
- **sales_order_items**　ライン項目情報を格納するテーブル sales_order_items を追加し、テーブル内には以下のカラムを作成します。
  - **id**　ライン項目を含む注文書の ID 番号です。このカラムは **integer** データ型であり、**プライマリ・キー** カラムとして設定します。

- **line_id**　各注文書の ID 番号です。このカラムは **integer** データ型であり、**プライマリ・キー** カラムとして設定します。
- **prod_id**　受注製品の ID 番号です。このカラムは **integer** データ型です。

これで、データベースに 4 つのテーブルが作成されました。ただし、まだテーブル間の関連付けはされていません。次のレッスンでは、テーブルを互いに関連付ける外部キーを定義します。

レッスン 6 ：テーブル間の関係の設計と作成

このレッスンでは、外部キーを使用して、テーブル間の関係を設計、作成する方法について学習します。

詳細については、「テーブルは外部キーによって関連付けられる」161 ページを参照してください。

概念

各テーブルには単一のサブジェクトに関する情報が格納されていますが、複数のテーブルに関連情報が含まれる場合があります。たとえば、「従業員」は「部門」の構成メンバであり、「注文」は一連の「製品」で構成されています。データベース内の関係はテーブル間の「**外部キー**」関係として表示されることも、それ自体が独立したテーブルとして表示されることもあります。この章ではそれぞれの例を紹介します。

データベースに、テーブル内のデータを管理するための規則や決まりをエンコードする関係を作成します。一度データベースの構造に関係が組み込まれると、例外はあり得ません。

テーブル間の関係は、次のように分類されます。

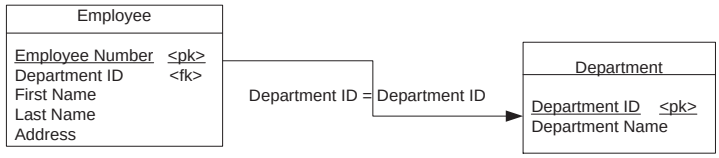
- **1 対 1 の関係**　1 つのエンティティの各項目は、別のテーブルのゼロまたは 1 つのエンティティと対応します。たとえば、このサンプル・データベースで *1* つの部には *1* 人の管理者がいます。部長をもう 1 人入力することはできません。部署のエントリの重複は部署 ID の重複にも関係しますが、部署 ID はプライマリ・キーであるため重複は許されません。

  通常は、1 対 1 の関係にある項目は 1 つのテーブルにまとめた方が便利です。つまり、別の manager テーブルを作成するよりも department テーブルにマネージャ・カラムを作成する方が効率的です。

  項目を分けた方がいい場合については、『ASA SQL ユーザーズ・ガイド』> 「データベースの設計」を参照してください。

- **多対 1 の関係**　多対 1 の関係は、テーブル間の外部キー関係になります。多対 1 の関係では、*1* つのエンティティのプライマリ・キーが*複数*のテーブルの新しい外部キーのカラムとして表示されます。

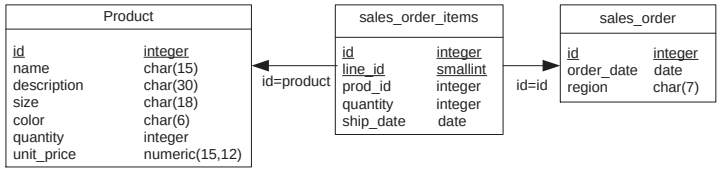
たとえば、演習で作成したデータベースでは、*1* 人の顧客は*複数*の注文を出せますが、各注文を出した顧客は 1 人だけです。1 対多の関係を表すには、customer テーブル (id) のプライマリ・キー・カラムにマッピングする sales_order テーブル (cust_id) に「**外部キー・カラム**」が必要です。多くの場合、この 2 つのカラムには同じ名前を付けると便利です。



sales_order テーブルの cust_id カラムにある各エントリは、customer テーブルの id カラムにあるエントリの 1 つと一致する必要があります。sales_order テーブル (関係に外部キーを持つ) は外部テーブルまたは参照元テーブルと呼ばれます。customer テーブル (参照先のプライマリ・キーを持つ) は、プライマリ・テーブルまたは参照先テーブルと呼ばれます。

- **多対多の関係**　多対多の関係を表すには中間テーブルを使用します。中間テーブルから各関連エンティティへは外部キーの関係となります。

  たとえば、同じデータベースで、製品と注文との間は多対多の関係です。1 つの注文で複数の製品を購入する場合があり、1 つの製品が複数の注文に入っている場合もあります。



場合によっては、中間テーブル (sales_order_items) に、注文のあった製品の品目数、発送日など、その他の情報が含まれる場合があります。この例の場合は、中間テーブルにはその他の情報は含まれません。

演習

データベースのテーブルに関連付けるための外部キーを追加します。

❖ 外部キーを作成するには、次の手順に従います。

1. 外部キーを作成するテーブルを選択します。
2. 右側のウィンドウ枠で、[外部キー] タブを選択します。
3. [ファイル] メニューから、[新規] － [外部キー] を選択して、[外部キー作成] ウィザードを開きます。
4. ウィザードの説明に従って、次の外部キーを追加します。
   - sales_order_items の id カラムから sales_order の id カラムを参照する外部キー。このキーは、注文と注文品目間の多対 1 の関係をデータベースに構築します。
5. 1 ～ 4 の手順を繰り返して、次のキーを作成します。
   - sales_order_items の prod_id カラムから product の id カラムを参照する外部キー。このキーは、注文品目と製品間の多対 1 の関係をデータベースに構築します。
   - sales_order の cust_id カラムから customer の id カラムを参照する外部キー。このキーは、注文と顧客間の多対 1 の関係をデータベースに構築します。

最初の 2 つの外部キーを一緒に使用すると、注文と製品の間の多対多の関係をデータベースに構築できます。

これで、リレーショナル・データベースの設計と構築に関する一般的な説明を終わります。このマニュアルの残りの章では、データベースにデータを追加し、データベースからデータを取得する方法について説明します。以下の章では、演習で作成したものよりも大きい Adaptive Server Anywhere データベースをサンプルとして使用します。

まとめ

この章では、データベース設計の原理について学習しました。また、Sybase Central を使用して、その設計原理を新しいデータベースの作成に応用しました。

第 17 章

データベースと Mobile Link の同期

この章の内容

この章はチュートリアルです。手順に従って操作することにより、Mobile Link を使って、2 つの Adaptive Server Anywhere データベース間で同期をとることができます。データベースの 1 つは統合データベースで、もう 1 つはリモート・データベースです。このチュートリアルでは、この 2 つのデータベースを作成してから同期します。

Mobile Link について

データの「レプリケーション」とは、物理的にまったく異なるデータベース間でデータを共有することです。Sybase ではデータのレプリケーション用に次の 3 つの性質の異なるテクノロジを提供しています。

- Mobile Link
- SQL Remote
- Replication Server

Mobile Link 同期を使用すると、ODBC 対応統合データベースと Adaptive Server Anywhere または Ultra Light リモート・データベース間のレプリケーションを容易に行えます。このチュートリアルでは、Adaptive Server Anywhere リモート・データベースを使用します。統合データベースは、Sybase Adaptive Server Anywhere、Sybase Adaptive Server Enterprise、Oracle、Microsoft SQL Server、または IBM DB2 で作成できます。

Mobile Link では、統合データ・サーバと多数のリモート・データベース (通常、多くのモバイル・データベースを含む) との間の同期が可能です。リモート・サイトで必要な管理とリソースは最小限で済みます。システムは接続ベースなので、リモート・サイトは必要なだけ何回でも接続できます。各接続の終了時には、双方のデータベースが完全に同期化されています。

Mobile Link は、リモート・データベース上の複数のトランザクションの結果を 1 セットの変更にまとめて、その変更を統合データベースに適用します。同期は常にトランザクション境界で発生するので、参照の整合性は保たれます。コンポーネントのトランザクション中に発生した個々の変更の順序は保持されません。コミットされなかったデータは決してレプリケートされないので、データの整合性が損なわれません。

同期方法の詳細については、『Mobile Link 管理ガイド』> 「Mobile Link 管理ガイド」を参照してください。Mobile Link の完全なマニュアルも含まれています。

SQL Remote 同期の概要については、「SQL Remote を使用したデータのレプリケーション」367 ページを参照してください。

概要

このチュートリアルでは、統合データベースとリモート・データベースを作成します。まず、同期パブリケーションとサブスクリプションを作成します。その後、Mobile Link 同期テクノロジを使用して、これらのデータベースを同期させます。

稼働条件

このチュートリアルの稼働条件を次に示します。

稼働条件	説明
所要時間	チュートリアルは、およそ 50 分で終了します。
ソフトウェア	• Adaptive Server Anywhere 9 のフル・インストール環境 • Mobile Link 同期サーバのフル・インストール環境
能力と経験	コマンド・ライン処理の知識と経験、またはそのいずれか ODBC と Sybase Central を使用してデータベースに接続する能力 ODBC の詳細については、「レッスン 1 ：ODBC データ・ソースの作成」264 ページを参照してください。

目的

このチュートリアルの目的は、次の項目に関する知識を習得し、活用することです。

- 統合システムとしての Mobile Link 同期サーバとクライアント
- Mobile Link 同期サーバとクライアントに関するコマンドの実行
- Mobile Link 同期サーバとクライアントのコマンド・ラインとスイッチ

主要な概念

Mobile Link 同期サーバは、ODBC インタフェースを使って統合データベースに接続します。Mobile Link 同期クライアントは、リモート・データベースに接続します。Mobile Link 同期サーバとクライアントは組になって機能し、1 つのデータベースから別のデータベースへのデータのアップロードとダウンロードを管理します。

このチュートリアルでは、次の重要な概念について学習します。

- Mobile Link 同期サーバ、Mobile Link 同期クライアント
- ODBC 接続、ODBC データ・ソース、同期サブスクリプションとパブリケーション、統合サーバ、リモート・データベース

レッスン 1 ：データベースの作成

Mobile Link 同期を実行するには、統合データベースとリモート・データベースに互換性があること、データベース・テーブル内にデータがあること、各データベース用の ODBC データ・ソースがあることが必要です。

データベース・ファイルの作成

最初の手順として、データベース・ファイルを作成します。この手順では、コマンド・ラインから *dbinit* 実行プログラムを使用して、統合データベースとリモート・データベースを構築します。

ヒント
dbinit を使用してデータベース・ファイルを作成することは、ディスクのフォーマットに似ています。これによって、ユーザ・テーブルもプロシージャもないデータベース・ファイルが作成されます。データベース・スキーマは、新しく初期化されたファイルの中にさまざまなユーザ定義テーブルとプロシージャを定義するときに作成します。

---

dbinit ユーティリティの詳細については、『ASA データベース管理ガイド』> 「dbinit コマンド・ライン・ユーティリティを使用したデータベースの作成」を参照してください。

❖ データベースを作成するには、次の手順に従います。

1. チュートリアルの全ファイルを格納するディレクトリを作成します。

   C:¥ ドライブから、コマンド・プロンプトで次のように入力します。

   ```
   mkdir MLTutorial
   ```

2. 作成したディレクトリに移動します。

   コマンド・プロンプトで次のように入力します。

   ```
   cd MLTutorial
   ```

3. 統合データベースのファイルを作成します。

コマンド・プロンプトで次のコマンドを入力します。

```
dbinit consol.db
```

4　リモート・データベースのファイルを作成します。

コマンド・プロンプトで次のコマンドを入力します。

```
dbinit remote.db
```

5　これらのデータベース・ファイルの作成が成功したことを確認するために、コマンド・プロンプトで次のコマンドを入力します。

```
dir
```

作成したデータベース・ファイルが、ディレクトリ・リストの中に表示されます。表示されない場合は手順を見直して、必要に応じて手順 1 または 2 を繰り返してください。

ODBC データ・ソースの作成

これで ODBC データ・ソースを構築する準備が整いました。この ODBC データ・ソースから、Adaptive Server Anywhere データベースに接続できます。

ODBC データ・ソースの作成については、『ASA データベース管理ガイド』> 「データ・ソース・ユーティリティ」を参照してください。

❖ ODBC データ・ソースを作成するには、次の手順に従います。

1　統合データベース用の ODBC データ・ソースを作成するには、コマンド・プロンプトで次のように入力します。

```
dbdsn -w test_consol -y -c
"uid=DBA;pwd=SQL;dbf=C:¥MLTutorial¥consol.db;eng=Co
nsol"
```

2　リモート・データベース用の ODBC データ・ソースを作成するには、コマンド・プロンプトで次のように入力します。

```
dbdsn -w test_remote -y -c
"uid=DBA;pwd=SQL;dbf=C:¥MLTutorial¥remote.db;eng=re
mote"
```

これで、データ・ソースを検証できるようになります。

❖ 新しいデータ・ソースを検証するには、次の手順に従います。

1. ODBC アドミニストレータを起動します。

   [スタート] メニューから、[プログラム] － [Sybase SQL Anywhere 9] － [Adaptive Server Anywhere] － [ODBC アドミニストレータ] の順に選択します。

   [ODBC データ・ソース・アドミニストレータ] が表示されます。

2. [ユーザ DSN] タブをクリックします。

3. リストをスクロールして、新しいデータ・ソースを探します。

4. データ・ソースを選択し、[構成] をクリックします。

   [Adaptive Server Anywhere 9 の ODBC 設定] ダイアログが表示されます。

5. [ODBC] タブで [テスト接続] ボタンをクリックし、作成したデータ・ソースをテストします。

Mobile Link 同期サーバと Mobile Link 同期クライアントは、それぞれ、ODBC データ・ソースを使用して統合データベースとリモート・データベースに接続します。

データベースへの移植

次に、Interactive SQL を使用してスクリプトで SQL 文を実行し、新しく初期化したデータベース用のテーブルを作成します。スクリプトには、統合データベースとリモート・データベースにテーブルを作成し、データを挿入する SQL 文が含まれています。このスクリプトは、リモート・データベース上に同期サブスクリプションとパブリケーションも作成します。

❖ Interactive SQL からスクリプトを実行するには、次の手順に従います。

1. Interactive SQL を起動します。

[スタート] － [プログラム] － [SQL Anywhere 9] － [Adaptive Server Anywhere] － [Interactive SQL] を選択します。

または

コマンド・プロンプトで、「*dbisql*」と入力します。

2 統合データベースに接続します。

[SQL] メニューから、[接続] を選択します。

[ODBC データ・ソース名] オプションを選択し、[参照] を使用して test_consol データ・ソースを選択します。

[OK] をクリックして、このデータベースに接続します。

3 統合データベースの中にテーブルを作成し、そのテーブルにローをいくつか追加します。

- 次のように入力します。

```
CREATE TABLE cust (
     cust_id int default autoincrement primary
key,
     emp_id int,
     cust_name varchar( 128 )
     );
 -- add data to cust table
 INSERT INTO cust ( emp_id, cust_name ) VALUES (
1, 'cust1' );
 INSERT INTO cust ( emp_id, cust_name ) VALUES (
1, 'cust2' );
 INSERT INTO cust ( emp_id, cust_name ) VALUES (
2, 'cust3' );
 COMMIT;
```

4 テーブルの作成が成功したことを検証します。

次のコマンドを入力し、3 つのローが返ることを確認します。

```
SELECT * FROM cust
```

5 次に、リモート・データベース *remote.db* に対して手順 2、3、4 を繰り返します。次の SQL 文を使用します。リモート・データベースのテーブルには、ローは追加されません。

```
CREATE TABLE cust (cust_id int default autoincrement
primary key,
     emp_id int,
     cust_name varchar( 128 )
     )
```

クエリ SELECT * FROM cust では、ローは表示されません。

6 Interactive SQL の両方のインスタンスを閉じます。

ヒント

作成したデータベースをすぐに起動する必要がある場合は、統合データベースとリモート・データベースに対して *C:¥MLTutorial* ディレクトリから次のコマンド・ラインを使用できます。

```
dbeng9 c:¥MLTutorial¥consol.db
```

と

```
dbeng9 c:¥MLTutorial¥remote.db
```

レッスン 2 ：同期のためのデータベースの準備

同期は、次の要素によって制御されます。

- **同期パブリケーション、ユーザ、サブスクリプション**　これらは、リモート・データベースごとに定義されます。
- **同期スクリプト**　これらは SQL で記述され、統合データベースの中に保持されます。または、同期スクリプトを Java で記述し、Mobile Link 同期サーバがアクセスできるロケーションに保存することもできます。このチュートリアルでは、SQL スクリプトを使用します。

同期スクリプトは、パブリケーションやサブスクリプションと同じように Sybase Central を使用して、記述、表示、修正を行うことができます。

同期サブスクリプションとパブリケーションの作成

Mobile Link 同期を実行するには、Mobile Link 同期パブリケーション、ユーザ、サブスクリプションが必要です。これらはそれぞれ、リモート・データベースに定義されます。

❖ リモート・データベースにパブリケーションと同期サブスクリプションを追加するには、次の手順に従います。

1. Sybase Central を起動し、リモート・データベースに接続します。
   - [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [Sybase Central] の順に選択します。
   - 左ウィンドウ枠で、Adaptive Server Anywhere プラグインを選択します。

     [ツール] メニューから [接続] を選択します。
   - ODBC データ・ソース名「test_remote」を入力し、[OK] をクリックして接続します。
2. リモート・データベースにパブリケーションを追加します。
   - Sybase Central で、リモート・データベースを開きます。

- [パブリケーション] フォルダを開きます。
- [ファイル] メニューから [新規] － [パブリケーション] を選択します。[パブリケーション作成] ウィザードが表示されます。
- パブリケーション Customer を指定し、[次へ] をクリックします。
- テーブル **cust** をダブルクリックしてそれを選択済みテーブル・リストに追加し、[完了] をクリックしてパブリケーションを作成します。

3 リモート・データベースに Mobile Link ユーザ名を追加します。

- Sybase Central で、リモート・データベースを開きます。
- [Mobile Link ユーザ] フォルダを開きます。
- [ファイル] メニューから [新規] － [Mobile Link ユーザ] を選択します。[新しい Mobile Link ユーザの作成] ウィザードが表示されます。
- ユーザ **ml_user** を指定し、[完了] をクリックしてユーザを作成します。

4 作成したパブリケーションに Mobile Link ユーザのサブスクリプションを作成します。

- 左ウィンドウ枠で、[パブリケーション] フォルダを展開して、Customer パブリケーションを表示します。
- 左ウィンドウ枠で、[Mobile Link ユーザ] フォルダを展開して、Mobile Link ユーザ ml_user を表示します。
- その Mobile Link ユーザを Customer パブリケーションにドラッグします。
- 作成したパブリケーションにそのユーザのサブスクリプションを作成するかどうかを確認するメッセージが表示されます。[OK] をクリックします。

5　アドレス情報をサブスクリプションに追加します。

アドレス情報は、Mobile Link クライアントが適切な Mobile Link 同期サーバを検出できるようにするために使用されます。

- 左ウィンドウ枠で、Mobile Link ユーザ [ml_user] を選択し、[ファイル] メニューから [プロパティ] を選択します。
- [接続] タブで次の情報を入力します。他のフィールドはブランクにしておきます。
  - **[ホスト]:** localhost

    Mobile Link 同期サーバが別のマシン上で実行されている場合は、**localhost** の代わりに、マシン名または IP 番号を入力します。
- [Certicom セキュリティを有効にする] チェックボックスがオフになっていることを確認します。
- [OK] をクリックして、アドレス情報を追加します。

これで、リモート・データベースの準備が完了しました。次の手順では、統合データベース用の同期スクリプトを追加します。

同期スクリプトの統合データベースへの追加

スクリプトは、それぞれ指定の「**スクリプト・バージョン**」に属しています。スクリプトは、統合データベースにスクリプト・バージョンを追加した後に追加してください。

❖ スクリプト・バージョンを追加するには、次の手順に従います。

1　Sybase Central を起動し、Mobile Link プラグ・インを使用して test_consol データ・ソースに接続します。

2　[test_consol] フォルダを展開します。

3　[バージョン] フォルダを開きます。

4 [バージョンを追加] をダブルクリックします。新しいバージョンに **default** という名前を付けます。

❖ 統合データベースに同期テーブルを追加するには、次の手順に従います。

1 test_consol アイコンの下にある [テーブル] フォルダを開きます。

2 DBA コンテナを開きます。

3 [ファイル] メニューから、[同期テーブルに追加] を選択します。

これで、これらのテーブルを同期させるテーブルとして指定したので、統合データベースへのアップロードとダウンロードを実行する新しいテーブル・スクリプトを追加できます。

❖ 各同期テーブルにテーブル・スクリプトを追加するには、次の手順に従います。

1 [同期テーブル] フォルダを開いて、[cust] テーブルをダブルクリックします。

2 右ウィンドウ枠で、[テーブル・スクリプトを追加] をダブルクリックします。

   [テーブル・スクリプトを追加] ウィザードが表示されます。

3 ドロップダウン・リストから、**upload_insert** イベントを選択します。

4 [完了] をクリックします。

5 右ウィンドウ枠で、**default** テーブル・スクリプトをダブルクリックします。

   ダイアログが表示されます。

6 次のコードをダイアログに入力します。

```
INSERT INTO cust ( cust_id, emp_id, cust_name )
 VALUES ( ?, ?, ? )
```

7　スクリプトを保存し、ダイアログを閉じます。

8　同じ手順を download_cursor イベントでも繰り返します。次のスクリプトを使用します。

```
SELECT cust_id, emp_id, cust_name
 FROM cust
```

これで、データベースのスナップショットを使った同期を実行して、新規ローを統合データベースにアップロードし、統合データベースのローをリモート・データベースにダウンロードするスクリプトを作成しました。Mobile Link の完全なインストール環境では、他のイベントのスクリプトも追加できます。

レッスン 3 ：Mobile Link 同期サーバの起動

このレッスンでは、統合データベースとリモート・データベースのデータを同期できるように、Mobile Link 同期サーバを起動します。

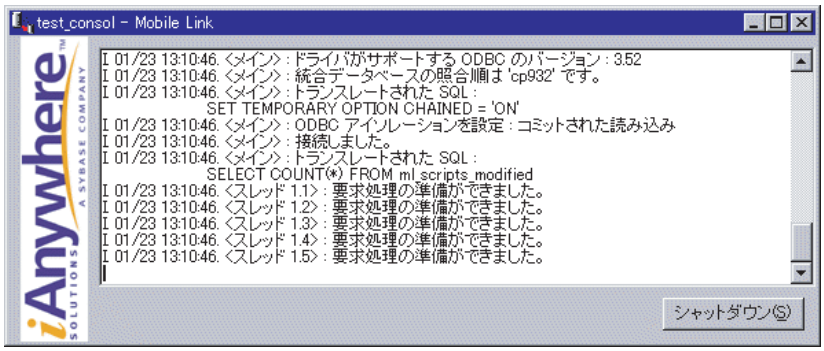
❖ **Mobile Link 同期サーバを起動するには、次の手順に従います。**

