



# SQL Anywhere® 11 紹介

2009 年 2 月

バージョン 11.0.1

## 版權と商標

Copyright © 2009 iAnywhere Solutions, Inc. Portions copyright © 2009 Sybase, Inc. All rights reserved.

iAnywhere との間に書面による合意がないかぎり、このマニュアルは現状のまま提供されるものであり、その使用または記載内容の誤りに対して一切の責任を負いません。

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。1) マニュアルの全部または一部にかかわらず、すべてのコピーにこの情報またはマニュアル内のその他の版權と商標の表示を含めること。2) マニュアルに変更を加えないこと。3) iAnywhere 以外の人間がマニュアルの著者または情報源であるかのように示す行為をしないこと。

iAnywhere®、Sybase®、および <http://www.sybase.com/detail?id=1011207> に記載されているマークは、Sybase, Inc. または子会社の商標です。® は米国での登録商標を示します。

このマニュアルに記載されているその他の会社名と製品名は各社の商標である場合があります。

---

## 目次

はじめに .....	v
SQL Anywhere のマニュアルについて .....	vi
<b>SQL Anywhere 11 の概要 .....</b>	<b>1</b>
SQL Anywhere の概要 .....	3
SQL Anywhere 11 の概要 .....	4
SQL Anywhere 11 のコンポーネント .....	6
SQL Anywhere 11 の特長 .....	14
サポートされるプラットフォーム .....	15
アクセシビリティ有効化オプション .....	16
データ管理テクノロジーの概要 .....	17
データベース・システムの各部分 .....	18
リレーショナル・データベースの概念 .....	20
SQL Anywhere の内部構造 .....	25
SQL Anywhere データベースと Ultra Light データベースとの比較 .....	29
データベースの使用例 .....	30
多層コンピューティングのアーキテクチャ .....	32
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 .....	33
ETL 機能 .....	34
プログラミング・インタフェース .....	35
データ交換テクノロジーの概要 .....	39
同期テクノロジーの比較 .....	40
モバイル・エンタープライズ・メッセージング : QAnywhere .....	49
<b>サンプル・データベース .....</b>	<b>53</b>
SQL Anywhere サンプル・データベース .....	55
サンプル・データベースについて .....	56
サンプル・データベースの再作成 .....	58
CustDB サンプル・データベース・アプリケーション .....	61
CustDB サンプル・データベースについて .....	62

<b>SQL Anywhere 11 のクイック・スタート .....</b>	<b>65</b>
SQL Anywhere 11 のクイック・スタート .....	67
概要 .....	68
SQL Anywhere サーバのクイック・スタート .....	70
その他のアプリケーション .....	72
 <b>用語解説 .....</b>	 <b>73</b>
用語解説 .....	75
 <b>索引 .....</b>	 <b>107</b>

---

# はじめに

## このマニュアルの内容

このマニュアルでは、データの管理および交換機能を提供する包括的なパッケージである SQL Anywhere 11 について説明します。SQL Anywhere を使用すると、サーバ環境、デスクトップ環境、モバイル環境、リモート・オフィス環境に適したデータベース・ベースのアプリケーションを迅速に開発できるようになります。

## 対象読者

このマニュアルは、SQL Anywhere 11 の主な機能の概要を説明することによって、アプリケーション開発者やデータベース管理者が SQL Anywhere 11 の適用範囲と機能を評価できるようにすることを目的としています。

## 始める前に

このマニュアルでは、リレーショナル・データベースについてある程度の知識を持っていることを前提としています。

## SQL Anywhere のマニュアルについて

SQL Anywhere の完全なマニュアルは 4 つの形式で提供されており、いずれも同じ情報が含まれています。

- **HTML ヘルプ** オンライン・ヘルプには、SQL Anywhere 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。

Microsoft Windows オペレーティング・システムを使用している場合は、オンライン・ヘルプは HTML ヘルプ (CHM) 形式で提供されます。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 11] - [マニュアル] - [オンライン・マニュアル] を選択します。

管理ツールのヘルプ機能でも、同じオンライン・マニュアルが使用されます。

- **Eclipse** UNIX プラットフォームでは、完全なオンライン・ヘルプは Eclipse 形式で提供されます。マニュアルにアクセスするには、SQL Anywhere 11 インストール環境の *bin32* または *bin64* ディレクトリから *sadoc* を実行します。
- **DocCommentXchange** DocCommentXchange は、SQL Anywhere マニュアルにアクセスし、マニュアルについて議論するためのコミュニティです。

DocCommentXchange は次の目的に使用できます (現在のところ、日本語はサポートされておられません)。

- マニュアルを表示する
- マニュアルの項目について明確化するために、ユーザによって追加された内容を確認する
- すべてのユーザのために、今後のリリースでマニュアルを改善するための提案や修正を行う

<http://dcx.sybase.com> を参照してください。

- **PDF** SQL Anywhere の完全なマニュアル・セットは、Portable Document Format (PDF) 形式のファイルとして提供されます。内容を表示するには、PDF リードが必要で、Adobe Reader をダウンロードするには、<http://get.adobe.com/reader/> にアクセスしてください。

Microsoft Windows オペレーティング・システムで PDF 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 11] - [マニュアル] - [オンライン・マニュアル - PDF] を選択します。

UNIX オペレーティング・システムで PDF マニュアルにアクセスするには、Web ブラウザを使用して *install-dir/documentation/ja/pdf/index.