- コマンド・プロンプトで次のように入力します。

```
dbmlsrv9 -c "dsn=test_consol" -o mlserver.mls -v+
-dl -zu+
```

各オプションの詳しい説明については、『Mobile Link 管理ガイド』>「Mobile Link 同期サーバのオプション」を参照してください。

Mobile Link 同期サーバ・コマンドを実行すると、次の出力が表示されます。



この画面が表示されたことによって、このチュートリアルの次のレッスンに進む準備が整っていることを確認できます。

レッスン 4 ：Mobile Link 同期 クライアント・ユーティリティの実行

このレッスンでは、Mobile Link 同期クライアントを起動します。dbmlsync クライアント・ユーティリティで -c スイッチを使用して、コマンド・ライン上で接続パラメータを指定します。これらのパラメータは、リモート・データベース用です。

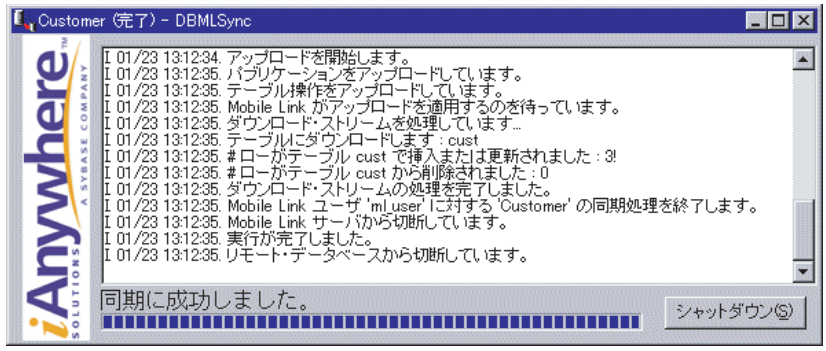
❖ Mobile Link 同期クライアントを実行するには、次の手順に従います。

1. Mobile Link 同期サーバが起動されていることを確認します。
2. コマンド・プロンプトで次のように入力します。

```
dbmlsync -c "dsn=test_remote" -o dbmlsync.out -v
```

詳細については、『Mobile Link クライアント』> 「Mobile Link 同期クライアント」を参照してください。

Mobile Link 同期リモート・データベース・コマンドを実行すると、次の出力が表示されます。



この画面で、同期に成功したことを確認します。次に、リモート・データベースに接続し、cust テーブルに 3 つのローが入力されていることを確認します。これらのローは、統合データベースで追加されたものです。

リモート・データベースと統合データベースでさらにローを追加して、同期させることもできます。

クリーンアップ

このチュートリアルを終了した後、チュートリアルに関連するすべてのデータ・ソースとデータベースを必ず削除してください。

まとめ

これで、2 つのデータベースを作成し、その内容を同期させたことになります。同期は、同期サーバとクライアントによって、いつでも実行できます。

このチュートリアルでは、以下の作業を行いました。

- それぞれが統合データベースとリモート・データベースとして機能する、新しい Adaptive Server Anywhere データベースの作成
- 同期パブリケーションとサブスクリプション定義の記述
- データのアップロードとダウンロードを制御するスクリプトの作成
- Mobile Link 同期サーバと Mobile Link 同期クライアントの起動、リモート・データベースと統合データベースの同期

学習の成果

このチュートリアルでは、次の項目を学習しました。

- 統合システムとしての Mobile Link 同期サーバとクライアントについての知識の習得と Mobile Link 同期サーバとクライアントのコマンド・ラインとスイッチについての学習
- Mobile Link 同期サーバとクライアントのコマンドの実行方法の習得
- 同期スクリプト記述方法の習得

次の作業

Mobile Link を最大限に活用するには、Mobile Link の機能についてさらに詳しく理解する必要があります。その出発点として役立つ次の情報を参照してください。

- このシリーズの次のチュートリアル『Mobile Link チュートリアル』> 「概要」を試す。
- Mobile Link の概要の章、『Mobile Link 管理ガイド』> 「Mobile Link 同期について」を読む。
- Mobile Link 同期サーバの実行の章、『Mobile Link 管理ガイド』> 「現在のセッション以外での Mobile Link の実行」を読む。

- Mobile Link クライアント・ユーティリティの章、『Mobile Link クライアント』> 「Mobile Link 同期クライアント」を読む。

第 18 章

SQL Remote を使用したデータのレプリケーション

この章の内容

この章では、SQL Remote を使用して簡単なレプリケーション・システムを設定する方法について説明します。

SQL Remote について

データの「レプリケーション」とは、物理的にまったく異なるデータベース間でデータを共有することです。どのデータベースで共有データに変更が加えられても、その変更は他のデータベースにレプリケートされます。Sybase では、データのレプリケーション用に次の 3 つの方法を提供します。

- SQL Remote
- Mobile Link
- Replication Server

SQL Remote は、統合データ・サーバと多数のリモート・データベース (通常、多くのモバイル・データベースを含む) との間で双方向レプリケーションを行います。リモート・サイトで必要な管理とリソースは最小限で済みます。このシステムはメッセージ・ベースです。

SQL Remote のインストール環境では、Sybase Adaptive Server Anywhere または Sybase Adaptive Server Enterprise を統合データベースとして使用してください。

SQL Remote は、トランザクション・ログのスキャンと、各トランザクションのために適切なメッセージを準備することによって、データをレプリケートします。それらのメッセージは順序付けされ、リモート・サイトまたは統合サイトに送信されます。SQL Remote が受信メッセージを処理する場合、別のデータベースに適用されたメッセージ順と常に同じ順序で処理します。必要な場合は、先に到着したメッセージがすべて適用されるまで、メッセージの処理を自動的に遅らせます。

同期方法の詳細については、『SQL Remote ユーザーズ・ガイド』> 「はじめに」を参照してください。SQL Remote の完全なマニュアルも含まれています。

Mobile Link 同期の概要については、「データベースと Mobile Link の同期」347 ページを参照してください。

レッスン 1：はじめに

このチュートリアルでは、シンプルな SQL Remote レプリケーション・システムを設定する方法について説明します。

SQL Remote を使用すると、**統合データベース**と一連の「**リモート・データベース**」間で双方向レプリケーションを実行できます。リモート・データベースはラップトップ・コンピュータ上にあることもありますが、統合データベースは通常オフィス内のネットワーク上にあります。すべての設定と管理は統合データベースで実行されます。

このチュートリアルでは、サンプル・データベースにある 1 つのテーブルを同じマシン上にある 1 つのリモート・データベースにレプリケートするという非常に単純な操作について説明します。

このチュートリアルでは、Adaptive Server Anywhere 統合データベースのシステム管理者としてタスクを実行します。レプリケーション・システムは、サンプル・データベースからの単純なテーブルで構成されています。レプリケートするテーブルは department テーブルです。これは、データベース内で最も単純なテーブルの 1 つです。

このチュートリアルの手順は次のとおりです。

1. 統合データベースの設定
2. 単一のリモート・データベースを持つファイル共有レプリケーション・システムの作成
3. 2 つのデータベース間でデータをレプリケート

チュートリアルで使用するディレクトリの作成

チュートリアルを開始する前に、チュートリアルで作成するデータベースと他のファイルを置くディレクトリを作成する必要があります。また、サンプル・データベースを変更後、元のデータベースが必要になる場合に備えて、サンプル・データベースのコピーも作成します。

❖ チュートリアルを準備するには、次の手順に従います。

1. このチュートリアルで作成するファイルを置くディレクトリを作成します。ここでは *c:¥tutorial* とします。

2. *DBA* と *field* という名前で、*tutorial* のサブディレクトリを作成します。ここでは、*c:¥tutorial¥DBA* と *c:¥tutorial¥field* にします。これらのサブディレクトリには、レプリケーション・システム内にある 2 つのユーザ ID のそれぞれに対するメッセージを置きます。

3. サンプル・データベースのコピーを作成します。コピー用の新しいフォルダを *demoback* などの名前で作成し、*asademo.db* をコピーしてそのフォルダに入れます。

次の手順

これで、統合データベースを作成する準備ができました。

レッスン 2：統合データベースの設定

この項では、簡単な SQL Remote レプリケーションシステムの統合データベースを準備する方法について説明します。この準備を行うには、DBA 権限が必要です。

レプリケーションに使用する統合データベースを準備するには、次の手順を実行します。

1. レプリケーションに使用するメッセージ・タイプを作成します。
2. 出力メッセージのソースを識別できるように、ユーザ ID に PUBLISH パーミッションを付与します。
3. メッセージを受信するすべてのユーザ ID に REMOTE パーミッションを付与します。
4. レプリケートするデータを記述するパブリケーションを作成します。
5. パブリケーションの受信側を記述するサブスクリプションを作成します。

SQL Remote メッセージ・タイプの追加

レプリケーションの一部として送信されるすべてのメッセージは、メッセージ・タイプを使用します。メッセージ・タイプの記述は、次の 2 つの部分で構成されます。

- SQL Remote でサポートされているメッセージ・リンク。このチュートリアルでは、FILE リンクを使用します。これはファイル共有メソッドであり、このメソッドではメッセージはディスク上のファイルに入れられ、意図された受信者によって読み込まれます。SQL Remote は、ftp プロトコルと e-mail プロトコルもサポートしています。
- 出力メッセージのソースを識別する、メッセージ・リンクのアドレス。このチュートリアルでは、ファイル・パスを使用して、メッセージを置く場所を指定します。

メッセージ・タイプはすべての新しいデータベースで作成されますが、使用するメッセージ・タイプにアドレスを指定する必要があります。

❖ メッセージ・タイプにアドレスを追加するには、次の手順に従います。

1. Sybase Central から、ユーザ ID を **DBA**、パスワードを **SQL** としてサンプル・データベースに接続し、左ウィンドウ枠でサンプル・データベースのコンテナ (**asademo**) を開きます。
2. 左ウィンドウ枠で、[SQL Remote ユーザ] フォルダをクリックします。
3. SQL Remote ユーザの右ウィンドウ枠で、[メッセージ・タイプ] タブをクリックします。
4. [FILE] メッセージ・タイプをクリックし、[ファイル] メニューから [プロパティ] を選択します。
5. リモート・ユーザ用のリターン・アドレスとなる、パブリッシャ・アドレスを入力します。パブリッシャ・アドレスは、統合データベース (*DBA*) のメッセージを置くためにレッスン 1 で作成したディレクトリです。ここでは、「*c:¥tutorial¥DBA*」と入力します。
6. [OK] をクリックして、メッセージ・タイプを保存します。

データベースへのパブリッシャの追加

SQL Remote レプリケーション・システム内の各データベースには、データの「**パブリッシャ**」を識別する単一のユーザ ID が必要です。ここでは、ユーザ ID DBA をパブリッシャにします。

❖ パブリッシャを設定するには、次の手順に従います。

1. 左ウィンドウ枠で、[ユーザとグループ] フォルダをクリックします。

2　右ウィンドウ枠で [DBA] をクリックし、[ファイル] メニューから [パブリッシャに変更] を選択します。

[DBA] のとなりの [タイプ] カラムに、[パブリッシャ] と表示されます。

1 つのデータベースに設定できるパブリッシャは、1 つだけです。どのユーザがパブリッシャかは、[ユーザとグループ] フォルダを開けばいつでも確認できます。

データベースへのリモート・ユーザの追加

統合データベースでは、各リモート・データベースは REMOTE パーミッションを持つユーザ ID で識別されます。

リモート・ユーザをデータベースに追加するときに、各ユーザが使用するメッセージ・システムとそのメッセージ・システムにおける各ユーザのアドレスを、それぞれのデータベース・ユーザ ID と共に格納する必要があります。

❖　リモート・ユーザを追加するには、次の手順に従います。

1　左ウィンドウ枠で、[SQL Remote ユーザ] フォルダをクリックします。

2　[ファイル] メニューから、[新規] － [SQL Remote ユーザ] を選択します。

[リモート・ユーザ作成] ウィザードが表示されます。

3　新しいリモート・ユーザの名前を「**field**」と入力して、[次へ] をクリックします。

4　ユーザに接続を許可することを確認して、パスワード「**field**」を入力します。確認のためにもう一度パスワードを入力します。[次へ] をクリックします。

5　リモート・ユーザのパーミッションには、[DBA] と [Remote DBA] を選択します。[次へ] をクリックします。

6 メッセージ・タイプ [FILE] を選択して、テキスト・ボックスにリモート・アドレス「**field**」を入力します。[次へ] をクリックします。

7 [送信して閉じる] を選択します。[次へ] をクリックします。

8 [完了] をクリックし、リモート・ユーザを作成します。

[SQL Remote ユーザ] フォルダに、リモート・ユーザ field が表示されます。

注意 ウィザードで DBA パーミッションを設定するのを忘れた場合は、そのユーザをクリックして [ファイル] メニューから [プロパティ] を選択し、[権限] タブで [DBA] をチェックすると設定できます。

以上でこのシステムを使用するユーザの作成が完了しました。

パブリケーションとサブスクリプションの追加

この項では、パブリケーションをデータベースに追加する方法と、そのパブリケーションにユーザのサブスクリプションを追加する方法について説明します。パブリケーションは、テーブル department のすべてのローをレプリケートします。

❖ パブリケーションを追加するには、次の手順に従います。

1 左ウィンドウ枠で、[パブリケーション] フォルダをクリックします。

2 [ファイル] メニューから、[新規] － [パブリケーション] を選択します。

[パブリケーション作成] ウィザードが表示されます。

3 新しいパブリケーションの名前を「**DepartmentPub**」と入力し、[次へ] をクリックします。

4 [使用可能なテーブル] のリストから、[department] を選択します。[追加] をクリックします。

右側の [選択したテーブル] リストに、そのテーブルが表示されます。

5　パブリケーションを作成するには、[完了] をクリックしてください。

サブスクリプションの追加

パブリケーションに加えられた変更を受信する各ユーザ ID には、そのパブリケーションに対する「**サブスクリプション**」が必要です。サブスクリプションは、有効なリモート・ユーザに対してのみ作成できます。リモート・データベースのユーザ field に対する DepartmentPub パブリケーションに、サブスクリプションを追加する必要があります。

❖ サブスクリプションを追加するには、次の手順に従います。

1　左ウィンドウ枠で、[パブリケーション] フォルダを展開します。

2　左ウィンドウ枠で、[DepartmentPub] パブリケーションを選択します。

3　右ウィンドウ枠で、[SQL Remote サブスクリプション] タブをクリックします。

4　[ファイル] メニューから、[新規] － [SQL Remote サブスクリプション] を選択します。

[新しい SQL Remote サブスクリプションの作成] ダイアログが表示されます。

5　ダイアログで、ユーザ [field] をクリックします。

6　[完了] をクリックして、サブスクリプションを作成します。

以上で統合データベースの設定が完了しました。

次の手順

これで、リモート・データベースを作成できます。

レッスン 3：リモート・データベースの設定

リモート・データベースを作成し、メッセージの送受信を行い SQL Remote セットアップに組み込まれるように設定する必要があります。

データベース抽出ユーティリティを使用すると、サブスクリプションと必要なユーザ ID がそろったリモート・データベースを作成するのに必要な手順を、すべて実行できます。

リモート・データベースの作成

リモート・ユーザ field の統合データベースからデータベースを 1 つ抽出する必要があります。

❖ **リモート・データベースを抽出するには、次の手順に従います。**

1. 左ウィンドウ枠でサンプル・データベース asademo (DBA) をクリックし、[ファイル] メニューから [データベースの抽出] を選択します。

   [データベースの抽出] ウィザードが表示されます。

2. ウィザードの概要ページで、[次へ] をクリックします。

3. ダイアログで、[asademo] をクリックします。[次へ] をクリックします。

4. 独立性レベルを 3 に設定し、[次へ] をクリックします。

5. ユーザ [field] をクリックし、[自動的にサブスクリプションを開始] オプションを選択します。[次へ] をクリックします。

6. [新しいデータベースへの抽出と再ロード] を選択します。[次へ] をクリックします。

7. データベースをファイル *c:¥tutorial¥field¥field.db* に保存します。[次へ] をクリックします。

8　[構造とデータを抽出] を選択します。

9　[プライマリ・キーでデータを並べ替える] オプションを選択します。[次へ] をクリックします。

10　[外部キー]、[プロシージャとファンクション]、[トリガ]、[ビュー] の各チェックボックスをオンにします。

11　[完全に修飾されたパブリケーション定義を抽出] チェックボックスをオフにします。[次へ] をクリックします。

12　[新しいデータベースに接続] チェックボックスをオフにします。

13　[完了] をクリックして、リモート・データベースを抽出します。

[データベースの抽出中] ウィンドウが表示され、抽出の進行状況が表示されます。抽出が完了すると、ウィンドウが閉じます。

注意　適切な SQL Remote セットアップのリモート・データベース **field** は、データベース・サーバと必要なすべてのクライアント・アプリケーションとともにコンピュータ上にロードされます。このチュートリアルでは、データベースの場所を変更しないで、Interactive SQL を使用してデータの入力とレプリケートを行います。

データベースが適切に作成されているかどうかの確認

抽出ユーティリティがどのような処理を行ったかを調べるため、*field* データベースに接続してすべてのデータベース・オブジェクトが作成されていることを確認します。

❖ リモート・データベースをブラウズするには、次の手順に従います。

1　Sybase Central で、[接続] ボタンをクリックします。

[新しい接続] ダイアログが表示されます。

2 [Adaptive Server Anywhere 9] を選択し、[OK] をクリックします。

3 [ID] タブで、ユーザ ID「**field**」とパスワード「**field**」を入力します。

4 [データベース] タブで、データベースのパス「*c:¥tutorial¥field¥field.db*」を入力します。

5 [OK] をクリックして、このデータベースに接続します。

   Sybase Central の左ウィンドウ枠に、データベース field が表示されます。

6 左ウィンドウ枠で、[テーブル] フォルダを展開します。リスト内に、ユーザ DBA が所有する department テーブルが表示されます。

7 左ウィンドウ枠で、[department] テーブルをクリックします。

8 右ウィンドウ枠で、[データ] タブをクリックし、この department テーブルのローを 5 つ表示します。

9 [SQL Remote ユーザ] フォルダを開きます。

   [SQL Remote] タブに、ユーザ DBA が統合ユーザとして指定されています。これは、DBA が統合データベースのパブリッシャで、階層では field データベースの上という意味です。

10 SQL Remote ユーザの右ウィンドウ枠で、[メッセージ・タイプ] タブをクリックします。

   field は、パブリッシャのアドレスとして指定されています。統合データベースから送られてくるすべてのデータは、ユーザ DBA が発信者です。これと同じように、このデータベースから送信されたデータはすべて、ユーザ **field** が発信者です。

11 [パブリケーション] フォルダを開きます。DepartmentPub パブリケーションが存在していることが確認できます。

次の手順

以上でこのシステムは、レプリケーションの準備が完了しました。

レッスン 4：データのレプリケーション

ここまでの作業で、レプリケーション・システムが使えるようになりました。この項では、統合データベースからリモート・データベースに、また、リモート・データベースから統合データベースに、データをレプリケートします。

統合データベースへのデータの追加

最初に、統合データベースにローを入力します。

❖ 統合データベースにデータを入力するには、次の手順に従います。

1. 左ウィンドウ枠で、統合データベース (asademo) の [テーブル] フォルダを展開します。
2. 左ウィンドウ枠で、department テーブルを選択します。
3. 右ウィンドウ枠で、[データ] タブをクリックします。
4. ツール・バーのプラス記号 (+) をクリックして、ローを 1 つ追加します。
5. 次の値を入力します。

カラム	値
dept_id	202
dept_name	Eastern Sales
dept_head_id	(NULL) のまま

6. [Enter] を押します。

   [確認] ダイアログが表示されたら、[OK] をクリックしてこのローを更新します。

次の手順では、追加した新しいローをリモート・データベースに送信します。

リモート・データベースにデータを送信するために、統合データベースで Message Agent を実行します。*dbremote* プログラムが Adaptive Server Anywhere の Message Agent です。

❖ リモート・データベースにデータを送信するには、次の手順に従います。

1. コマンド・プロンプトで、チュートリアル・ディレクトリに移動します。ここでは、*c:¥tutorial* に移動します。
2. 統合データベースに対して Message Agent を実行するには、次のコマンドを実行します。

   ```
   dbremote -c "dbn=asademo;uid=DBA;pwd=SQL"
   ```

   [SQL Remote] ウィンドウが表示され、レプリケーションのステータスに関するメッセージが表示されます。
3. SQL Remote に [実行が完了しました] というメッセージが表示されたら、[シャットダウン] をクリックします。

リモート・データベースで insert 文を受信するには、そのリモート・データベースで Message Agent である *dbremote* を実行します。

❖ リモート・データベースでデータを受信するには、次の手順に従います。

1. コマンド・プロンプトで、チュートリアル・ディレクトリに移動します。ここでは、*c:¥tutorial* に移動します。
2. 次のコマンドを実行して、**field** データベースに対して Message Agent を実行します。

   ```
   dbremote -c "dbn=field;uid=field;pwd=field"
   ```

   [SQL Remote] ウィンドウが表示され、レプリケーションのステータスが表示されます。DBA からメッセージを受信したことがこのウィンドウで示されます。

3　SQL Remote に [実行が完了しました] というメッセージが表示されたら、[シャットダウン] をクリックします。

注意

- SQL Remote の稼働中は、[SQL Remote] ウィンドウにステータス情報が表示されます。この情報は、記録を保存するログ・ファイルに出力できます。このステータス情報によって、Message Agent がまず *asademo* からメッセージを受信し、そのあとでメッセージを送信するのがわかります。このリターン・メッセージには、レプリケーションの更新の受信に成功したことの確認が含まれています。このような確認は、メッセージ・システム・エラーが発生した場合でもメッセージの配信を保証する、SQL Remote メッセージ・トラッキング・システムの一部です。

- データベースの現在のステータスに応じて、次の 3 種類の接続パラメータのうちいずれかを使用してデータベースに接続します。

  - **dbf**　データベース・ファイルを使用してデータベースに接続します。このパラメータを使用する場合は、データベース・ファイル自体を指定する必要があります。稼働中のサーバがない場合はサーバを起動し、データベースをそのサーバ上にロードします。サーバがすでに稼働している場合は、データベースをデフォルト・サーバ上にロードします。

  - **dbn**　データベース名を使用してデータベースに接続します。このパラメータを使用する場合は、データベースの名前を指定する必要があります。データベースがすでに稼働している場合にのみ、このパラメータを使用できます。

  - **dsn**　データベース・ソースを使用してデータベースに接続します。データ・ソースとは、システム・レジストリまたはファイルのセットに保管されるパラメータの集合です。ソースは単にその名前で参照されます。

    データ・ソースの詳細については、『ASA データベース管理ガイド』> 「DataSourceName 接続パラメータ [DSN]」を参照してください。

データ受信の確認

Sybase Central から、department テーブルを調べ、さきほどのローが受信していることを確認します。

❖ データの受信を確認するには、次の手順に従います。

1. [*field*] データベース・コンテナを開きます。
2. 左ウィンドウ枠で、[department] テーブルを選択します。右ウィンドウ枠で [データ] タブをクリックして、このテーブルのローを表示します。

   統合データベースに入力した Eastern Sales 部署が department テーブルに入っています。

リモート・データベースから統合データベースへのレプリケーション

データをリモート・データベースに入力して、統合データベースに送信してみてください。ここではその概略だけを示します。

❖ リモート・データベースから統合データベースにレプリケートするには、次の手順に従います。

1. リモート・データベースで部署のローを 1 つ挿入します。部署の dept_id は 203、dept_name は Western Sales とします。
2. *dbremote* を次のように実行して、統合データベースにメッセージを送信します。

   ```
   dbremote -c "dbn=field;uid=field;pwd=field"
   ```

3. *dbremote* を次のように実行して、統合データベースでメッセージを受信します。

   ```
   dbremote -c "dbn=asademo;uid=DBA;pwd=SQL"
   ```

4. 統合データベースにある department テーブルのデータを表示します。データの中に Western Sales というローがあります。

次の手順

以上で、チュートリアルは終了しましたが、次の項に進んで、サンプル・データベースを元のフォームに戻してください。

レッスン 5：データベースとデータベース設定値のリストア

チュートリアルを完了したら、サンプル・データベースに対して行った変更を取り消すことが重要です。設定が正しくリセットされるように、次の手順を確実に完了してください。

❖ リモート・ユーザを削除するには、次の手順に従います。

1 左ウィンドウ枠で、サンプル・データベースの [ユーザとグループ] フォルダをクリックします。

2 ユーザ [field] をクリックし、[編集] メニューから [削除] を選択します。[はい] をクリックして、リストからこのユーザを削除します。

❖ パブリケーションを削除するには、次の手順に従います。

1 左ウィンドウ枠で、[パブリケーション] フォルダをクリックします。

2 左ウィンドウ枠で [DepartmentPub] をクリックし、[編集] メニューから [削除] をクリックします。[はい] をクリックして、削除を確認します。

❖ サンプル・データベースからパブリッシュ中のステータスを取り消すには、次の手順に従います。

1 左ウィンドウ枠で、[ユーザとグループ] フォルダをクリックします。

2 左ウィンドウ枠で DBA ユーザをクリックし、[ファイル] メニューから [パブリッシャの取り消し] を選択します。

データベース実行中のステータスの取り消し
ユーザがデータベースを実行している間はそのユーザのステータスを修正できません。ユーザがアイドル状態であることを確認してから、ステータスの取り消しまたは呼び出しを行ってください。

❖ 元のメッセージ・タイプの設定値をリストアするには、次の手順に従います。

1. [SQL Remote ユーザ] フォルダを選択し、右ウィンドウ枠で [メッセージ・タイプ] タブをクリックします。
2. 右ウィンドウ枠で [FILE] メッセージ・タイプを右クリックして、[プロパティ] を選択します。
3. パブリッシャ・アドレスを削除してから、[OK] をクリックしてこの設定をリストアします。

❖ リモート・データベースの停止

1. 左ウィンドウ枠で、field データベース・コンテナを選択します。
2. [ツール] メニューから、[切断] を選択します。

   [接続] ダイアログが表示されます。
3. field データベースへの接続を選択します。
4. [切断] をクリックします。

❖ リモート・データベースを削除するには、次の手順に従います。

1. 左ウィンドウ枠で、Adaptive Server Anywhere プラグインを選択します。
2. 右ウィンドウ枠で、[ユーティリティ] タブをクリックします。

3 右ウィンドウ枠で、[データベースの消去] をダブルクリックします。

[データベースの消去] ウィザードが表示されます。

4 ウィザードの指示に従って、field データベースを消去します。

5 削除するデータベースの名前を入力するか、または [参照] をクリックしてそのデータベースを検索します。たとえば、「*c:¥tutorial¥field¥field.db*」と入力します。[完了] をクリックし、データベース ファイルを消去します。

注意 以上でチュートリアルは完了しました。ディスク領域を節約するために、チュートリアル・ディレクトリ (*c:¥tutorial*) を削除することをおすすめします。

データベース内のデータのリストア

クリーンアップ処理の最も重要な点は、サンプル・データベースに対する変更を確実に逆の手順で元に戻すことです。このマニュアルのあとの章で他のチュートリアルを実行するため、サンプル・データベースの整合性は大変重要です。

❖ 挿入されたデータをサンプル・データベースから削除するには、次の手順に従います。

1 Sybase Central から、asademo データベースの department テーブルのデータを表示します。

2 dept_id の値が 202 と 203 のローを削除して、テーブルを元の状態にリストアします。

まとめ

このチュートリアルでは、以下のことについて学びました。

- 簡単なレプリケーション・システムの統合データベースの準備。
- リモート・データベースの作成と設定。
- 2 つのデータベース間における双方向のデータのレプリケーション。
- データベースとデータベース設定のリストア。
- すべての手順の確認。

参考資料

SQL Remote の詳細については、『SQL Remote ユーザーズ・ガイド』>「はじめに」を参照してください。

Mobile Link 同期の概要については、「データベースと Mobile Link の同期」347 ページを参照してください。

第 19 章

PowerDesigner によるデータベースの設計

この章の内容

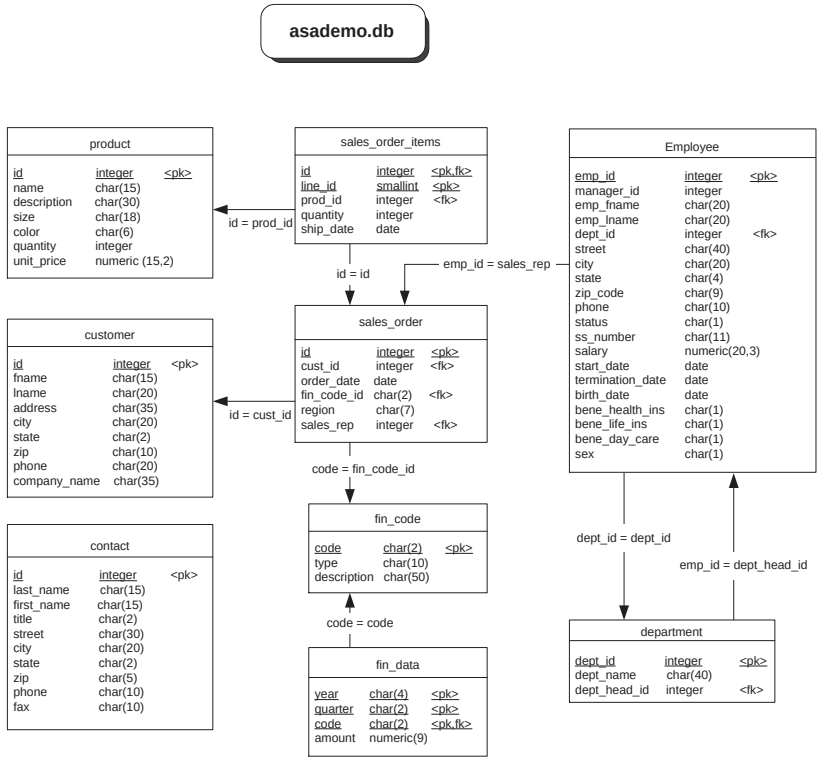
SQL Anywhere Studio には、Sybase の強力なデータベース設計ツール PowerDesigner のモジュールである PowerDesigner Physical Architect が含まれています。このモジュールは、データベース構造をグラフィックで表現する手法を使ってデータベースを生成し、修正する方法を提供します。この手法で、テーブル、カラム、インデックス、参照整合性、ビュー、物理記憶領域、トリガ、およびストアド・プロシージャをカスタマイズしてデータベースを最適化できます。

PowerDesigner について

テーブル、関係、ビュー、トリガなどのデータベースの構造を、データベースの「**スキーマ**」といいます。SQL 文を使用してリンクに対するこれらの要素を作成して配列しますが、グラフィカル・ツールを使用しないでこの作業を行うと混乱する可能性があります。

PowerDesigner を使用すると、データベースの構造がグラフィックで表示されます。さらに、新しいテーブルをドローしたり情報を入力したりするだけで、データベースの構造の修正や、まったく新しい構造の作成が可能です。設計が完了すると、PowerDesigner は新しいデータベースを生成する SQL スクリプトを生成できます。

次の図はサンプル・データベースの構造を示します。この図は PowerDesigner を使用して簡単に作成できます。



データベースのパフォーマンスは、設計によって大きく異なります。一般に、従業員や製品など、個別のオブジェクトのタイプについての情報は別々のテーブルに保管してください。

これらのテーブル間の関係は参照を使用して識別できます。この参照とは、一方のテーブルの外部キーが別のテーブルにある特定のローを識別することを意味します。多対 1 の関係と 1 対多の関係は参照で表現できます。多対多の関係の場合は、2 つの参照と別のテーブルが必要になります。

データベース設計の詳細については、『ASA SQL ユーザーズ・ガイド』> 「データベースの設計」を参照してください。

PowerDesigner の詳細については、PowerDesigner メイン・ウィンドウのツールバーから [Help] をクリックして、次の 3 種類のマニュアルにアクセスしてください。『*PowerDesigner General Features Guide*』『*PowerDesigner PDM User's Guide*』『*PowerDesigner Report User's Guide*』

PowerDesigner のその他のチュートリアルについては、PowerDesigner を開いてください。[Help] メニューから、[Where to Start] － [PDM Getting Started] の順に選択します。これにより、『*Physical Data Model Getting Started*』にアクセスできます。

レッスン 1：はじめに

PowerDesigner は、データベースを作成するスクリプト・ファイルからデータベースの構造を読み込みます。ただし一般的には、単に PowerDesigner からデータベースに接続して、PowerDesigner のリバース・エンジニアリング機能を使って設計を直接抽出するほうが簡単です。

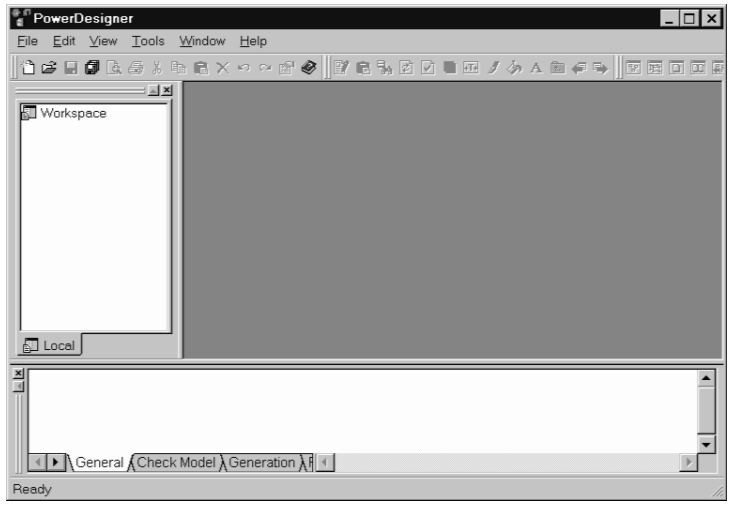
次のチュートリアルでは、サンプル・データベースを使用します。チュートリアルでサンプル・データベースの設計を改善しながら PowerDesigner について説明します。

それぞれの製品の価格は、常に product テーブルから読み込みます。そのため、価格を更新すると、それ以前に受けた注文についてその製品の小売り価格がすべて変更されてしまいます。この問題は、unit_price カラムを sales_order_item テーブルに追加して解消します。追加すると、それぞれの顧客に対する実際の売り値を別々に保管できます。product テーブル内の価格には、現在の定価が記録されます。

❖ PowerDesigner を起動するには、次の手順に従います。

1. [スタート] メニューから、[プログラム] － [SQL Anywhere 9] － [PowerDesigner 9] － [PowerDesigner] の順に選択します。

   次のような PowerDesigner のメイン・ウィンドウが表示されます。



PowerDesigner のメイン・ウィンドウには、左側にオブジェクト・ブラウザが、下に出力ウィンドウがあります。

2　[File] メニューから、[New] を選択します。

[New] ダイアログが表示されます。

3　[New] ダイアログで、[Physical Data Model] を選択し、[OK] をクリックします。

[New Physical Data Model] ダイアログが表示されます。

4　[General] タブで、ドロップダウン・リストから [Sybase AS Anywhere 9] を選択します。

5　残りのオプションについては、デフォルト設定をそのまま使用して [OK] をクリックします。

ブラウザとタイトル バーに、「PhysicalDataModel_1」というモデル名が表示されます。

データベースをリバース・エンジニアリングする

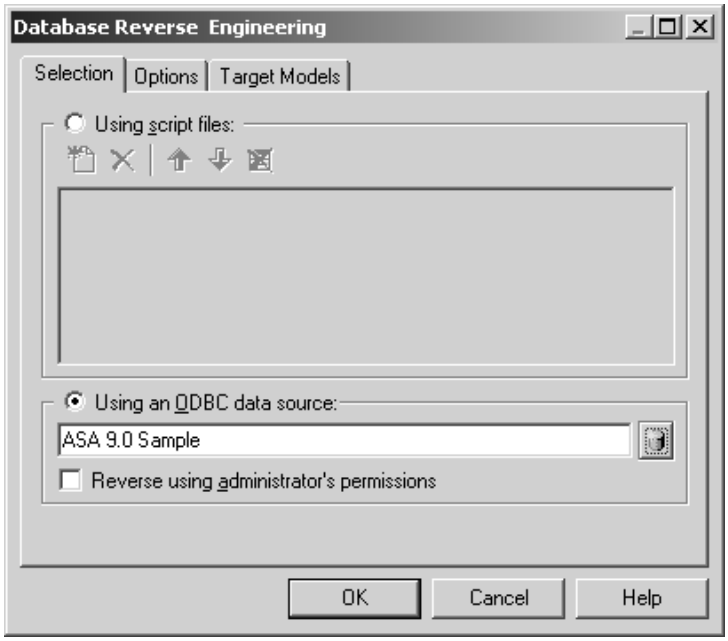
この項では、サンプル・データベースをリバース・エンジニアリングして物理データ・モデル (PDM) を生成します。

❖ データベースをリバース・エンジニアリングするには、次の手順に従います。

1 模式図ウィンドウ (PowerDesigner の中央にある大きなウィンドウ枠) をクリックします。

2 [Database] メニューから [Reverse Engineering Database] を選択します。

[Database Reverse Engineering] ダイアログが表示されます。

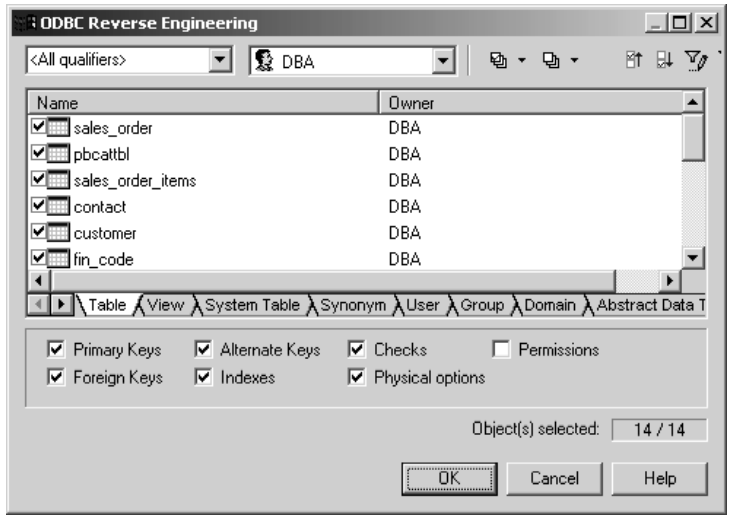


3 [Using a ODBC data source] が選択されていて、データ・ソースが [ASA 9.0 Sample] であることを確認します。

[ASA 9.0 Sample] が表示されない場合は、データ・ソース・フィールドの右にあるアイコンをクリックします。[Connect to an ODBC Data Source] ダイアログが表示されます。[Machine Data Source] を選択し、ドロップダウン・リストから [ASA 9.0 Sample (Adaptive Server Anywhere 9.0)] を選択します。ユーザ ID「**DBA**」とパスワード「**SQL**」を入力します。[Connect] をクリックして、[Database Reverse Engineering] ダイアログに戻ります。

4　[Database Reverse Engineering] ダイアログで、[OK] をクリックします。

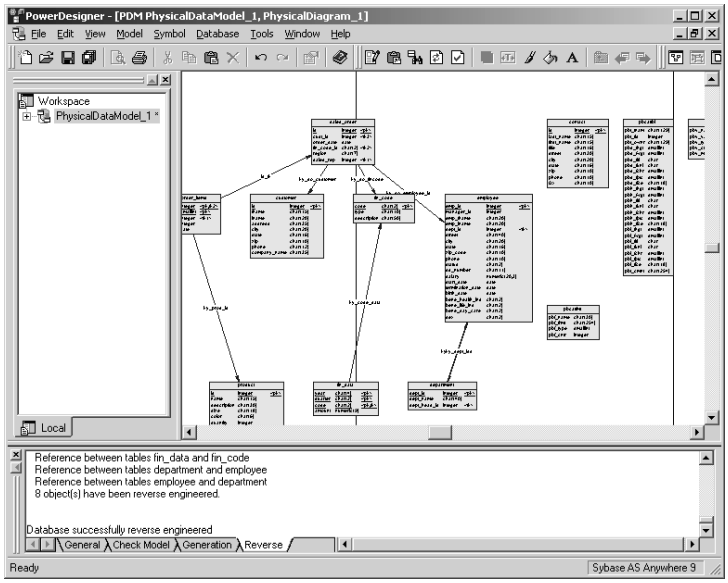
[ODBC Reverse Engineering] ダイアログが表示されます。



ダイアログ・ボックスの下半分に、キーやインデックスなどを選択するチェックボックスが 7 つあります。これらは、「**リバース・エンジニアリング・オプション**」です。[Permission] 以外のすべてのチェックボックスが選択されていること (デフォルト) を確認します。さらに、すべてのテーブルが選択されていること (デフォルト) も確認します。

5　[OK] をクリックして、データベースをリバース・エンジニアリングします。

サンプル・データベースが模式図ウィンドウに表示され、次のようにモデル・オブジェクトがブラウザに表示されます。



6 [Tools] メニューから [Display Preferences] を選択して、表示をどのように修正できるかを調べます。左側にある個々のオプションをクリックして、表示の設定パラメータのページを見ます。個々のページの下にある [Help] ボタンをクリックして、詳細な情報を表示します。

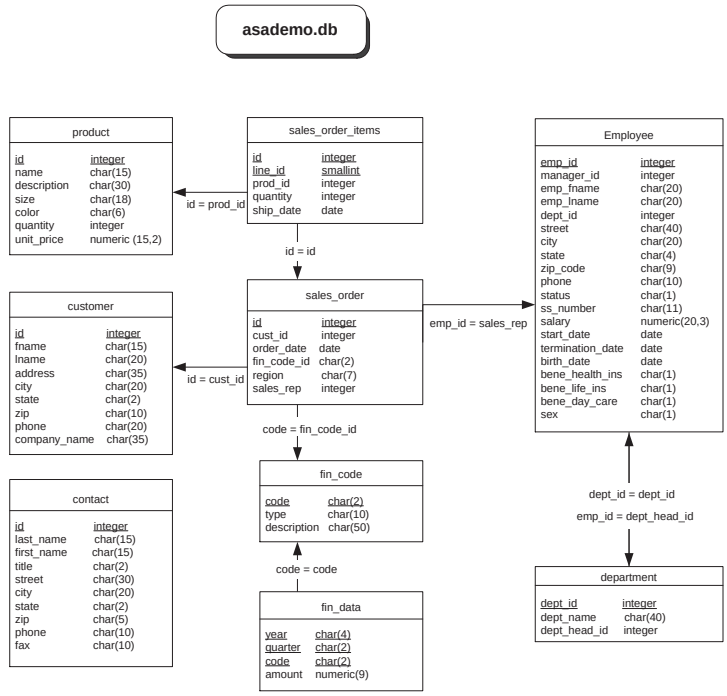
図をチェックします。[F6] キーを使って拡大したり、[F7] キーを使って縮小したりして、テーブルを読みやすくします。

データベース内のそれぞれのテーブルはボックスで示されます。テーブルの名前は、ボックスの一番上に表示されます。その下にはカラムのリストが表示されます。下線が付いているカラム名は、そのテーブルのプライマリ・キーの部分です。それぞれのカラムのデータ型は右側に表示されます。リバース・エンジニアリング後にテーブルの一部が重なって表示される場合があります。[Symbol] メニューから [Auto-Layout] を選択して、それらをレイアウトし直します。

テーブル間の参照は矢印で示されます。矢印の終点は親テーブル (プライマリ・キーが入っているテーブル) を指しています。それぞれの矢印の隣には等号が表示され、参照を示します。

7　テーブルを配置し直して、図を読みやすくします。これは、テーブルをマウスでドラッグして行います。参照の矢印も自動的に一緒に移動します。[SHIFT] キーを押しながらクリックすると、複数のオブジェクトを選択できます。

たとえば、次のような配置が考えられます。



レッスン 2：カラムの追加

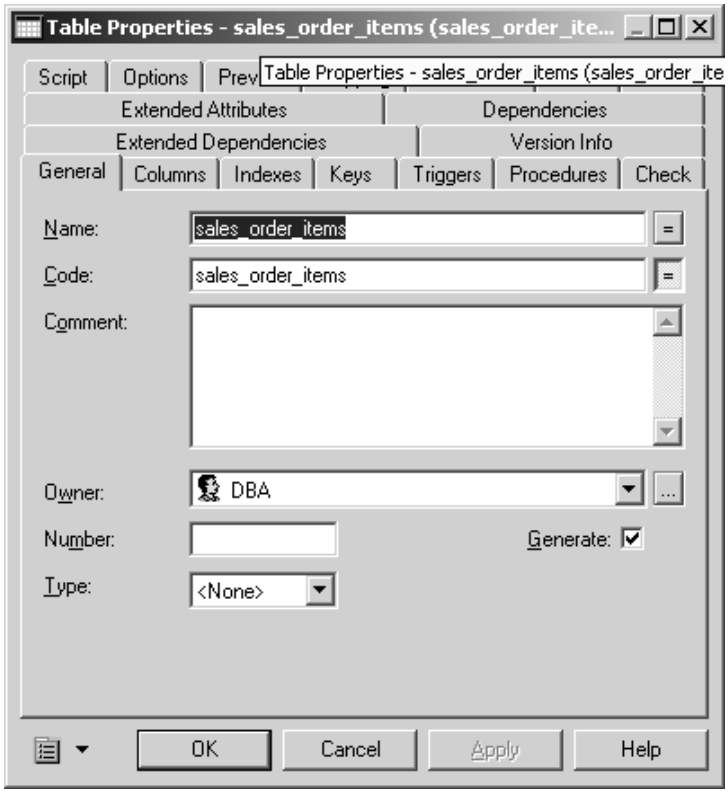
これで、unit_price カラムを sales_order_items テーブルに追加する準備ができました。[Table Properties] シートからカラムのリストにアクセスすることによって、この処理を実行できます。

❖ カラムを追加するには、次の手順に従います。

1 sales_order_items テーブルを選択します。

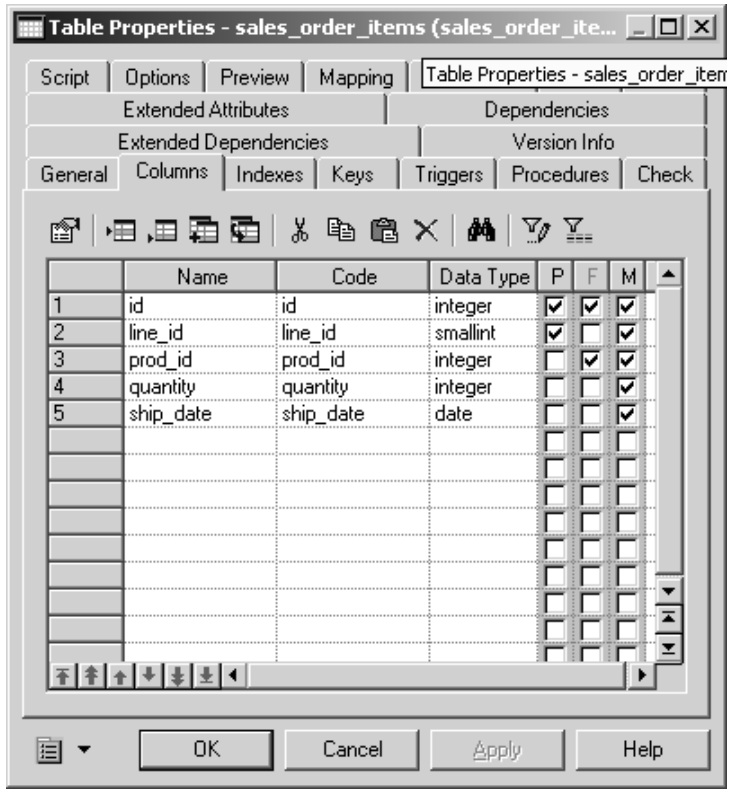
2 [View] メニューから、[Properties] を選択します。

[Table Properties] ダイアログが表示されます。



3　[Columns] タブをクリックします。

カラムのリストが表示されます。



4　単価を入れる新しいカラムを追加します。

[Insert a Row] ボタンをクリックします。

行頭に矢印が表示され、デフォルト名 (Column_6) がついたカラムが表示されます。

5　[Name] カラムに「**unit_price**」と入力します。この名前は、自動的に [Code] に複製されます。

[Data Type] カラムで、ドロップダウン・リストから [Numeric] を選択します。

[Data] フィールドが狭すぎて読めない可能性があります。その場合は、カラムの端を引っ張ると幅が広がります。

6 カラム・プロパティの P、F、M は、それぞれ次に説明するプライマリ・キー (Primary Key)、外部キー (Foreign Key)、必須キー (Mandatory) を表します。

- プライマリ・キーは、テーブルのローをユニークに識別する値があるカラムを示します。
- 外部キーは、別のテーブルのプライマリ・キー・カラムに依存し、そのカラムから移植するカラムを示します。
- 必須キーは、値を割り当てる必要があるカラムを示します。

7 [Mandatory] を選択して、[OK] をクリックします。

8 データベースの図で実行した変更がどのように反映されるかを確認します。

sales_order_items テーブルには、unit_price という名前の新しいカラムが含まれています。

レッスン 3：作業の確認

PowerDesigner を使用すると、新しいモデルのデータベースの設計ミスをすばやく検出できます。

❖ 新しいスキーマを確認するには、次の手順に従います。

1 [Tools] メニューから、[Check Model] を選択します。

[Check Model Parameters] ダイアログが表示されます。デフォルトのパラメータを使用します。

2 [OK] をクリックします。

[Check Model] の結果が [Results List] ダイアログに表示されます。

レッスン 4：変更の保存とデータベースの生成

PowerDesigner では、テーブルとカラムを含むデータベース設計の物理コンポーネントを表すモデルを「**物理データ・モデル（PDM)**」といいます。PowerDesigner は、拡張子 *.PDM* を持つファイルにこのモデルを保管します。

❖ 物理データ・モデル (PDM) を保存するには、次の手順に従います。

1 [File] メニューから、[Save As] を選択します。

2 ファイル名を「*c:¥Temp¥NewDemo.pdm*」と入力します。

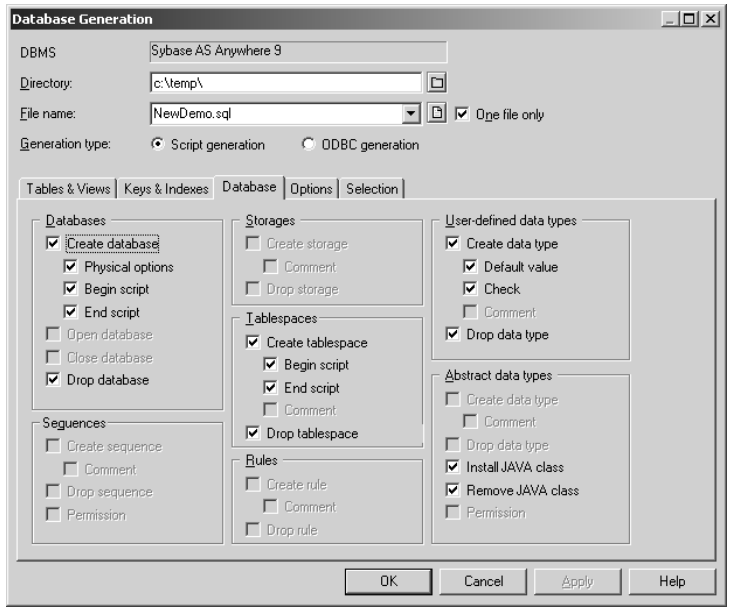
3 [保存] をクリックします。

モデルのコンポーネントをすべて実装する SQL スクリプトを生成できます。その後、その SQL スクリプトを使ってデータベースを生成できます。

❖ SQL スクリプトを生成して新しいデータベースを作成するには、次の手順に従います。

1 [Database] メニューから、[Generate Database] を選択します。

次のような [Database Generation] ダイアログが表示されます。



2　[Directory] フィールドに「*c:¥Temp¥*」、[File Name] フィールドに「*NewDemo.sql*」と入力します。

3　[Script generation] が選択されていることを確認します。

4　[Database] タブをクリックし、[Create database] が選択されていることを確認します。

ほかのタブをクリックして、生成されるスクリプトのその他のさまざまなプロパティを制御するオプションについて確認します。

5　[OK] をクリックします。

スクリプトを作成すると、[Result] ダイアログが表示されます。

6　[Edit] をクリックして、スクリプトを表示します。

変更した内容がスクリプトに反映されていることを確認します。たとえば、新しい office テーブルの定義は次のようになります。

```
/* ========================================= */
 /*   Table: office                            */
 /* ========================================= */
 create table office
 (
     id            integer         not null
         default autoincrement
         check (
             id >= 100),
     name          char(15)        not null,
     street        char(30)        not null,
     city          char(20)        not null,
     state         char(2)         not null,
     zip           char(5)         not null,
     phone         char(10)                ,
     fax           char(10)                ,
     primary key (id)
 );
```

7 確認し終わったら、[Result] ダイアログで [Close] をクリックして、ダイアログを閉じます。

Interactive SQL を使って新しいデータベースを作成する準備ができました。

❖ 新しいデータベースを作成するには、次の手順に従います。

1 Interactive SQL を起動します。

[スタート] メニューから、[プログラム] － [Sybase SQL Anywhere 9] － [Adaptive Server Anywhere] － [Interactive SQL] の順に選択します。

2 ASA 9.0 Sample ODBC データ・ソースを使用して、サンプル・データベースに接続します。

3 空のデータベースを作成します。

- 適切なディレクトリに置き換えて、次の SQL 文を実行します。

```
CREATE DATABASE 'c:¥¥Temp¥¥newdemo.db'
```

ヒント

Interactive SQL で SQL 文を実行するには、[SQL 文] ウィンドウ枠に SQL 文を入力するかコピーしてから、[F5] キーを押します。または、[SQL] メニューから [実行] を選択します。

4 サンプル・データベースとの接続を閉じます。

[SQL] メニューから、[切断] を選択します。

5 新しいデータベースに接続します。

[SQL] メニューから [接続] を選択します。

- [ユーザ ID] に「**DBA**」と入力します。
- [パスワード] に「**SQL**」と入力します。
- [データベース] タブをクリックし、[データベース・ファイル] ボックスに新しいデータベース・ファイルのフル・パスとファイル名を入力します。
- [OK] をクリックします。

6 READ 文を使用します。この文を使用する場合には、ファイル名を二重引用符で囲む必要があります。

- 次の SQL 文を入力して実行します。

```
READ "c:¥¥Temp¥¥newdemo.SQL"
```

これらの基本的な手順に従って、他のデータベースを修正できます。

まとめ

このチュートリアルで説明したのは、PowerDesigner の基本機能の一部です。実際には、PowerDesigner を使用すると、すべてのテーブル、ビュー、インデックス、参照、トリガ、プロシージャを含むデータベース・スキーマの設計全般または修正の全般を行うことができます。

ドメイン

ここで説明した機能以外にも、より大きなデータベースを設計するときの作業を非常に容易にするための機能があります。たとえば、特定の「**ドメイン**」を指定できます。ドメインには、電話番号などの特定な型のデータが入っています。ドメインは関連づけられたデータ型を持っていますが、より特定的です。たとえば、ID 番号のドメインを作成できます。テーブルで ID 番号が必要な場合は、そのカラムを ID 番号のドメインと関連づけることができます。すると、そのドメインと関連づけられているすべてのプロパティとチェックが自動的に付け加えられます。

ドメインを指定することによって、定義を繰り返す回数が減ります。また、ユーザの作業を軽減させるだけでなく、異なる型の定義をしたり、不適切な検査手順を実行したりする誤りが減ります。カラムを単に整数として識別するのではなく、そのカラムにどの型のデータが入っているかを指定します。そのデータ型のすべてのインスタンスは共通の定義を共有しています。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「ドメインの使い方」を参照してください。

ビジネス・ルール

ビジネス・ルールとは、ビジネスが行われる方法を文章で表現したものです。たとえば、「*受注商品の出荷日は受注日以降とする*」はビジネス・ルールです。

ビジネス・ルールは次の 4 つのカテゴリに分けられます。

- **定義**　オブジェクト本来のプロパティを表します。定義は一般にエンティティを記述します。
- **事実**　確実な事柄または存在を表します。事実は一般に関係を記述します。
- **検証**　ある値についての制約です。

- **式**　値を算出するために使用される計算です。

ビジネス・ルールは、顧客がデータベースに実行させようとするタスクに直接関連しているので大変便利です。ビジネス・ルールを記録して特定のオブジェクトに付加することによって、データベースは要求されたタスクを確実に実行します。

参考資料

PowerDesigner の詳細については、PowerDesigner メイン・ウィンドウのツールバーから [Help] を選択して、次の 3 種類のマニュアルにアクセスしてください。『*PowerDesigner General Features Guide*』『*PowerDesigner PDM User's Guide*』『*PowerDesigner Report User's Guide*』

PowerDesigner のその他のチュートリアルについては、PowerDesigner を開いてください。[Help] メニューから、[Where to Start] － [PDM Getting Started] の順に選択します。これにより、『*Physical Data Model Getting Started*』にアクセスできます

データベース設計の詳細については、『ASA SQL ユーザーズ・ガイド』> 「データベースの設計」を参照してください。

第 20 章

InfoMaker を使用したレポート作成

この章の内容

この章は、InfoMaker のレポート作成について、その導入と InfoMaker 環境で効率的に作業するのに必要な基本的スキルを習得するための簡単なチュートリアルで構成されています。

InfoMaker について

InfoMaker は、レポート作成とデータのメンテナンスのための強力なツールです。InfoMaker を使用すると以下のオブジェクトを作成できます。

- データを表示するためのレポート
- データを表示および変更するためのフォーム
- レポートまたはフォームから自動的にデータを取得するクエリ
- 1 つのデータベース (または DBMS) から別のデータベースに導くパイプライン
- レポートとフォームをバンドルしてユーザに分配するアプリケーション

InfoMaker はさまざまな種類のデスクトップやサーバ・ベースのデータベースへのビルトイン接続を可能にします。グラフィック環境で作業できる InfoMaker では、SQL の知識がなくてもデータを処理できます。グラフィック環境でレポートやその他のオブジェクトを構築するときに InfoMaker がバックグラウンドですべての SQL 文を作成します。

InfoMaker の詳細については、[スタート] メニューから [プログラム] － [Sybase] － [InfoMaker 9.0] － [Online Help Files] の順に選択して、オンライン・ヘルプを参照してください。また、アプリケーション内で [Help] メニューにアクセスすると、マニュアル全文を参照することもできます。

レッスン 1：はじめに

この InfoMaker チュートリアルを使用するには、Adaptive Server Anywhere サンプル・データベースに接続する必要があります。SQL Anywhere Studio 9.0 サンプル・データベースに接続するには、データベース・プロファイルを作成する必要があります。

❖ InfoMaker を起動するには、次の手順に従います。

- [スタート] メニューから、[プログラム] － [Sybase] － [InfoMaker 9.0] － [InfoMaker] の順に選択します。

  InfoMaker のメイン・ウィンドウが表示されます。このウィンドウには、InfoMaker ペインタとオンライン・ヘルプにアクセスするためのボタンが付いた [PowerBar] があります。

❖ InfoMaker を使用して、SQL Anywhere Studio サンプル・データベースのデータベース・プロファイルを作成するには、次の手順に従います。

1. InfoMaker で [Database Profiles] ダイアログを開きます。

   [Tools] メニューから、[Database Profile] を選択します。

2. [ODB ODBC] を選択し、[New] をクリックします。

   [Database Profile Setup] ダイアログが表示されます。

3. [Connection] タブで、次の値を入力します。

   - **[Profile Name] Anywhere**
   - **[Data Source] ASA 9.0 Sample**
   - **[User ID] DBA**
   - **[Password] SQL**

4. [OK] をクリックします。

プロファイル Anywhere が、ODB ODBC の下にリストされます。

❖ **データ・ソースの作成後、Adaptive Server Anywhere サンプル・データベースに接続するには、次の手順に従います。**

1. [Database Profiles] ウィンドウが開いていない場合は、[Tools] メニューから [Database Profile] をクリックして、これを開きます。
2. ODBC グループから [Anywhere] を選択し、[Connect] をクリックします。

次の手順

次は、InfoMaker ウィザードを使ってレポートを作成します。

レッスン 2：基本レポートの作成

この項では、基本レポートを作成する方法、それをプレビューし、保存する方法、デザイン環境の設定を変更する方法について説明します。

基本レポートの作成

作成できるレポートの種類はたくさんあります。この項では、1 つのテーブルからレポートを作成する方法について説明します。

❖ レポートを作成するには、次の手順に従います。

1. [File] メニューから、[New] を選択します。
2. [New] ダイアログで、[Object] タブをクリックします。
3. [Tabular] アイコンをダブルクリックして、表形式を選択します。

   [Tabular Report Generator] ウィザードが表示されます。
4. [Quick Select] データ・ソースを選択し、[Retrieve on Preview] が選択されていることを確認します。[Next] をクリックします。

   [Quick Select] ダイアログが表示されます。ここでは、データベース・テーブルとそのテーブルのカラムの一部または全部を選択できます。
5. contact テーブルを選択します。
6. 次のカラムを選択します。last_name、first_name、title、phone、fax (場合によってはスクロールする必要があります)。

   選択したカラムが、ダイアログ下部にあるグリッドに移動します。このグリッドは、カラムの並べ替えや、ソートおよび選択基準の指定に使用できます。
7. [OK] をクリックします。

[Select Color and Border Settings] ダイアログが表示されます。

8 [Next] をクリックして、デフォルト設定を使用します。

指定事項の要約をまとめたダイアログが表示されます。

9 [Finish] をクリックします。

レポートのプレビュー

この項では、実際に出力するとどのように表示されるか調べるために、レポートをプレビューします。

InfoMaker のメイン・ウィンドウの一番上には、ツールバーがあります。そのツールバーの下に、残りの画面全体を占める [Preview] ウィンドウ枠が表示されます。[Preview] ウィンドウ枠の一番上にはタイトル・バーがあります。タイトル・バーの左側には、プレビューしているレポートの名前が表示されます。右側には [最大化] または [最小化] ボタンが表示され、[Preview] ウィンドウ枠のサイズを調整できます。

❖ レポートをプレビューするには、次の手順に従います。

1 [Preview] ウィンドウ枠の右上にある最大化アイコンをクリックします。

[Preview] ウィンドウ枠が全画面表示になります。こうすると、レポートのヘッダ情報とデータベースに関する情報も表示されます。

2 ルーラのオンとオフを切り替えるには、データのあるセクション内をクリックします。[File] メニューから、[Print Preview Rulers] を選択します。

3 [Preview] ウィンドウ枠を元のサイズと場所に戻すには、右上にある最小化アイコンをクリックします。

レポートの保存

❖ レポートを保存するには、次の手順に従います。

1 [File] メニューから、[Save] を選択します。

[Save Report] ダイアログが表示されます。

2 [Reports] フィールドに、「**contacts_by_jobrole**」と入力します。これがレポート名となります。

3 オプションとして、[Comments] ボックスにメモを入力すると、このレポートについて詳細を追加できます。

[Comments] ボックスをクリックし、「**This report shows my contacts grouped by job role.**」と入力します。

4 [OK] をクリックして、レポートを保存します。

デザイン環境の設定

この項では、コントロール、グリッド、ルーラに変更を加えます。

❖ デザイン環境を設定するには、次の手順に従います。

1 [Design] メニューから、[Options] を選択します。

[Report Options] ダイアログが表示されます。

2 次のオプションが選択されていることを確認します。Show Grid、Show Ruler、Show Edges、Retrieve on Preview。

3 [Snap to Grid] が選択されていないことを確認します。

4 オプションすべての説明を表示するには、[Help] をクリックします。

5 [OK] をクリックします。

次の手順

これで、レポートのレイアウトをカスタマイズする準備ができました。

レッスン 3：レポートの強化

この項では、以下の機能を含むレポートの強化方法について学習します。

- データのソート
- ヘッダとタイトルの作成およびフォーマッティング
- 日付、ページ番号、合計などの計算フィールドの追加

ソートとグループ化の定義

この項では、取引先の担当者を肩書きをもとにグループ化します。これを行うには、データを title でソートしてから、title によるグループ化を指定します。

最初に、ソート方法を定義します。

❖ 連絡先をソートするには、次の手順に従います。

1. [Rows] メニューから、[Sort] を選択します。

   [Specify Sort Columns] ダイアログが表示されます。

2. title を [Source Data] ボックスから [Columns] ボックスにドラッグします。
3. last_name と first_name を [Columns] ボックスにドラッグします。
4. デフォルトの昇順を受け入れるには、チェックボックスをオンのままにします。
5. [OK] をクリックします。

次に、title カラムを基準にしてグループを定義し、同じ title を持つ取引先の担当者すべてを同じグループに入れます。

❖ 連絡先をグループ化するには、次の手順に従います。

1 [Rows] メニューから、[Create Group] を選択します。

[Specify Group Columns] ダイアログが表示されます。

2 title を [Columns] ボックスにドラッグし、[OK] をクリックします。

グループ化はこれで完了です。[Preview] ウィンドウ枠を最大化すると、データがソートされているのが確認できます。グループ分けはまだ表示されません。

レポートの強化

この項では、コントロールの配置の変更、タイトルと日付の追加、ページ番号と合計数の追加などによってレポート機能を拡張します。

❖ グループのヘッダを調整するには、次の手順に従います。

1 マウス・ポインタを [Header Group Title] というグレーのバーに置きます。

ポインタが両方向矢印に変わります。

2 そのバーを 5 グリッド目盛分下にドラッグして、区域を広げます。

3 [Preview] ウィンドウ枠の下にある title カラムを、[Header Group] 区域の該当位置までドラッグします。

4 [title] カラムを選択したままの状態で、[StyleBar] の [B] と [I] をクリックします。

5 [Preview] ウィンドウ枠でレポートを確認します。

❖ ヘッダを配列するには、次の手順に従います。

1 [First Name] と [Last Name] の 2 つのテキスト・ボックスを [Header] 区域内の空きスペースにドラッグします。

2　[first_name] カラムと [last_name] カラムを [Detail] 区域内の右側の空きスペースにドラッグします。

3　[Edit] メニューから、[Select] － [Select Text] を選択します。

すべてのヘッダが選択されます。この区域は幅が狭いため、これらが選択されているかどうかわかりにくい場合があります。

4　[StyleBar] の [Left] ボタンをクリックします。ヘッダが太字で表示されてない場合は、[B] ボタンをクリックします。

5　[Preview] ウィンドウ枠でレポートを確認します。

❖ レポートにタイトルを追加するには、次の手順に従います。

1　[Header] というグレーのバーを約 10 グリッド目盛分下にドラッグします。

2　[Edit] メニューから、[Select] － [Select Text] を選択します。すべてのヘッダが選択されます。

3　すべてのヘッダをグレーのバー近くまでドラッグします。

4　[Insert] メニューから、[Control] － [Text] の順に選択します。

5　マウス・ポインタの先を、ページ最上部よりも 1 グリッド目盛下の [First Name] ボックスの上に移動します。1 回クリックします。ここが、title 用のテキスト・ボックスの位置になります。

6　「**My Contacts**」(取引先担当者) と入力します。

7　[StyleBar] のサイズのドロップダウン・リストから、[14] を選択します。

❖ レポートに日付を追加するには、次の手順に従います。

1　[Insert] メニューから、[Control] － [Today()] を選択します。

2 ポインタの先をレポートの左上に移動し、クリックします。これでレポートに日付が追加されます。

❖ レポートにページ番号を追加するには、次の手順に従います。

1 [Footer] というグレーのバーを約 4 グリッド目盛分下にドラッグします。必要であれば、スクロール・バーを使用してデザイン・ビューの一番下に空きスペースを作ります。[Footer] バーを下にドラッグすると、[Footer] 区域にページ番号用の空きスペースができます。

2 [Insert] メニューから、[Control] － [Page n of n] を選択します。

3 [Footer] 区域の中央より約 2 グリッド目盛分下にマウス・ポインタを置いて、クリックします。

4 [StyleBar] でサイズ [10] を選択します。

❖ last name の数をカウントして合計を追加するには、次の手順に従います。

1 [Summary] というグレーのバーを約 6 グリッド目盛分下にドラッグします。ここが合計数を入力するためのスペースになります。

2 [Insert] メニューから、[Control] － [Computed Field] を選択します。

3 [Summary] 区域の中央より約 4 グリッド目盛分下にマウス・ポインタを置いて、クリックします。

[Modify Expression] ダイアログが表示されます。

4 [Functions] ボックス内で、[Count(#x for all)] をクリックします。

5 [Columns] ボックス内で、[last_name] をクリックします。

6 [Verify] をクリックします。

式が有効であることを示すメッセージ (Expression is OK) が表示されます。[OK] をクリックします。

7　[OK] をクリックして、計算フィールドの式の定義を終了します。

❖ 合計をフォーマットするには、次の手順に従います。

1　[Insert] メニューから、[Control] － [Text] の順に選択します。

2　マウス・ポインタを計算フィールドの左側に置き、クリックします。

3　「**Total contacts:**」と入力します。

4　テキスト・ボックスをドラッグし、計算フィールドと並べます。

5　左マウス・ボタンを押したまま、テキスト・ボックスと計算フィールドのまわりを囲んでボタンを離します。テキスト・ボックスと計算フィールドが選択されます。

6　[StyleBar] でサイズ [10] を選択し、[B] と [I] をクリックします。

❖ レポートを印刷、保存、終了するには、次の手順に従います。

1　[File] メニューから、[Print Report] を選択します。

2　[Print] ダイアログで、[OK] をクリックします。

3　[File] メニューから、[Close] をクリックします。

4　変更の保存を確認するプロンプトが表示された場合は、[Yes] をクリックします。

まとめ

このチュートリアルでは、以下のことについて学びました。

- InfoMaker に接続する。
- 基本レポートを作成する。
- レポートをプレビューする。
- レポートをフォーマットする (データのソート、ヘッダ、タイトル、日付、ページ番号、合計の追加)。

参考資料

InfoMaker には、レポートのほかに、グラフ、クエリ、およびデータ入力フォームを作成する機能があります。

InfoMaker の詳細については、[スタート] － [プログラム] － [Sybase SQL Anywhere 9] － [InfoMaker 9.0] － [Online Help Files] の順に選択して、オンライン・ヘルプを参照してください。また、InfoMaker の [Help] メニューをクリックすると、マニュアル全文を参照することもできます。

第 21 章

Microsoft VisualBasic クイック・スタート

この章の内容

この章では、Adaptive Server Anywhere と Microsoft Visual Basic を使用して、簡単なデータベース・アプリケーションを開発する方法について説明します。

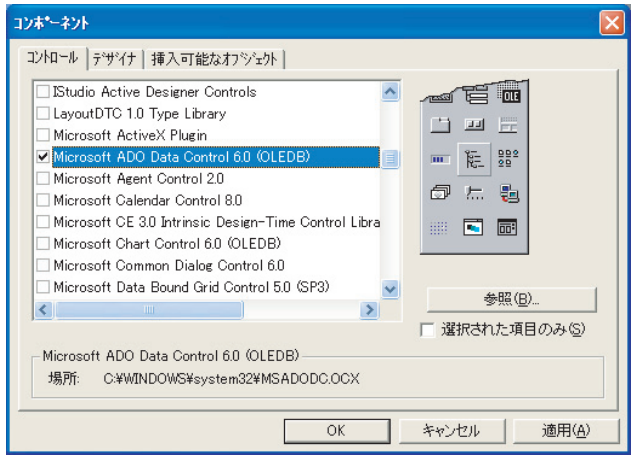
チュートリアル：Visual Basic アプリケーションの開発

この簡単なチュートリアルは、Visual Basic 6.0 を基にしています。完全なアプリケーションは、Visual Basic プロジェクトの *Samples¥ASA¥VBStarter¥ASAStarter.vbp* で参照できます。

Visual Basic は、複数のデータ・アクセス技術を提供しています。このチュートリアルでは、Microsoft ADO Data Control を Adaptive Server Anywhere の OLE DB プロバイダと一緒に使用して、Visual Basic から Adaptive Server Anywhere のサンプル・データベースにアクセスします。

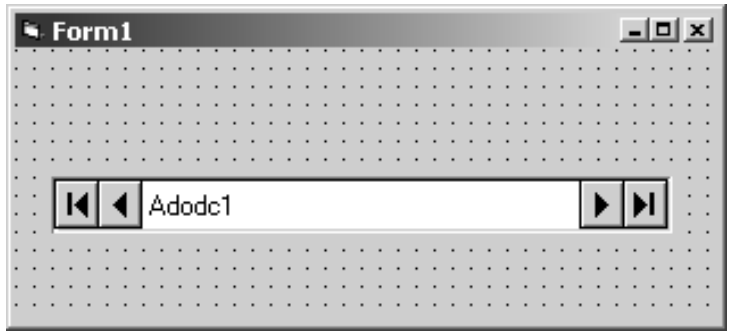
❖ Visual Basic でデータベース・アプリケーションを開発するには、次の手順に従います。

1. Visual Basic を起動して、[Standard Executable] プロジェクトを選択します。
2. Microsoft ADO Data Control 6.0 をツール・パレットに追加します。
   - [プロジェクト] メニューから、[コンポーネント] を選択します。
   - リストから、[Microsoft ADO Data Control 6.0] コンポーネントを選択します。



- [OK] をクリックして、コントロールをパレットに追加します。

3　次のように、ADO Data Control をフォームに追加します。



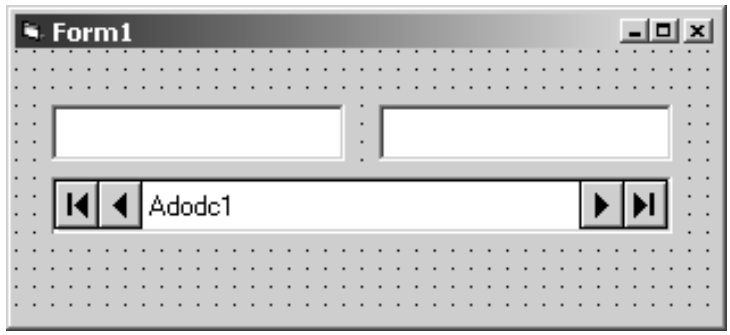
4　ADO Data Control の設定

プロパティ	値
ConnectionString	Provider=ASAPROV;DSN=ASA 9.0 Sample

プロパティ	値
CursorLocation	2 - asUseServer
CursorType	1 - adOpenKeyset
RecordSource	SELECT * FROM EMPLOYEE

ConnectionString は、Adaptive Server Anywhere OLE DB Provider (ASAProv) を使用して ASA 9.0 Sample データ・ソースに接続します。カーソルの設定では、クライアント側のカーソルではなく、Adaptive Server Anywhere のカーソルを利用します。

5 次のように、2 つのテキスト・ボックスをフォームに追加します。



6 テキスト・ボックスを ADO Data Control にバインドします。

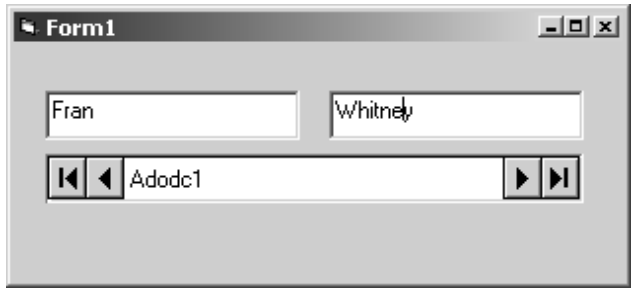
- それぞれの DataSource プロパティを「**Adodc1**」に設定します。
- 左側のテキスト・ボックスの DataField プロパティを「**emp_fname**」に設定します。これは、従業員の名を格納しているカラムです。
- 右側のテキスト・ボックスの DataField プロパティを「**emp_lname**」に設定します。これは、従業員の姓を格納しているカラムです。

7　プロジェクトを保存します。

8　サンプルを実行します。

- [Run] － [Start] を選択してアプリケーションを実行します。

  アプリケーションが Adaptive Server Anywhere サンプル・データベースに接続され、次のように最初の従業員の名前がテキスト・ボックスに表示されます。



- ADO Data Control のボタンを使用して、結果セットのローをスクロールします。

これで、Adaptive Server Anywhere と連携する簡単な Visual Basic アプリケーションを作成できました。

第 5 部　付録

付録には、SQL Anywhere Studio に関連する用語の解説が含まれています。

用語集

Adaptive Server Anywhere (ASA)

SQL Anywhere Studio のリレーショナル・データベース・サーバ・コンポーネントであり、主に、モバイル環境と埋め込み環境、または小規模および中規模のビジネス用のサーバとして使用されます。

DB 領域

データ用の領域をさらに作成する追加のデータベース・ファイルです。1 つのデータベースには 13 個までファイルを保管できます (初期ファイル 1 つと 12 の DB 領域)。各テーブルは、そのインデックスとともに、単一のデータベース・ファイルに含まれている必要があります。CREATE DBSPACE という SQL コマンドで、新しいファイルをデータベースに追加できます。

参照 :「データベース・ファイル」439 ページ。

DBA 権限

ユーザに、データベース内の管理作業を実行させるレベルのパーミッションです。DBA ユーザにはデフォルトで DBA 権限が与えられています。

参照 :「データベース管理者 (DBA)」439 ページ。

Embedded SQL

C プログラム用のプログラミング・インタフェースです。Adaptive Server Anywhere の embedded SQL は ANSI と IBM 規格に準拠して実装されています。

FILE

SQL Remote のレプリケーションでは、レプリケーション・メッセージのやりとりのために共有ファイルを使うメッセージ・システムのことです。これは (MAPI などの) 特定のメッセージ送信システムに頼らずにテストやインストールを行うのに便利です。

参照 :「レプリケーション」446 ページ、「MAPI」432 ページ。

grant オプション

他のユーザにパーミッションを許可できるレベルのパーミッションです。

InfoMaker

レポート作成とデータ管理用のツールです。洗練されたフォーム、レポート、グラフ、クロスタブ、テーブルを作成できます。また、これらを基本的な構成要素とするアプリケーションも作成できます。

Interactive SQL

データベース内のデータの変更や問い合わせ、データベース構造の修正ができる、Adaptive Server Anywhere のアプリケーションです。Interactive SQL では、SQL 文を入力するためのウィンドウ枠が表示されます。また、クエリの進捗情報や結果セットを表示するウィンドウ枠も表示されます。

JAR ファイル

Java アーカイブ・ファイルです。Java のアプリケーションで使用される 1 つ以上のパッケージの集合からなる圧縮ファイルのフォーマットです。Java プログラムをインストールしたり実行したりするのに必要なリソースが 1 つの圧縮ファイルにすべて収められています。

Java クラス

Java のコードの主要な構造単位です。これはプロシージャや変数の集まりで、すべてがある一定のカテゴリに関連しているためグループ化されたものです。

jConnect

JavaSoft JDBC 標準を Java で実装したものです。これにより、Java 開発者は多層／異機種環境でもネイティブなデータベース・アクセスができます。

参照：「JDBC」432 ページ。

JDBC

Java Database Connectivity の略です。Java アプリケーションからリレーショナル・データにアクセスすることを可能にする SQL 言語プログラミング・インタフェースです。

LTM

Log Transfer Manager の略です。「Replication Agent」434 ページを参照してください。

MAPI

Microsoft の Messaging Application Programming Interface の略です。Microsoft Mail など、普及しているいくつかの電子メール・システムで使用されているメッセージ・システムです。

Mobile Link クライアント

2 種類の Mobile Link クライアントがあります。Adaptive Server Anywhere リモート・データベース用の Mobile Link クライアントは、dbmlsync コマンド・ライン・ユーティリティです。Ultra Light リモート・データベース用の Mobile Link クライアントは、Ultra Light ランタイム・ライブラリに組み込まれています。

Mobile Link ユーザ

Mobile Link ユーザは、同期システムの Mobile Link リモート・データベースをユニークに識別する名前です。Mobile Link 同期サーバに接続したときに、クライアントによってこの名前と、オプションで関連したパスワードが指定されます。Mobile Link ユーザ名はデータベース・ユーザ名から完全に独立しています。

Mobile Link

Ultra Light と Adaptive Server Anywhere の各データベースと、Sybase や他のベンダによって開発された多数の業界標準 SQL データベース管理システムを同期するために設計された、セッション・ベースの同期テクノロジです。

参照：「Ultra Light」435 ページ。

NetWare

Novell が定義したネットワーク・オペレーティング・システムで、広く使われています。通常、NetWare では IPX/SPX プロトコルが使用されますが、TCP/IP プロトコルを使うこともできます。

ODBC アドミニストレータ

Windows オペレーティング・システムに付属している Microsoft のプログラムです。ODBC データ・ソースの設定に使用します。

ODBC データ・ソース

ユーザが ODBC からアクセスするデータと、そのデータにアクセスするために必要な情報の仕様です。

ODBC

Open Database Connectivity の略です。データベース管理システムに対する Windows の標準的なインタフェースです。ODBC は、Adaptive Server Anywhere がサポートするインタフェースの 1 つです。

PowerDesigner

データベース・モデリング・アプリケーションです。これを使用すると、データベースやデータ・ウェアハウスの設計に対する構造的なアプローチが可能となります。

PowerDynamo

データベースとリンクした Web アプリケーションを構築したり管理したりするための Sybase 製品です。

PowerJ

Java アプリケーション開発に使用する Sybase 製品です。

REMOTE パーミッション

SQL Remote レプリケーションでは、レプリケーション・メッセージをパブリッシュするデータベースとレプリケーション・メッセージを交換できるできるパーミッションを指します。REMOTE パーミッションを付与されたユーザはリモート・ユーザになります。メッセージ・タイプ、該当するリモート・アドレス、レプリケーションの頻度を指定します。広い意味では、REMOTE パーミッションとは SQL

Remote のレプリケーションに関わるすべてのユーザ (たとえば統合データベースのパブリッシャやリモート・データベースのパブリッシャ) のことも指します。

参照 :「レプリケーション」446 ページ。

Replication Agent

Replication Server では Log Transfer Manager (LTM) とも呼ばれるプログラムです。データベースのトランザクション・ログを読み込み、コミットされた変更を Replication Server に送信します。

Replication Server

Adaptive Server Anywhere と Adaptive Server Enterprise で動作する、Sybase による接続ベースのレプリケーション・テクノロジです。Replication Server は、比較的少数のデータベース間でほぼリアルタイムのレプリケーションを行うことを目的に設計されています。

SQL Remote

統合データベースとリモート・データベース間で双方向レプリケーションを行うための、メッセージベースのレプリケーション・テクノロジです。統合データベースには、Adaptive Server Anywhere か Adaptive Server Enterprise を使用します。リモート・データベースには Adaptive Server Anywhere を使用します。

SQL 文

DBMS に命令を渡すために設計された、SQL キーワードを含む文字列です。

参照 :「スキーマ」438 ページ、「SQL」434 ページ、「データベース管理システム (DBMS)」439 ページ。

SQL

リレーショナル・データベースとの通信に使用される言語です。SQL は ANSI により標準が定義されており、その最新版は SQL-99 (SQL3 とも言います) です。SQL は、公認されてはいませんが、Structured Query Language の略です。

Sybase Central

Adaptive Server Anywhere データベースのさまざまな設定、プロパティ、ユーティリティを使用できる、グラフィカル・ユーザ・インタフェースを持つデータベース管理ツールです。Mobile Link などの他の iAnywhere 製品を管理する場合にも使用できます。

SYS

システム・オブジェクトの大半を所有する特別なユーザです。一般のユーザは SYS でログインできません。

Ultra Light

Adaptive Server Anywhere データベース用の配備テクノロジです。小型デバイス、モバイル・デバイス、埋め込みデバイスを対象としています。対象となるプラットフォームとして、携帯電話、ポケットベル、パーソナル・オーガナイザなどが挙げられます。

Windows CE

Microsoft がモバイル・デバイス用に開発したオペレーティング・システムのファミリです。

Windows

Windows 95、Windows 98、Windows Me、Windows CE、Windows NT、Windows 2000、Windows XP といった Microsoft Windows オペレーティング・システムのファミリのことです。

アーティクル

SQL Remote または Mobile Link では、テーブル全体もしくはテーブル内のカラムとローのサブセットを表すデータベース・オブジェクトを指します。アーティクルの集合がパブリケーションです。

参照：「レプリケーション」446 ページ、「パブリケーション」442 ページ。

アップロード

同期中に、リモート・データベースから統合データベースにデータが転送される段階です。

アトミックなトランザクション

トランザクション内のコマンドのいくつかが正しく処理されず、変更が取り消されるか、ロールバックされる状態です。

アンロード

データベースをアンロードすると、データベースの構造かデータ、またはその両方がテキスト・ファイルにエクスポートされます (構造は SQL コマンド・ファイルに、データはカンマ区切りの ASCII ファイルにエクスポートされます)。データベースのアンロードには、アンロード・ユーティリティを使用します。

また、UNLOAD 文を使って、データから抜粋した部分だけをアンロードできます。

インクリメンタル・バックアップ

トランザクション・ログ専用のバックアップです。通常、フル・バックアップとフル・バックアップの間に使用します。

参照：「トランザクション・ログ」441 ページ。

インデックス

ベース・テーブルにある 1 つ以上のカラムに関連付けられた、キーとポインタのソートされたセットです。テーブルの 1 つ以上のカラムにインデックスが設定されていると、パフォーマンスが向上します。

オブジェクト・ツリー

Sybase Central では、データベース・オブジェクトの階層を指します。オブジェクト・ツリーの最上位には、現在使用しているバージョンの Sybase Central がサポートするすべての製品が表示されます。それぞれの製品を拡張表示すると、オブジェクトの下位ツリーが表示されます。

参照：「Sybase Central」434 ページ。

カーソル

結果セットへの関連付けに名前を付けたもので、プログラミング・インタフェースからローにアクセスしたり更新したりするときに使用します。Adaptive Server Anywhere では、カーソルはクエリ結果内で前方や後方への移動をサポートします。カーソルは、カーソル結果セット (通常 SELECT 文で定義される) とカーソル位置の 2 つの部分から構成されます。

参照：「カーソル結果セット」436 ページ、「カーソル位置」436 ページ。

カーソル位置

カーソル結果セット内の 1 つのローを指すポインタ。

参照：「カーソル」436 ページ、「カーソル結果セット」436 ページ。

カーソル結果セット

カーソルに関連付けられたクエリから生成されるローのセットです。

参照：「カーソル」436 ページ、「カーソル位置」436 ページ。

クエリ

データベースのデータにアクセスしたり、そのデータを操作したりする SQL 文や SQL 文のグループです。

参照：「SQL」434 ページ。

クライアント／サーバ

あるアプリケーション (クライアント) が別のアプリケーション (サーバ) に対して情報を送受信するソフトウェア･アーキテクチャのことです。通常この 2 種類のアプリケーションは、ネットワークに接続された異なるコンピュータ上で実行されます。

グローバル・テンポラリ・テーブル

明示的に削除されるまでデータ定義がすべてのユーザに表示されるテンポラリ・テーブルです。グローバル・テンポラリ・テーブルを使用すると、各ユーザが、1 つのテーブルのまったく同じインスタンスを開くことができます。デフォルトでは、コミット時にローが削除され、接続終了時にもローが削除されます。

参照：「テンポラリ・テーブル」440 ページ、「ローカル・テンポラリ・テーブル」446 ページ。

コマンド・ファイル

SQL 文で構成されたテキスト・ファイルです。コマンド・ファイルは手動で作成できますが、データベース・ユーティリティによって自動的に作成することもできます。たとえば、dbunload ユーティリティを使うと、指定されたデータベースの再構築に必要な SQL 文で構成されたコマンド・ファイルを作成できます。

コンボイ

1 つを除くすべてのデータベース接続が共有リソースを待機している状態です。

サーバ起動同期

統合データベースから Mobile Link 同期をプログラムによって開始する方法です。

サービス

Windows オペレーティング・システムで、アプリケーションを実行するユーザ ID がログオンしていないときにアプリケーションを実行する方法です。

サブクエリ

別の SELECT 文、INSERT 文、UPDATE 文、DELETE 文、または別のサブクエリの中にネストされた SELECT 文です。

相関とネストの 2 種類のサブクエリがあります。

サブスクリプション

SQL Remote レプリケーションでは、パブリケーションとリモート・ユーザ間のリンクのことで、これによりリモート・ユーザはそのパブリケーションの更新内容を統合データベースとの間で交換できます。

Mobile Link 同期では、同期サブスクリプションはパブリケーションと Mobile Link ユーザ間のクライアント・データベース内のリンクであり、そのパブリケーションが記述したデータの同期を可能にします。

参照：「パブリケーション」442 ページ、「リモート・ユーザ」445 ページ、「Mobile Link ユーザ」433 ページ。

システム・オブジェクト

SYS または dbo が所有するデータベース・オブジェクトです。

システム・テーブル

SYS または dbo が所有するテーブルです。メタデータが格納されています。システム・テーブル (データ辞書テーブルとしても知られています) はデータベース・サーバが作成し管理します。

システム・ビュー

すべてのデータベースに含まれているビューです。システム・テーブル内に格納されている情報をわかりやすいフォーマットで示します。

ジョイン

指定されたカラムの値を比較することによって 2 つ以上のテーブルにあるローをリンクする、リレーショナル・システムでの基本的な操作です。

ジョイン・タイプ

Adaptive Server Anywhere では、クロス・ジョイン、キー・ジョイン、ナチュラル・ジョイン、ON 句を使ったジョインの 4 種類のジョインが使用されます。

参照：「ジョイン」438 ページ。

ジョイン条件

ジョインの結果に影響を及ぼす制限です。ジョイン条件は、JOIN の直後に ON 句か WHERE 句を挿入して指定します。ナチュラル・ジョインとキー・ジョインについては、Adaptive Server Anywhere がジョイン条件を生成します。

参照：「ジョイン」438 ページ、「生成されたジョイン条件」450 ページ。

スキーマ

テーブル、カラム、インデックス、それらの関係などを含んだデータベース構造です。

スクリプト

Mobile Link では、Mobile Link のイベントを処理するために記述されたコードです。スクリプトは、業務上の要求に適合するように、データ交換をプログラム的に制御します。

ストアド・プロシージャ

一連の SQL 命令から構成されたプログラムです。データベースに格納され、特定のタスクを実行するときに使用します。

セッション・ベースの同期

統合データベースとリモート・データベースの両方でデータ表現の一貫性が保たれる同期です。Mobile Link はセッション・ベースです。

ダウンロード

同期中に、統合データベースからリモート・データベースにデータが転送される段階です。

チェックポイント

データベースに加えたすべての変更内容がデータベース・ファイルに保存されるポイントです。通常、コミットされた変更内容はトランザクション・ログだけに保存されます。

データベース

プライマリ・キーと外部キーによって関連付けられているテーブルの集合です。これらのテーブルでデータベース内の情報が保管されます。また、テーブルとキーによってデータベースの構造が定義されます。データベース管理システムでこの情報にアクセスします。

参照：「外部キー」448 ページ、「プライマリ・キー」443 ページ、「データベース管理システム (DBMS)」439 ページ、「リレーショナル・データベース管理システム (RDBMS)」446 ページ。

データベース・オブジェクト

情報を保管したり受け取ったりするデータベース・コンポーネントです。テーブル、インデックス、ビュー、プロシージャ、トリガはデータベース・オブジェクトです。

データベース・サーバ

データベース内にある情報へのすべてのアクセスを規制するコンピュータ・プログラムです。Adaptive Server Anywhere には、ネットワーク・サーバとパーソナル・サーバの 2 種類のサーバがあります。

データベース・ファイル

データベースは 1 つまたは複数のデータベース・ファイルに保持されます。まず、初期ファイルがあり、それに続くファイルは DB 領域と呼ばれます。各テーブルは、それに関連付けられているインデックスとともに、単一のデータベース・ファイルに含まれている必要があります。

参照：「DB 領域」431 ページ。

データベース管理システム (DBMS)

データベースを作成したり使用したりするためのプログラムの集合です。

参照：「リレーショナル・データベース管理システム (RDBMS)」446 ページ。

データベース管理者 (DBA)

データベースの管理に必要なパーミッションを持つユーザです。DBA は、データベース・スキーマのあらゆる変更や、ユーザやユーザ・グループの管理に対して、全般的な責任を負います。データベース管理者のロールはデータベース内に自動的に作成されます。その場合、ユーザ ID は DBA であり、パスワードは SQL です。

データベース所有者 (dbo)

SYS が所有しないシステム・オブジェクトを所有する特別なユーザです。

参照：「データベース管理者 (DBA)」439 ページ、「SYS」434 ページ。

データベース接続

クライアント・アプリケーションとデータベース間の通信チャネルです。接続を確立するためには有効なユーザ ID とパスワードが必要です。接続中に実行できるアクションは、そのユーザ ID に付与された権限によって決まります。

データベース名

サーバがデータベースをロードするとき、そのデータベースに指定する名前です。デフォルトのデータベース名は、初期データベース・ファイルのルート名です。

参照：「データベース・ファイル」439 ページ。

データ型

CHAR や NUMERIC などのデータのフォーマットです。ANSI SQL 規格では、サイズ、文字セット、照合に関する制限もデータ型に組み込めます。

参照：「ユーザ定義データ型」444 ページ。

データ操作言語 (DML)

データベースの内容を取り出したり更新したりするための SQL 文のサブセットです。

データ定義言語 (DDL)

データベースの構造をモデル化するときに使う SQL 文のサブセットです。DDL 文は、ユーザなどのデータベース・オブジェクトを作成、修正、削除できます。

デッドロック

先へ進めない場所に一連のトランザクションが到達する状態です。

テンポラリ・テーブル

データを一時的に保管するために作成されるテーブルです。グローバルとローカルの 2 種類があります。

参照：「ローカル・テンポラリ・テーブル」446 ページ、「グローバル・テンポラリ・テーブル」436 ページ。

トランザクション

作業の論理単位を構成する一連の SQL 文です。1 つのトランザクションは完全に処理されるかまったく処理されないかのどちらかです。Adaptive Server Anywhere は、ロック機能のあるトランザクション処理をサポートしているので、複数のトランザクションが同時にデータベースにアクセスしてもデータを壊すことがありません。トランザクションは、データに加えた変更を永久なものにする COMMIT 文か、トランザクション中に加えられたすべての変更を元に戻す ROLLBACK 文のいずれかで終了します。

トランザクション・ログ

データベースに対するすべての変更内容が、変更された順に格納されるファイルです。パフォーマンスを向上させ、データベース・ファイルが破損した場合でもデータをリカバリできます。最適な結果を得るために、データベース・ファイルとは別のデバイスに保存してください。

トランザクション・ログ・ミラー

オプションで設定できる、トランザクション・ログ・ファイルの完全なコピーのことで、トランザクション・ログと同時に管理されます。データベースの変更がトランザクション・ログへ書き込まれると、トランザクション・ログ・ミラーにも同じ内容が書き込まれます。

ミラー・ファイルは、トランザクション・ログとは別のデバイスに置いてください。一方のデバイスに障害が発生しても、もう一方のログにリカバリのためのデータが確保されます。

参照：「トランザクション・ログ」441 ページ。

トランザクション単位の整合性

Mobile Link で、同期システム全体でのトランザクションの管理を保証します。トランザクション全体が同期されるか、トランザクション全体がまったく同期されないかのどちらかになります。

トリガ

データを修正するクエリをユーザが実行すると、自動的に実行されるストアド・プロシージャの特別な形式です。

参照：「ロー・レベルのトリガ」446 ページ、「文レベルのトリガ」452 ページ、「競合トリガ」449 ページ、「整合性」450 ページ。

ネットワーク・サーバ

共通ネットワークを共有するコンピュータからの接続を受け入れるデータベース・サーバです。

参照：「パーソナル・サーバ」441 ページ。

パーソナル・サーバ

クライアント・アプリケーションが実行されているコンピュータと同じマシンで実行されているデータベース・サーバです。パーソナル・データベース・サーバは、単一のコンピュータ上で単一のユーザが使用しますが、そのユーザからの複数の同時接続をサポートできます。

パススルー

SQL Remote レプリケーションでは、統合データベースのパブリッシャが、SQL 文を使ってリモート・データベースを直接変更できるモードのことを指します。パススルーは特定のリモートに対して設定されます。通常のパススルー・モードでは、統合データベースでのす

べての変更が指定されたリモート・データベースに反映されます。パススルーのみのモードでは、変更内容はリモード・データベースに反映されますが、統合データベースには反映されません。

パッケージ

Java では、それぞれが互いに関連のあるクラスの集合を指します。

パフォーマンス統計値

データベース・システムのパフォーマンスを反映する値です。たとえば、CURRREAD 統計値は、エンジンが要求したファイル読み込みのうち、現在まだ完了していないものの数を表します。

パブリケーション

SQL Remote または Mobile Link では、レプリケートされたデータを識別するデータベース・オブジェクトを指します。Mobile Link では、クライアント上にのみ存在します。1 つのパブリケーションは複数のアーティクルから構成されています。各パブリケーションに対して加えられた変更は、そのパブリケーションのすべてのサブスクライバに定期的にレプリケートされます。SQL Remote ユーザは、パブリケーションに対してサブスクリプションを作成することによって、パブリケーションを受信できます。Mobile Link ユーザは、パブリケーションに対して同期サブスクリプションを作成することによって、パブリケーションを同期できます。

参照：「レプリケーション」446 ページ、「アーティクル」435 ページ、「パブリケーションの更新」442 ページ。

パブリケーションの更新

SQL Remote レプリケーションでは、単一のデータベース内の 1 つまたは複数のパブリケーションに対して加えられた変更のリストを指します。パブリケーションの更新は、レプリケーション・メッセージの一部として定期的にリモート・データベースへ送られます。

参照：「レプリケーション」446 ページ、「パブリケーション」442 ページ。

パブリッシャ

SQL Remote レプリケーションでは、レプリケートできる他のデータベースとレプリケーション・メッセージを交換できるデータベースの単一ユーザを指します。

参照：「レプリケーション」446 ページ。

ビジネス・ルール

実世界の要求に基づくガイドラインです。通常ビジネス・ルールは、検査制約、ユーザ定義データ型、適切なトランザクションの使用により実装されます。

参照：「制約」450 ページ、「ユーザ定義データ型」444 ページ。

ビュー

データベースにオブジェクトとして格納される SELECT 文です。ビューを使用すると、ユーザは 1 つまたは複数のテーブルのローやカラムのサブセットを参照できます。ユーザが特定のテーブルやテーブルの組み合わせのビューを使うたびに、テーブルに保持されているデータから再計算されます。ビューは、セキュリティの目的に有用です。またデータベース情報の表示を調整して、データへのアクセスが簡単になるようにする場合も役立ちます。

ファイルベースのダウンロード

Mobile Link では、ダウンロードがファイルとして配布されるデータの同期方法であり、同期変更のオフライン配布を可能にします。

フェールオーバ

アクティブなサーバ、システム、またはネットワークで障害や予定外の停止が発生したときに、冗長な (スタンバイ) サーバ、システム、またはネットワークに切り替えることです。フェールオーバは自動的に実行され、多くの場合、連続的な可用性を提供するシステムに組み込まれています。

プライマリ・キー

テーブル内のすべてのローをユニークに識別する値を持つカラムまたはカラムのリストです。

「外部キー」448 ページを参照してください。

プライマリ・キー制約

プライマリ・キーのカラムに対する一意性制約です。テーブルにはプライマリ・キー制約を 1 つしか設定できません。

参照 :「制約」450 ページ、「検査制約」449 ページ、「外部キー制約」448 ページ、「一意性制約」448 ページ、「整合性」450 ページ。

プライマリ・テーブル

外部キー関係でプライマリ・キーを含むテーブルです。

プラグイン・モジュール

Sybase Central で、製品にアクセスしたり管理したりする方法です。プラグインは、通常、インストールすると Sybase Central にもインストールされ、自動的に登録されます。プラグインは、多くの場合、Sybase Central のメイン・ウィンドウに最上位のコンテナとして、その製品名 (たとえば Adaptive Server Anywhere) で表示されます。

参照 :「Sybase Central」434 ページ。

フル・バックアップ

データベース全体をバックアップすることです。オプションでトランザクション・ログのバックアップも可能です。フル・バックアップには、データベース内のすべての情報が含まれるため、システム障害やメディア障害が発生した場合に保護できます。

参照：「インクリメンタル・バックアップ」435 ページ。

プロキシ・テーブル

メタデータを含むローカル・テーブルです。リモート・データベース・サーバのテーブルに、ローカル・テーブルであるかのようにアクセスするときに使用します。

参照：「メタデータ」444 ページ。

ベース・テーブル

データを格納する永久テーブルです。テーブルは、テンポラリ・テーブルやビューと区別するために、「**ベース・テーブル**」と呼ばれることがあります。

参照：「テンポラリ・テーブル」440 ページ、「ビュー」443 ページ。

メタデータ

データについて説明したデータです。メタデータは、他のデータの特質と内容について記述しています。

参照：「スキーマ」438 ページ。

メッセージ・システム

SQL Remote のレプリケーションでは、統合データベースとリモート・データベースの間でのメッセージのやりとりに使用するプロトコルのことです。Adaptive Server Anywhere では、FILE、MAPI、FTP、SMTP、VIM のメッセージ・システムがサポートされています

参照：「レプリケーション」446 ページ、「FILE」431 ページ、「MAPI」432 ページ。

メッセージ・タイプ

SQL Remote のレプリケーションでは、リモート・ユーザと統合データベースのパブリッシャとの通信方法を指定するデータベース・オブジェクトのことを指します。統合データベースには、複数のメッセージ・タイプが定義されていることがあります。これによって、リモート・ユーザはさまざまなメッセージ・システムを使って統合データベースと通信できることになります。

参照：「レプリケーション」446 ページ、「統合データベース」451 ページ、「MAPI」432 ページ。

ユーザ定義データ型

基本データ型、およびオプションでデフォルト値、CHECK 条件、null 入力属性を指定するためにユーザが作成するデータ型です。ユーザ定義データ型 (ユーザ定義ドメインとも呼ばれます) をカラムに適用すると、データベース全体の一貫性を確保できます。

参照：「データ型」440 ページ。

ライト・ファイル

読み取り専用データベースへの変更内容を記録するために使用するファイルです。圧縮データベースでよく使用されます。

参照：「圧縮データベース・ファイル」448 ページ。

リダイレクタ

クライアントと Mobile Link 同期サーバ間で要求と応答をルート指定する Web サーバ・プラグインです。このプラグインによって、負荷分散メカニズムとフェールオーバ・メカニズムも実装されます。

リファレンス・データベース

Mobile Link で、Ultra Light クライアントの開発に使用される Adaptive Server Anywhere データベースです。開発中は、1 つの Adaptive Server Anywhere データベースをリファレンス・データベースとしても統合データベースとしても使用できます。他の製品によって作成されたデータベースは、リファレンス・データベースとして使用できません。

リモート DBA 権限

SQL Remote では、Message Agent で必要なパーミッションのレベルを指します。Mobile Link では、Adaptive Server Anywhere 同期クライアント (*dbmlsync*) で必要なパーミッションのレベルを指します。Message Agent または同期クライアントがこの権限を持つユーザとして接続した場合、DBA のフル・アクセス権が与えられます。このユーザ ID は、Message Agent または同期クライアントから接続しない場合にはパーミッションは追加されません。

参照：「DBA 権限」431 ページ。

リモート・データベース

SQL Remote レプリケーションまたは Mobile Link 同期では、統合データベースとデータを交換するデータベースを指します。リモート・データベースは、統合データベース内のすべてまたは一部のデータを共有できます。

参照：「レプリケーション」446 ページ、「統合データベース」451 ページ。

リモート・ユーザ

SQL Remote レプリケーションでは、REMOTE パーミッションを付与され、レプリケーションの設定時に特定のリモート・データベースに関連付けられている統合データベースのデータベース・ユーザを指します。リモート・ユーザを作成するには、通常のデータベース・ユーザに REMOTE パーミッションを付与します。そうすることによって、特定のリモート・データベースが存在するかどうかを識別するだけでなく、その特定のリモート・サイトと通信するときに使うメッセージ・タイプとアドレスも指定します。

リモート・データベースが統合データベースからの抽出によって作成されると、統合データベース内の各リモート・ユーザは、ある特定のリモート・データベースのデータのパブリッシャになります。

参照：「SQL Remote」434 ページ、「統合データベース」451 ページ、「パブリッシャ」442 ページ。

リレーショナル・データベース管理システム (RDBMS)

関連するテーブルの形式でデータを格納するデータベース管理システムです。

参照：「データベース管理システム (DBMS)」439 ページ。

レプリケーション

物理的に異なるデータベース間でデータを共有することです。Sybase では、Mobile Link、SQL Remote、Replication Server の 3 種類のレプリケーション・テクノロジを提供しています。

レプリケーション・メッセージ

SQL Remote または Replication Server では、パブリッシュするデータベースとサブスクリプションを作成するデータベース間で送信される通信内容を指します。メッセージにはデータを含み、レプリケーション・システムで必要なパススルー文、情報があります。

参照：「パススルー」441 ページ、「レプリケーション」446 ページ、「パブリケーションの更新」442 ページ。

レプリケーションの頻度

SQL Remote レプリケーションでは、リモート・ユーザに対する設定の 1 つで、パブリッシャの Message Agent がレプリケーション・メッセージを他のリモート・ユーザに送信する頻度を定義します。

参照：「レプリケーション」446 ページ、「リモート・ユーザ」445 ページ。

ロー・レベルのトリガ

変更されているローごとに一回実行するトリガです。

参照：「トリガ」441 ページ、「文レベルのトリガ」452 ページ。

ローカル・テンポラリ・テーブル

複合文を実行する間だけ存在したり、接続が終了するまで存在したりするテンポラリ・テーブルです。データのセットを 1 回だけロードする必要がある場合にローカル・テンポラリ・テーブルが便利です。デフォルトでは、COMMIT を実行するとローが削除されます。

参照：「テンポラリ・テーブル」440 ページ、「グローバル・テンポラリ・テーブル」436 ページ。

ロール

概念データベース・モデルで、ある視点からの関係を説明する動詞または句を指します。各関係は 2 つのロールを使用して表すことができます。"contains (A は B を含む)" や "is a member of (B は A のメンバー)" などのロールがあります。

ロールバック・ログ

コミットされていない各トランザクションの最中に行われた変更のレコードです。ROLLBACK 要求やシステム障害が発生した場合、コミットされていないトランザクションはデータベースから破棄され、データベースは前の状態に戻ります。各トランザクションにはそれぞれロールバック・ログが作成されます。このログは、トランザクションが完了すると削除されます。

参照：「トランザクション」440 ページ。

ロール名

外部キーの名前です。この外部キーがロール名と呼ばれるのは、外部テーブルとプライマリ・テーブル間の関係に名前を指定するためです。デフォルトでは、テーブル名がロール名になります。ただし、別の外部キーがそのテーブル名を使用している場合、デフォルトのロール名はテーブル名に 3 桁のユニークな数字を付けたものになります。名前は独自に作成もできます。

参照：「外部キー」448 ページ。

ログ・ファイル

Adaptive Server Anywhere によって管理されているトランザクションのログです。ログ・ファイルを使用すると、システム障害やメディア障害が発生してもデータベースを回復させることができます。また、データベースのパフォーマンスを向上させたり、SQL Remote を使用してデータをレプリケートしたりする場合にも使用できます。

参照：「トランザクション・ログ」441 ページ、「トランザクション・ログ・ミラー」441 ページ、「フル・バックアップ」443 ページ。

ロック

複数のトランザクションを同時に実行しているときにデータの整合性を保護する同時制御メカニズムです。Adaptive Server Anywhere では、2 つの接続によって同じデータが同時に変更されないようにするために、また変更処理の最中に他の接続によってデータが読み込まれないようにするために、自動的にロックが適用されます。

ロックの制御は、独立性レベルを設定して行います。

参照：「独立性レベル」452 ページ、「同時性 (同時実行性)」452 ページ、「整合性」450 ページ。

ワーク・テーブル

クエリの最適化の最中に中間結果を保管する内部保管領域です。

圧縮データベース・ファイル

dbshrink ユーティリティを使って、物理的なサイズを小さくしたデータベース・ファイルです。圧縮データベースは読み込み専用です。圧縮データベース・ファイルに変更を加える場合は、関連付けられているライト・ファイルを使う必要があります。圧縮されたデータベース・ファイルを通常のサイズに戻すには dbexpand ユーティリティを使用します。

一意性制約

null 以外のすべての値が重複しないことを要求するカラムまたはカラムのセットに対する制限です。テーブルには複数の一意性制約を指定できます。

参照：「外部キー制約」448 ページ、「プライマリ・キー制約」443 ページ、「制約」450 ページ。

外部キー

別のテーブルにあるプライマリ・キーの値を複製する、テーブルの 1 つ以上のカラムです。テーブル間の関係は、外部キーによって確立されます。

参照：「プライマリ・キー」443 ページ、「外部テーブル」448 ページ。

外部キー制約

カラムまたはカラムのセットに対する制約で、テーブルのデータが別のテーブルのデータとどのように関係しているかを指定するものです。カラムのセットに外部キー制約を加えると、それらのカラムが外部キーになります。

参照：「制約」450 ページ、「検査制約」449 ページ、「プライマリ・キー制約」443 ページ、「一意性制約」448 ページ。

外部ジョイン

テーブル内のすべてのローを保護するジョインです。Adaptive Server Anywhere では、左外部ジョイン、右外部ジョイン、全外部ジョインがサポートされています。左外部ジョインは JOIN 演算子の左側にあるテーブルのローを保護し、右側にあるテーブルのローがジョイン条件を満たさない場合には null を返します。全外部ジョインは両方のテーブルに含まれるすべてのローを保護します。

参照：「ジョイン」438 ページ、「内部ジョイン」452 ページ。

外部テーブル

外部キーを持つテーブルです。

参照：「外部キー」448 ページ。

外部ログイン

リモート・サーバとの通信に使用される代替のログイン名とパスワードです。デフォルトでは、Adaptive Server Anywhere は、クライアントに代わってリモート・サーバに接続するときは、常にそのクライアントの名前とパスワードを使用します。外部ログインを作成することによって、このデフォルトを上書きできます。外部ログインは、リモート・サーバと通信するときに使用する代替のログイン名とパスワードです。

競合

リソースについて対立する動作のことです。たとえば、データベース用語では、複数のユーザがデータベースの同じローを編集しようとした場合、そのローの編集権についての競合が発生します。

競合トリガ

SQL Remote レプリケーションでは、更新の競合が検出されると、更新が実行される前に起動するトリガを指します。競合トリガは、UPDATE 文の VERIFY 句の値がデータベースの現在値と一致しなかった場合に起動されます。

参照：「レプリケーション」446 ページ、「トリガ」441 ページ。

検査制約

指定された条件をカラムやカラムのセットに課す制約です。

参照：「制約」450 ページ、「外部キー制約」448 ページ、「プライマリ・キー制約」443 ページ、「一意性制約」448 ページ。

検証

データベース、テーブル、またはインデックスについて、特定のタイプのファイル破損をテストすることです。

参照整合性

データの整合性、特に異なるテーブルのプライマリ・キー値と外部キー値との関係を管理する規則を厳守することです。参照整合性を備えるには、それぞれの外部キーの値が、参照テーブルにあるローのプライマリ・キー値に対応するようにします。

参照：「プライマリ・キー」443 ページ、「外部キー」448 ページ。

識別子

テーブルやカラムなどのデータベース・オブジェクトを参照するときに使う文字列です。A ～ Z、a ～ z、0 ～ 9、アンダースコア (_)、アットマーク (@)、シャープ記号 (#)、ドル記号 ($) のうち、任意の文字を識別子として使用できます。

述部

条件式です。オプションで論理演算子 AND や OR と組み合わせて、WHERE 句または HAVING 句に条件のセットを作成します。SQL では、unknown と評価される述部が false と解釈されます。

照合

データベース内のテキストのプロパティを定義する文字セットとソート順の組み合わせのことです。Adaptive Server Anywhere データベースでは、サーバを実行しているオペレーティング・システムと言語によって、デフォルトの照合が決まります。たとえば、英語版 Windows システムのデフォルトの照合は 1252LATIN1 です。照合は、照合順とも呼ばれ、文字列の比較とソートに使用します。

制約

テーブルやカラムなど、特定のデータベース・オブジェクトに含まれた値に関する制約です。たとえば、一意性制約があるカラム内の値は、すべて異なっている必要があります。テーブルに、そのテーブルの情報と他のテーブルのデータがどのように関係しているのかを指定する外部キー制約が設定されていることもあります。

参照：「検査制約」449 ページ、「外部キー制約」448 ページ、「プライマリ・キー制約」443 ページ、「一意性制約」448 ページ。

整合性

データが適切かつ正確であり、データベースの関係構造が保たれていることを保証する規則を厳守することです。

参照：「参照整合性」449 ページ。

正規化

データベース構造を改善することです。リレーショナル・データベース理論に基づく規則に従って、冗長性を排除したり、編成を改良します。

生成されたジョイン条件

キーワード KEY または NATURAL に基づくジョイン結果に対する制限です。ナチュラル・ジョインでは、生成されたジョイン条件は、2 つのテーブルの共通のカラム名に基づきます。キー・ジョインでは、生成されたジョインの条件は 2 つのテーブル間の外部キー関係に基づきます。

参照：「ジョイン」438 ページ、「ジョイン条件」438 ページ。

接続 ID

クライアント・アプリケーションとデータベース間の特定の接続に付けられるユニークな識別番号です。現在の接続 ID を確認するには、次の SQL 文を使用します。

```
SELECT connection_property( 'Number' )
```

接続プロファイル

ユーザ名、パスワード、サーバ名などの、データベースに接続するために必要なパラメータのセットです。便宜的に保管され使用されます。

相関名

クエリの FROM 句内で使用されるテーブルやビューの名前です。テーブルやビューの元の名前か、FROM 句で定義したエイリアスのいずれかになります。

抽出

SQL Remote レプリケーションでは、統合データベースから適切な構造とデータをアンロードする動作を指します。この情報は、リモート・データベースを初期化するときに使用されます。

Mobile Link 同期では、リファレンス・データベースから適切な構造とデータをアンロードする動作を指します。

参照：「レプリケーション」446 ページ。

通信ストリーム

Mobile Link では、Mobile Link クライアントと Mobile Link 同期サーバ間での通信にネットワーク・プロトコルが使用されます。

統合データベース

データベースをレプリケートするときに、データのマスタ・コピーを格納するデータベースです。統合データベースにはすべてのデータが保管されます。これに対して、リモート・データベースには、通常、データのサブセットだけが保管されます。競合や不一致が発生した場合、すべてのデータのプライマリ・コピーは統合データベースにあるとみなされます。

Mobile Link では、統合データベースとして Oracle、IBM の DB2、Microsoft SQL Server、Adaptive Server Anywhere、または Adaptive Server Enterprise を使用できます。

参照：「レプリケーション」446 ページ。

統合ログイン

オペレーティング・システムへのログイン、ネットワークへのログイン、データベースへの接続に、同一のユーザ ID とパスワードを使用するログイン機能の 1 つです。

同期

Mobile Link テクノロジを使用してデータベース間でデータをレプリケートする処理です。

SQL Remote では、同期はデータの初期セットを使ってリモート・データベースを初期化する処理を表すために特に使用されます。

参照：「Mobile Link」433 ページ、「SQL Remote」434 ページ。

同時性 (同時実行性)

互いに独立し、場合によっては競合する可能性のある 2 つ以上の処理を同時に実行することです。Adaptive Server Anywhere では、自動的にロックを使用して各トランザクションを独立させ、同時に稼働するそれぞれのアプリケーションが一貫したデータのセットを参照できるようにします。

参照 :「トランザクション」440 ページ、「ロック」447 ページ、「独立性レベル」452 ページ。

独立性レベル

あるトランザクションの動作が、同時に処理されている別のトランザクションの動作からどの程度参照できるかを示します。独立性レベルには 0 から 3 までの 4 つのレベルがあります。最も高い独立性レベルには 3 が設定されます。デフォルトでは、レベルは 0 に設定されています。

内部ジョイン

2 つのテーブルがジョイン条件を満たす場合だけ、結果セットにローが表示されるジョインです。内部ジョインがデフォルトです。

参照 :「ジョイン」438 ページ、「外部ジョイン」448 ページ。

文レベルのトリガ

トリガ付きの文の処理が完了した後に実行されるトリガです。

参照 :「トリガ」441 ページ、「ロー・レベルのトリガ」446 ページ。

索引

記号

.NET 同期論理
サポートされる UNIX プラットフォーム 140
サポートされる Windows プラットフォーム 135
-? サーバ・オプション
WindowsCE でサポートされていない 116
@data オプション
WindowsCE でサポートされていない 116
1 対 1 の関係
定義 343
1 対多の関係
定義 343
508 条
準拠 5

A

ActiveSync
ASA に必要なバージョン 47
サポートされるバージョン 147
Adaptive Server Anywhere
SQLAnywhere Studio のインストール 46
SQLAnywhere Studio の使用 45
Ultra Light との比較 21, 22
WindowsCE でサポートされていない機能 112
WindowsCE のデータベースの設定 66
アプリケーション 17
概要 12
システムの稼働条件 15
使用目的 13
設計目的 21
説明 21
特長 14
内部 180
プログラミング・インタフェース 174
用語定義 431
Adaptive Server Anywhere と Ultra Light の概要
概要 11
ADO.NET Sample
使用 58
ADO
開発ツール 179
データ制御 424
ADOCE Sample
使用 61
ALTER DATABASE 文
WindowsCE の制限 121
ALTER WRITEFILE 文
WindowsCE でさぽーとされていない 119
ALTER 文
オートコミット 240
API
Adaptive Server Anywhere 174
ADO 179
Embedded SQL 177
JDBC 178
ODBC 176
OLE DB 179
Open Client 177
ARMV4T
プラットフォームのサポート 149
ARM V4T モード
Windows CE 149
ARM チップ
Windows CE 149
ARM プロセッサ
サポート 149
ASA
用語定義 431
asademo.db ファイル

説明　252
ASA Server Example
使用　55
AUTO_COMMIT オプション
Interactive SQL での変更のグループ分け　240

B

BACKUP 文
WindowsCE の制限　121
BETWEEN 条件
WHERE 句　203
BLOB
説明　340

C

-ca オプション
WindowsCE でサポートされていない　116
CE
サポートされるバージョン　149
サポートされるプロセッサ　149
Certicom
暗号化ソフトウェアの注文　5
FIPS
暗号化ソフトウェアの注文　5
-ch オプション
WindowsCE でサポートされていない　116
Command Sequence 通信プロトコル
図　174
説明　175
COMMENT 文
オートコミット　240
COMMIT 文
説明　241
トランザクション　240
COUNT 関数
グループ分けされたデータに対する適用　231
CREATE COMPRESSED DATABASE 文
WindowsCE でサポートされていない　119
CREATE DATABASE 文
WindowsCE の制限　121
creating databases for WindowsCE のデータベースの作成　73
CREATE EVENT 文
WindowsCE の制限　121
CREATE EXTERNLOGIN 文
WindowsCE でサポートされていない　119
CREATE FUNCTION 文
WindowsCE の制限　121
CREATE SERVER 文
WindowsCE でサポートされていない　119
CREATE TABLE 文
WindowsCE の制限　121
CREATE WRITEFILE 文
WindowsCE でさぽーとされていない　119
-cs オプション
WindowsCE でサポートされていない　116
current publisher
設定　372
-cw オプション
WindowsCE でサポートされていない　116

D

DBA 権限
用語定義　431
dbeng9
制限　16
データベース・サーバの起動　255
dbinit ユーティリティ
WindowsCE のデータベースの作成　72
初期化ユーティリティ
WindowsCE のデータベースの作成　72
dbisql ユーティリティ
説明　271
DBMS
用語定義　439
dbremote
サポートされる UNIX プラットフォーム　141
サポートされる Windows プラットフォーム　136

dbsrv9
WindowsCE 78
dbxtract
Windows CE でサポートされていない 115
DB 領域
用語定義 431
DDL
オートコミット 240
用語定義 440
DELETE 文
エラー 245
説明 239
例 246
DML
用語定義 440
DROP DATABASE 文
WindowsCE でサポートされていない 119
DROP SERVER 文
WindowsCE でサポートされていない 119
DROP 文
オートコミット 240
-d オプション
WindowsCE でサポートされていない 116

E

Embedded SQL
開発ツール 177
用語定義 431
ESQL Sample
使用 62

F

FILE
用語定義 431
FILE メッセージ・タイプ
用語定義 431
FTP メッセージ・タイプ
SQL Remote がサポートする UNIX プラットフォーム 141
SQL Remote がサポートする Windows プラットフォーム 136

G

-gb オプション
WindowsCE でサポートされていない 116
-ge オプション
WindowsCE でサポートされていない 116
go
文デリミタ 290
GRANT PUBLISH 文
概要 372
GRANT REMOTE 文
概要 372
grant オプション
用語定義 431
GROUP BY 句
エラー 232
集合関数 231
-gss オプション
WindowsCE でサポートされていない 116
-gx オプション
WindowsCE でサポートされていない 116

H

HAVING 句
GROUP BY 句 233
WHERE 句 234

I

iAnywhere JDBC ドライバ
WindowsCE でサポートされていない 112
InfoMaker
説明 409
チュートリアル 409
用語定義 432
INPUT 文

Interactive SQL での新しいローの挿入　286
INSERT 文
概要　237
例　244
INSTALL JAVA 文
WindowsCE でサポートされていない　119
Interactive SQL
[SQL 文] ウィンドウ枠　188
[SQL 文] ウィンドウ枠で選択したテキストのみ実行　291
[SQL 文] ウィンドウ枠のすべてのテキストの実行　277
WindowsCE 上のデータベースの管理　95
InteractiveSQL
WindowsCE のデータベースの作成　73
Interactive SQL
オプション　280
概要　17, 272
カラム名の検索　291
キーボード・ショートカット　277
起動　274
クイック・スタート　273
クエリ・エディタの表示　277
計算カラムの更新　284
コマンドの概要　288
コマンドのキャンセル　288
コマンドの再呼び出し　293
コマンドの実行　188, 288
コマンドの取得　188
コマンドの中断　288
コマンドの停止　288
コマンドの保存　188
コマンドのロード　188
コマンドのロギング　296
[コマンド履歴] ダイアログ　293
終了の影響　240
設定　280
説明　271
ツールバーの説明　275
データの表示　281
テーブル値の編集　282, 284
テーブル名の検索　291
テーブル・リストの表示　206, 291
トランザクションによる変更のグループ分け　240
ファンクション・キー　277
複数のウィンドウを開く　276
複数の文の結合　290
プロシージャ名の検索　291
プロシージャ・リストの表示　291
メイン・ウィンドウの説明　274
用語定義　432
レポートされるエラー　289
ローのコピー　287
ローの削除　286
ローの挿入　285
Interactive SQL コマンドのキャンセル　288
Interactive SQL での SQL 文の使用
説明　288
Interactive SQL の使用
概要　271
Interactive SQL を使用したデータの表示
説明　281
IN 条件
複合探索条件のショートカットとして使用　203
IP アドレス
CE デバイスでの特定　48

J

JAR ファイル
用語定義　432
Java
別のライセンス　5
Java クラス
用語定義　432
Java 同期論理
サポートされる UNIX プラットフォーム　140
サポートされる Windows プラットフォーム　135
jConnect
WindowsCE　67
WiondowsCE の制限　113

用語定義　432
JDBC
開発ツール　178
用語定義　432

L

LIKE 条件
概要　200
Linux
Adaptive Server Anywhere コンポーネント 139
Replication Agent がサポートするプラットフォーム　142
LTM
用語定義　434

M

Maintosh
サポートされるバージョン　149
Mac OS X
サポートされるバージョン　149
MAPI
用語定義　432
MAPI メッセージ・タイプ
SQL Remote がサポートする UNIX プラットフォーム　141
SQL Remote がサポートする Windows プラットフォーム　136
WindowsCE でサポートされていない　115
用語定義　432
Message Agent
サポートされる UNIX プラットフォーム　141
サポートされる Windows プラットフォーム 136
Microsoft VisualBasic クイック・スタート 423
MIPS チップ
Windows CE　149
Mobile Link
チュートリアル - 入門　347
用語定義　433
Mobile Link クライアント
用語定義　432
Mobile Link 同期
機能の比較　36
サポートされる UNIX プラットフォーム　140
サポートされる Windows プラットフォーム 135
スクリプト作成のチュートリアル　350
チュートリアル　347
別のアプローチ　25
Mobile Link 統合データベース
サポートされる RDBMS　144
Mobile Link の同期
サブスクリプション作成のチュートリアル 350
パブリケーション作成のチュートリアル　350
Mobile Link ユーザ
用語定義　433
[Mobile Link ユーザ作成] ウィザード
使用　357

N

NetWare
Adaptive Server Anywhere コンポーネント 132
用語定義　433
Novell NetWare
Adaptive Server Anywhere コンポーネント 132
NULL 値
カラムでの許可　237, 340
n 層コンピューティング
概要　171

O

ODBC
WiondowsCE の制限　113

開発ツール　176
説明　264
データ・ソース　264
データ・ソースの概要　263
用語定義　433
ODBC Sample
使用　64
ODBC アドミニストレータ
用語定義　433
ODBC データ・ソース
WindowsCE　52
用語定義　433
OLE DB
開発ツール　179
OLE DB と ADO のプログラミング・インタフェース　423
OLE DB ドライバ
サポートされるプロセッサ　149
ON 句
概要　209
Open Client
WindowsCE でさぽーとされていない　112
開発ツール　177
ORDER BY 句
パフォーマンス改善のためのインデックス使用　195
例　193
ローを常に同じ順序で表示するために必要　195

P

Palm Computing Platform
Ultra Light 開発環境　134
PDF
オンライン・マニュアルのアクセス　xiv
Physical Architect
PowerDesigner 参照　389
Pocket PC
Windows CE　149
PowerDesigner
説明　389
チュートリアル　389
用語定義　433
PowerDynamo
用語定義　433
PowerJ
用語定義　433
PUBLISH パーミッション
Remote パーミッション　372

Q

-qi オプション
WindowsCE でサポートされていない　116

R

RDBMS
定義　158
用語定義　446
REMOTE パーミッション
用語定義　433
REMOVE JAVA 文
WindowsCE でサポートされていない　119
REORGANIZE TABLE 文
WindowsCE でサポートされていない　119
Replication Agent
サポートされるオペレーティング・システム　142
サポートされるプラットフォーム　137
Replication Server
特徴　41
用語定義　434
ROLLBACK 文
概要　239
説明　242
トランザクション　240

S

-sc オプション

WindowsCE でサポートされていない　116
SELECT 文
Interactive SQL　281
概要　187
サブクエリ　218
select リスト
カラム名　191
計算カラム　191
SMTP メッセージ・タイプ
SQL Remote がサポートする UNIX プラットフォーム　141
SQL Remote がサポートする Windows プラットフォーム　136
SOUNDEX 関数
説明　201
SPX
WindowsCE でサポートされていない　112
SQL
WindowsCE でサポートされていない文　119
クエリの開発　272
説明　184
用語定義　434
SQL Anywhere Studio
WindowsCE でサポートされていない機能　112
SQLAnywhere Studio
WindowsCE での使用　45
概要　4
コンポーネント　4
SQL Anywhere Studio
マニュアル　x
SQL Modeler
PowerDesigner 参照　389
SQL Remote
WindowsCE でサポートされていない機能　115
PUBLISH パーミッションの付与　372
REMOTE パーミッションの付与　372
サポートされる UNIX プラットフォーム　141
サポートされる Windows プラットフォーム　136
チュートリアル　367
統合データベースの設定　371
特徴　40
用語定義　434
リモート・データベースの設定 (チュートリアル)　376
SQL とデータベース・コンピューティング
説明　184
SQL 文
用語定義　434
WindowsCE でサポートされていない文　119
ssqueue
サポートされる UNIX プラットフォーム　141
サポートされる Windows プラットフォーム　136
ssremote
サポートされる UNIX プラットフォーム　141
サポートされる Windows プラットフォーム　136
START JAVA 文
WindowsCE でサポートされていない　119
STOP JAVA 文
WindowsCE でサポートされていない　119
Sybase Central
Mobile Link ユーザの作成　357
SQL Remote パブリケーションの作成　374
WindowsCE 上のデータベースの管理　87
WindowsCE でサポートされていないウィザード　123
WindowsCE 用データベースの作成　69
ウィザード　322
概要　17
起動　301
グループの作成　314
コード・エディタ　317
スクリプト・バージョンの作成　359
説明　299
チュートリアル　299
データの表示　309
データベースのバックアップ　322
データベースへの接続　301
同期サブスクリプションの作成　357
統合データベースへのテーブル・スクリプト

の追加　360
メイン・ウィンドウ　304
ユーザのグループへの追加　315
ユーザの作成　315
用語定義　434
SYS
用語定義　434
-s オプション
WindowsCE でサポートされていない　116

T

TDS 通信プロトコル
図　174
-tmf オプション
WindowsCE でサポートされていない　116
-tmt オプション
WindowsCE でサポートされていない　116

U

-ua オプション
WindowsCE でサポートされていない　116
-ud オプション
WindowsCE でサポートされていない　116
Ultra Light
Adaptive Server Anywhere との比較　21, 22
開発プラットフォーム　133
サポートされる Windows プラットフォーム　133
設計目的　22
説明　22
ターゲット・プラットフォーム　147
用語定義　435
UNIX
Adaptive Server Anywhere コンポーネント　139
Replication Agent がサポートするプラットフォーム　142
UPDATE 文
エラー　246
概要　238
例　246
-ut オプション
WindowsCE でサポートされていない　116
-u オプション
WindowsCE でサポートされていない　116

V

V4T モード
ARM プロセッサ　149
VIM メッセージ・タイプ
SQL Remote がサポートする UNIX プラットフォーム　141
SQL Remote がサポートする Windows プラットフォーム　136
WindowsCE でサポートされていない　115
Visual Basic
クイック・スタート　424

W

WHERE 句
BETWEEN 条件　203
HAVING 句　234
テーブル内のローの変更　238
パターン一致　200
日付の比較の概要　199
例　197
ローの削除　239
Windows
サポートされるオペレーティング・システム　15
用語定義　435
Windows 98/Me
Adaptive Server Anywhere コンポーネント　132
Replication Agent がサポートするプラットフォーム　137
Windows CE
Adaptive Server Anywhere コンポーネント

132
WindowsCE
ADO.NET Sample の使用 58
ADOCE Sample の使用 61
ASA Server Example の使用 55
ESQL Sample 62
InteractiveSQL からのデータベースの管理 95
jConnect 67
ODBC Sample 64
ODBC データ・ソース 52
Windows CE
OLE DB サポート 149
Replication Agent がサポートするプラットフォーム 137
WindowsCE
SQLAnywhere Studio のインストール 46
SQLAnywhere Studio の使用 45
SQLAnywhere Studio の設定 46
Sybase Central からのデータベースの管理 87
アーカイブ・バックアップ 104
暗号化 67
イメージ・バックアップ 105
管理ユーティリティの使用 87
検証 103
コンピュータからの接続 47
サーバの起動 81
サーバの停止 85
サポートされていない Adaptive Server Anywhere の機能 112
サポートされていない SQLAnywhere Studio の機能 112
サポートされていない SQL Remote の機能 115
サポートされていない SQL 文 119
サポートされていない Sybase Central のウィザード 123
サポートされていない管理ツール 116
サポートされていないデータベース・サーバ・オプション 116
Windows CE
サポートされるバージョン 149
サポートされるプロセッサ 149
WindowsCE
サンプル・アプリケーション 55
サンプル・データベース 55
データベース・サーバ 78
複数のデータベースの開始 83
データベースの検証 103
データベースの作成 69
データベースの消去 76
データベースの設定 66
データベースのプロパティ 66
テーブルの検証 103
デバイスへのデータベースのコピー 75
トランザクション・ログ 66
トランザクション・ログの検証 103
バックアップとデータ・リカバリ 102
Windows CE
用語定義 435
WindowsCE
ライブ・バックアップ 104
リカバリ 106
WindowsCE でサポートされていない
-? サーバ・オプション 116
@data オプション 116
ALTER WRITEFILE 文 119
-ca オプション 116
-ch オプション 116
CREATE COMPRESSED DATABASE 文 119
CREATE EXTERNLOGIN 文 119
CREATE SERVER 文 119
CREATE WRITEFILE 文 119
-cs オプション 116
-cw オプション 116
dbxtract 115
DROP DATABASE 文 119
DROP SERVER 文 119
-d オプション 116
-gb オプション 116
-ge オプション 116
-gss オプション 116
-gx オプション 116
iAnywhere JDBC ドライバ 112
INSTALL JAVA 文 119

MAPI メッセージ・タイプ　115
Open Client　112
-qi オプション　116
REMOVE JAVA 文　119
REORGANIZE TABLE 文　119
-sc オプション　116
SPX　112
START JAVA 文　119
STOP JAVA 文　119
-s オプション　116
-tmf オプション　116
-tmt オプション　116
-ua オプション　116
-ud オプション　116
-ut オプション　116
-u オプション　116
VIM メッセージ・タイプ　115
-y オプション　116
外部ストアド・プロシージャ　112
[サービス作成] ウィザード　123
抽出ユーティリティ　115
[データベース圧縮] ウィザード　123
[データベース・アップグレード] ウィザード　123
[データベース・アンロード] ウィザード　123
[データベース作成] ウィザード　123
[データベース消去] ウィザード　123
[データベース展開] ウィザード　123
[データベース・バックアップ] ウィザード　123
動的キャッシュ・サイズ　112
パーソナル・サーバ　112
リモート・データ・アクセス　112
[ログ・ファイル設定の変更] ウィザード　123
[ログ・ファイル変換] ウィザード　123
WindowsCE デバイスへの接続
説明　47
WindowsCE の制限
ALTER DATABASE 文　121
BACKUP 文　121
CREATE DATABASE 文　121
CREATE FUNCTION 文　121
CREATE TABLE 文　121
jConnect　113
WindowsCE の制限
ODBC　113
WindowsCE のデータベースの作成
CREATE DATABASE 文　73
dbinit ユーティリティ　72
InteractiveSQL　73
WindowsCE のデータベースへの接続
説明　47
WindowsCE の制限
CREATE EVENT 文　121
WindowsCE 用データベースの作成
Sybase Central　69
Windows NT/2000/XP
Adaptive Server Anywhere コンポーネント　132
Replication Agent がサポートするプラットフォーム　137

X

x86 チップ
Windows CE　149
XScale プロセッサ
サポート　149

Y

-y オプション
WindowsCE でサポートされていない　116

あ

アーカイブ・バックアップ
WindowsCE　104
アーティクル
用語定義　435
アイコン

マニュアルで使用　xix
アクセシビリティ
アクセシビリティ有効化　5
値
Interactive SQL での編集　284
圧縮データベース・ファイル
用語定義　448
アップロード
用語定義　435
アトミック・トランザクション
用語定義　435
アプリケーションとデータベースの接続
説明　261
アルファベット順
ORDER BY 句　193
暗号化
WindowsCE　67
暗号化ソフトウェアの注文　5
アンロード
用語定義　435

い

一意性制約
用語定義　448
イメージ・バックアップ
WindowsCE　105
インクリメンタル・バックアップ
用語定義　435
印刷
InfoMaker レポート　421
インストール
SQLAnywhere Studio　7
WindowsCE への SQLAnywhere Studio のインストール　46
インタフェース
データベース・サーバ　256
インデックス
概要　195
用語定義　435
インデックス・コンサルタント
Interactive SQL　280

う

ウィザード
[Mobile Link ユーザ作成]　357
Sybase Central　322
unsupported Sybase Central wizards on WindowsCE でサポートされていない Sybase Central のウィザード　123
[外部キー作成]　345
[グループ作成]　314
[データベース作成]　330
[データベース・バックアップ]　322
[同期テーブル・スクリプト追加]　360
[バージョン追加]　359
[パブリケーション作成]　357, 374
[ユーザ作成]　315

え

エイリアス
カラム　191
エラー
Interactive SQL　289

お

オートコミット
ALTER 文　240
COMMENT 文　240
DROP 文　240
データ定義文　240
大文字と小文字の区別
SQL　188
テーブル名　188
オブジェクト・ツリー
用語定義　436
[オプション] ダイアログ
Interactive SQL　280
オペレーティング・システム
Mobile Link がサポートする UNIX プラットフォーム　140
Mobile Link がサポートする Windows プラット

フォーム 135
Replication Agent 137, 142
Ultra Light 147
Ultra Light がサポートする Windows プラットフォーム 133
UNIX 上の Adaptive Server Anywhere 139
UNIX 上の Embedded SQL 139
UNIX 上の Interactive SQL 139
UNIX 上の SQL Remote 141
UNIX 上の Sybase Central 139
UNIX 上の管理ユーティリティ 139
UNIX 上のリモート・データ・アクセス 139
Windows 上の Adaptive Server Anywhere 132
Windows 上の Embedded SQL 132
Windows 上の Interactive SQL 132
Windows 上の Java 132, 139
Windows 上の SQL Remote 136
Windows 上の Sybase Central 132
Windows 上の管理ユーティリティ 132
Windows 上のリモート・データ・アクセス 132
サポート 130
サポートされるバージョン 149
サポート対象 15
オンライン・ヘルプ
使用 xiv
オンライン・マニュアル (PDF)
使用 xiv

か

カーソル
用語定義 436
カーソル位置
用語定義 436
カーソル結果セット
用語定義 436
カーディナリティ
関係 343
開始
トランザクション 240
階層データベースの構成
説明 29
概念データベース・モデル
定義 329
開発
SQL 文 272
開発プラットフォーム
Ultra Light 133
外部キー
説明 161
挿入 244
定義 160
用語定義 448
[外部キー作成] ウィザード
使用 345
外部キー制約
用語定義 448
外部参照
定義 224
外部ジョイン
概要 216
用語定義 448
外部ストアド・プロシージャ
WindowsCE でサポートされていない 112
外部テーブル
用語定義 448
外部ログイン
用語定義 449
可用性
Adaptive Server Anywhere コンポーネント 17
カラム
Interactive SQL 内の検索 291
NULL の許可 340
エイリアス 191
計算 191
順番 190
説明 158
テーブルからの選択 190
プライマリ・キー 311
関係
1 対 1 343
1 対多 343

エンティティ　159
カーディナリティ　343
説明　343
多対多　344
関数
SOUNDEX 関数　201
完全リリース
説明　131
管理ツール
WindowsCE でサポートされていない機能　116
管理ユーティリティ
WindowsCE での使用　87

き

キー
外部　160
説明　160
プライマリ　160
キー・ジョイン
概要　211
キーボード・ショートカット
Interactive SQL　277
規格
508 条　5
規則
表記　xviii
起動
InfoMaker　411
Interactive SQL　274
Sybase Central　301
強化
InfoMaker を使用したレポート　417
競合
用語定義　449
競合トリガ
用語定義　449
強力な暗号化
暗号化ソフトウェアの注文　5

く

クイック・スタート
Interactive SQL　273
Visual Basic アプリケーションの開発　424
クエリ
Interactive SQL　288
SELECT 文　187
定義　185
用語定義　436
クエリ・エディタ
Interactive SQL　280
Interactive SQL での表示　277
クエリの最適化
Adaptive Server Anywhere の特長　14
組
概要　159
組み込みデータベース
定義済み　167
要件　13
クライアント／サーバ
定義済み　170
用語定義　436
クリア
[SQL 文] ウィンドウ枠　288
グループ
Sybase Central を使用したユーザの追加　315
作成　314
追加　314
グループ化
InfoMaker レポート　417
[グループ作成] ウィザード
使用　314
グループ分けされたデータ　229
グローバル・テンポラリ・テーブル
用語定義　436

け

計算カラム
Interactive SQL での新しいローへの追加　285
Interactive SQL での更新　284

Interactive SQL での再計算　284
結果セット
Interactive SQL でのテーブル値の編集　282, 284
トラブルシューティング　195
複数回のクエリの実行　195
ローのコピー　287
ローの削除　286
ローの挿入　285
結合
Interactive SQL での複数の文　290
検査
データ整合性　244
検索
Interactive SQL 内カラム　291
Interactive SQL 内テーブル　291
Interactive SQL 内プロシージャ　291
検査制約
用語定義　449
検証
WindowsCE　103
用語定義　449

こ

更新
Interactive SQL での値　284
データ　235
コピー
Interactive SQL でのロー　287
データベース
WindowsCE デバイスへのコピー　75
コピー
WindowsCE デバイスへのデータベースのコピー　75
コマンド
Interactive SQL でのキャンセル　288
Interactive SQL での再呼び出し　293
Interactive SQL での実行　188, 288
Interactive SQL での取得　188
Interactive SQL での中断　288
Interactive SQL での停止　288
Interactive SQL での編集　293
Interactive SQL での保存　188
Interactive SQL でのロード　188
Interactive SQL でのロギング　296
コマンド・ファイル
[SQL 文] ウィンドウ枠　290
概要　290
構築　290
用語定義　437
コマンド・ライン・ユーティリティ
概要　18
[コマンド履歴] ダイアログ
Interactive SQL でのコマンドの再呼び出し　293
Interactive SQL での使用　293
コンボイ
用語定義　437
コンポーネント
可用性　17

さ

サーバ・オプション
WindowsCE でサポートされていない　116
サーバ起動同期
用語定義　437
サーバ名
概要　257
サービス
用語定義　437
[サービス作成] ウィザード
WindowsCE でサポートされていない　123
再呼び出し
Interactive SQL のコマンド　293
削除
Interactive SQL でのローの使用　286
テーブル　313
テーブルのロー　286
作成
InfoMaker データベース・プロファイル　411

InfoMaker を使用したフォーム　409
InfoMaker を使用したレポートの作成　409, 413
SQLRemote 用のリモート・データベース　376
WindowsCE のデータベース　69
グループ　314
単純な ODBC データ・ソース　263
データベース　330
テーブル　311
ユーザ　315
サブクエリ
概要　218
ジョイン　224
相関サブクエリ　224
トラブルシューティング　222
比較　221
用語定義　437
サブクエリを使用したデータの選択
概要　217
サブスクリプション
用語定義　437
サポート
ニュースグループ　xxiii
サポートされていない機能
WindowsCE の制限　112
サポートされるプラットフォーム
Adaptive Server Anywhere　15
Adaptive Server Anywhere 版　131
Mobile Link がサポートする UNIX プラットフォーム　140
Mobile Link がサポートする Windows プラットフォーム　135
UNIX 上の SQL Anywhere Studio コンポーネント　139
Windows 上の SQL Anywhere Studio コンポーネント　132
サポートされるオペレーティング・システム　130
説明　129
参照整合性
用語定義　449
3 層コンピューティング
概要　171
サンプル・アプリケーション
ADO.NET Sample　58
ADOCE Sample　61
ASA Server Example　55
ESQL Sample　62
ODBC Sample　64
WindowsCE　55
サンプル・データベース
WindowsCE　55
起動　255
接続　274
説明　252

し

識別子
用語定義　449
システム・オブジェクト
用語定義　437
システム障害
トランザクション　242
システム・テーブル
定義　186
用語定義　437
システムの稼働条件
Adaptive Server Anywhere　15, 132
システム・ビュー
用語定義　438
実行
Interactive SQL でのコマンド　288
Interactive SQL のコマンド　288
複数回のクエリ　195
射影
定義　185
集合関数
概要　229
グループ分けされたデータに対する適用　231
集合データの選択
概要　227
述部

概要　202
用語定義　449
取得
Interactive SQL のコマンド　293
ジョイン
概要　206
サブクエリ　224
用語定義　438
ジョイン条件
用語定義　438
ジョイン・タイプ
用語定義　438
消去
WindowsCE デバイスのデータベース　76
条件
GROUP BY 句　233
検索　197, 203
パターン一致　200
照合
用語定義　450

す

スキーマ
定義　330
データベースの設計　390
表示　304
用語定義　438
スクリプト
用語定義　438
スタート・メニュー
Adaptive Server Anywhere　8
スタンドアロン・アプリケーション
定義済み　167
ストアド・プロシージャ
パーミッションの設定　318
表示　317
用語定義　438

せ

正規化
用語定義　450
制限
定義　185
整合性
検査　244
用語定義　450
生成されたジョイン条件
用語定義　450
制約
Sybase Central　304
用語定義　450
セキュリティ・オプション
購入　5
別のライセンス　5
設計
PowerDesigner の使用　389
データベース　327, 329
セッション・ベースの同期
用語定義　438
接続
Sybase Central からの接続　301
WindowsCE　47
WindowsCE のデータベースへの接続　47
概要　262
サンプル・データベース　274
接続 ID
用語定義　450
接続プロファイル
用語定義　450
設定
SQLRemote のリモート・データベース (チュートリアル)　376
WindowsCE のデータベース　66
統合データベース　371
選択
定義　185

そ

相関サブクエリ
定義 224
相関名
用語定義 451
挿入
Interactive SQL でのテーブルへのローの挿入 285
双方向レプリケーション
説明 31
ソート
InfoMaker レポート 417
クエリの結果 193
属性
テーブル 159

た

ターゲット・プラットフォーム
Ultra Light 147
ダイアログ・ボックス
[コマンド履歴] 293
ダウンロード
用語定義 438
多層コンピューティング
概要 171
探索条件
GROUP BY 句 233
概要 197
サブクエリ 218
ショートカット 203
パターン一致 200
日付の比較 199

ち

チェックポイント
用語定義 438
蓄積転送
SQL Remote レプリケーション 32
抽出
用語定義 451
抽出ユーティリティ
WindowsCE でサポートされていない 115
チュートリアル
InfoMaker 409
Mobile Link 同期の基本 347
PowerDesigner 389
SQL Remote 367
Sybase Central 299
Visual Basic アプリケーションの開発 424
サンプル・データベースへの接続 251
データベースの設計 329
直積
概要 208

つ

追加
Interactive SQL での新しいロー 285
ロー 237
通信ストリーム
用語定義 451
通信プロトコル
Adaptive Server Anywhere 175
ツールバー
Interactive SQL 275

て

停止
WindowsCE 上のサーバ 85
データベース・サーバ 259
データ型
用語定義 440
データ・ソース
WindowsCE 52
概要 263
データの要約 228
データベース
SQL 184

Sybase Central を使用した WindowsCE の管理 87
WindowsCE でのアーカイブ・バックアップ 104
WindowsCE でのイメージ・バックアップ 105
WindowsCE でのライブ・バックアップ 104
WindowsCE でのリカバリ 106
WindowsCE デバイスからの消去 76
WindowsCE の InteractiveSQL を使用した管理 95
WindowsCE の設定 66
オブジェクト 162
オブジェクト・プロパティ 306
管理 299
クエリ 185
クライアント・アプリケーション 164
言語インタフェース 164
コンポーネント 164
サーバ 164
作成 330
システム・テーブル 186
スキーマの表示 304
設計 327
設計概念 329
接続 274, 301
バックアップ 322
ファイル 181
用語定義 439
リバース・エンジニアリング 392
リレーショナル 158
[データベース圧縮] ウィザード
WindowsCE でサポートされていない 123
[データベース・アップグレード] ウィザード
WindowsCE でサポートされていない 123
[データベース・アンロード] ウィザード
WindowsCE でサポートされていない 123
データベース・エンジン
定義済み 167
データベース・オブジェクト
説明 162
用語定義 439
データベース管理者
用語定義 439
データベース・サーバ
WindowsCE 78
Windows CE 上で停止 85
WindowsCE でサポートされていないオプション 116
インタフェース 256
ウィンドウ 256
起動 255
実行 251, 261
接続先 251, 261
説明 252
停止 259
内部 180
パーソナルとネットワークの違い 16
用語定義 439
データベース・サイズ
マルチギガバイトのデータベース 13
[データベース作成] ウィザード
WindowsCE でサポートされていない 123
使用 330
[データベース消去] ウィザード
WindowsCE でサポートされていない 123
データベース所有者
用語定義 439
データベース・スキーマ
表示 304
データベース設計のモデル化
PowerDesigner の使用 389
データベース接続
用語定義 440
データベース・テーブルからのデータの選択
概要 183
[データベース展開] ウィザード
WindowsCE でサポートされていない 123
データベース内の Java
購入 5
別のライセンス 5
データベースの暗号化

WindowsCE　67
暗号化ソフトウェアの注文　5
データベースの管理
Sybase Central for WindowsCE からの管理　87
WindowsCE の InteractiveSQL からの管理　95
データベースの検証
WindowsCE　103
データベースの更新
概要　235
データベースのコピー
WindowsCE デバイスへのコピー　75
データベースの作成
WindowsCE　69
データベース
WindowsCE 用に作成　69
データベースの消去
WindowsCE デバイスからの消去　76
データベースの設計と構築
説明　327
データベースの設定
WindowsCE　66
データベースの定義
PowerDesigner　389
データベースのプロパティ
WindowsCE　66
[データベース・バックアップ] ウィザード
WindowsCE でサポートされていない　123
使用　322
データベース・ファイル
概要　181
用語定義　439
データベース・プロファイル
InfoMaker　411
データベース名
用語定義　440
データベース・ユーティリティ
ウィザード　322
データ・リカバリ
トランザクション　242
テーブル
Interactive SQL 内の検索　291
外部キー　161
カラムの追加　311
削除　313
作成　311
設計　334
説明　158
追加　311
特性　158
表示　304
プライマリ・キーの作成　311
編集　311
テーブルからのローの選択
説明　197
テーブル値
Interactive SQL での編集　282, 284
テーブルの検証
WindowsCE　103
[テーブル名のルックアップ] ダイアログ
テーブル・リストの表示　206
テクニカル・サポート
ニュースグループ　xxiii
デッドロック
用語定義　440
展開版
説明　131
展開リリース
説明　131
テンポラリ・テーブル
用語定義　440
テンポラリ・ファイル
概要　181

と

同期
Mobile Link 階層データベースの構成　29
Mobile Link 機能の比較　36
Mobile Link 入門チュートリアル　347
テクノロジの比較　36
別のアプローチ　25
用語定義　451
[同期テーブル・スクリプト追加] ウィザード

使用　360
統合データベース
サポートされる RDBMS　144
設定　371
用語定義　451
統合ログイン
用語定義　451
同時性 (同時実行性)
用語定義　452
動的キャッシュ・サイズ
WindowsCE でサポートされていない　112
特徴
Mobile Link 同期　39
Replication Server　41
SQL Remote　40
独立性レベル
用語定義　452
ドメイン
概要　406
トラブルシューティング
GROUP BY 句　232
結果セットが変わったように見える　195
サブクエリ　222
ナチュラル・ジョイン　214
トランザクション
開始　240
完了　240
データ・リカバリ　242
変更のグループ分け　240
用語定義　440
トランザクション処理
Adaptive Server Anywhere の特長　14
データ・リカバリ　242
トランザクション単位の整合性
用語定義　441
トランザクションによる変更のグループ分け
説明　240
トランザクションの完了
説明　240
トランザクションの終了
説明　240
トランザクション・ログ
WindowsCE　66
概要　181
用語定義　441
トランザクション・ログの検証
WindowsCE　103
トランザクション・ログ・ミラー
用語定義　441
トランスポート・レイヤ・セキュリティ
暗号化ソフトウェアの注文　5
注文　5
トリガ
表示　317
用語定義　441

な

内部
Adaptive Server Anywhere　180
データベース・サーバ　180
内部ジョイン
用語定義　452
ナチュラル・ジョイン
エラー　214
概要　214

に

ニュースグループ
テクニカル・サポート　xxiii
入力
Interactive SQL コマンド　288
Interactive SQL での複数の文　290

ね

ネットワーク サーバ
ソフトウェア要件　15
ネットワーク・サーバ
プラットフォーム・サポート　16

用語定義　441

は

[バージョン追加] ウィザード
使用　359
パーソナル・サーバ
WindowsCE でサポートされていない　112
制限　16
プラットフォーム・サポート　16
用語定義　441
ハードウェア要件
Adaptive Server Anywhere の特長　14
パーミッション
PUBLISH　371
Remote　371
設定　318
バイナリ・ラージ・オブジェクト
説明　340
パイプライン
InfoMaker を使用して作成　409
パススルー・モード
用語定義　441
パスワード
新しいデータベースへの接続　332
使用　274
パターン一致
概要　200
バックアップ
Sybase Central　322
WindowsCE での実行　87
WindowsCE のデータベース　102
概要　322
実行中のデータベース　322
バックアップとデータ・リカバリ
WindowsCE　102
WindowsCE での方法　87
パッケージ
用語定義　442
パフォーマンス
レプリケーションの利点　26
パフォーマンス統計値
用語定義　442
パフォーマンスのモニタ
Adaptive Server Anywhere の特長　14
パブリケーション
用語定義　442
[パブリケーション作成] ウィザード
Mobile Link パブリケーションの作成　357
SQL Remote パブリケーションの作成　374
パブリケーションの更新
用語定義　442
パブリッシャ
作成　372
用語定義　442
パラメータ
関数　229
ハンドヘルド PC
Windows CE　149

ひ

比較
概要　199
サブクエリの使用　221
説明　197
ビジネス・ルール　406
用語定義　442
日付
組み合わせ　202
探索条件　203
探索条件の概要　199
複合　202
ビットマップ
blob の格納　340
ビュー
表示　317
用語定義　443
表記
規則　xviii
表示
Sybase Central を使用したデータ　309

ふ

ファイル共有メッセージ・タイプ
SQL Remote がサポートする UNIX プラットフォーム 141
SQL Remote がサポートする Windows プラットフォーム 136
ファイルベースのダウンロード
用語定義 443
ファンクション・キー
Interactive SQL 277
フィードバック
提供 xxiii
マニュアル xxiii
フェールオーバ
用語定義 443
フォーム
InfoMaker を使用して作成 409
複合探索条件
使用 202
複数の Interactive SQL ウィンドウを開く
説明 276
複数のデータベース
単一サーバでの実行 172
複数のデータベースの開始
WindowsCE 83
複数のテーブルからのデータ選択
概要 205
物理データ・モデル
PowerDesigner 389
不等号
テスト 199
プライマリ・キー
作成 311
定義 160
用語定義 443
プライマリ・キー制約
用語定義 443
プライマリ・テーブル
用語定義 443
プラグイン・モジュール
用語定義 443
プラットフォーム
Mobile Link UNIX オペレーティング・システム 140
Mobile Link Windows オペレーティング・システム 135
サポートされるオペレーティング・システム 130
サポート対象 15
フル・バックアップ
用語定義 443
プレビュー
InfoMaker を使用したレポート 414
プロキシ・テーブル
概要 173
用語定義 444
プログラミング・インタフェース
Adaptive Server Anywhere 174
Adaptive Server Anywhere でのサポート対象 174
ADO 179
Embedded SQL 177
JDBC 178
ODBC 176
OLE DB 179
Open Client 177
プログラム・グループ
Adaptive Server Anywhere 8
プロシージャ
Interactive SQL 内の検索 291
プロシージャ・エディタ
Sybase Central 317
プロパティ
データベース・オブジェクト 306
文レベルのトリガ
用語定義 452

へ

ベース・テーブル 162
用語定義 444
別途ライセンスが入手可能なソフトウェア
Java とセキュリティ 5
ヘルプに関するヘルプ

情報のアクセス　xiv
編集
Interactive SQL でのテーブル値　282, 284

ほ

ホスト・プラットフォーム
Ultra Light がサポートする Windows プラットフォーム　133
保存
InfoMaker を使用したレポート　415
トランザクション結果　240

ま

マニュアル
SQL Anywhere Studio　x
マルチユーザ・データベース
定義済み　170

め

メタデータ
システム・テーブル　186
用語定義　444
メッセージ・システム
用語定義　444
メッセージ・タイプ
用語定義　444
メッセージベースのレプリケーション
説明　32

も

モバイル・コンピューティング
要件　13

ゆ

ユーザ
Sybase Central を使用してグループへ追加　315
管理　314
作成　315
追加　315
ユーザ ID
新しいデータベース　332
使用　274
説明　274
[ユーザ作成] ウィザード
使用　315
ユーザ定義データ型
用語定義　444
ユーティリティ
概要　18

よ

要件
Adaptive Server Anywhere　15

ら

ライセンス
Java　5
セキュリティ・オプション　5
ライト・ファイル
用語定義　445
ライブ・バックアップ
WindowsCE　104

り

リカバリ
Adaptive Server Anywhere の特長　14
WindowsCE　106
WindowsCE での方法　87
WindowsCE のデータベース　102

リダイレクタ
サポートされる Windows プラットフォーム 136
用語定義 445
リバース・エンジニアリング
データベース 392
リファレンス・データベース
用語定義 445
リモート DBA 権限
用語定義 445
リモート・データ・アクセス
WindowsCE でサポートされていない 112
概要 173
サポートされる UNIX プラットフォーム 139
サポートされる Windows プラットフォーム 132
リモート・データベース
SQLRemote 用の設定 (チュートリアル) 376
用語定義 445
リモート・ユーザ
用語定義 445
リレーショナル・データベース
概念 158
説明 159
用語 159
リレーショナル・データベース管理システム
定義 158

れ

レジストリ
WindowsCE で ASA を実行するための設定 50
レプリケーション
SQL Remote チュートリアル 367
階層データベースの構成 29
接続ベース 32
テクノロジの比較 36
別のアプローチ 25
メッセージベース 32
用語定義 446
レプリケーション・テクノロジ 36
特性 39
レプリケーション・テクノロジの課題 26
レプリケーション・テクノロジの選択 36
レプリケーションの頻度
用語定義 446
レプリケーション・メッセージ
用語定義 446
レポート
InfoMaker レポートの印刷 421
InfoMaker レポートのグループ化 417
InfoMaker レポートのソート 417
InfoMaker を使用した強化 417
InfoMaker を使用したプレビュー 414
InfoMaker を使用した保存 415
InfoMaker を使用して作成 409, 413
レポートとフォームの分配
InfoMaker を使用 409
連邦リハビリテーション法
508 条 5

ろ

ロー
Interactive SQL での値の編集 284
Interactive SQL でのコピー 287
Interactive SQL での挿入 285
Interactive SQL を使用した削除 286
Interactive SQL を使用した追加 285
説明 158
追加 237
テーブルからの選択 197
ローカル・テンポラリ・テーブル
用語定義 446
ロール
用語定義 447
ロールバック
トランザクション 240
ロールバック・ログ
用語定義 447
ロール名

用語定義　447
ロー・レベルのトリガ
用語定義　446
ローレベルのロック
Adaptive Server Anywhere の特長　14
ロギング
Interactive SQL 内のコマンド　296
ログ・ファイル
用語定義　447
[ログ・ファイル設定の変更] ウィザード
WindowsCE でサポートされていない　123
[ログ・ファイル変換] ウィザード
WindowsCE でサポートされていない　123
ロック
用語定義　447

わ

ワークグループ・コンピューティング
要件　13
ワーク・テーブル
用語定義　448
ワイルドカード
パターン一致　200

記号

用語解説
SQL Anywhere Studio 入門　431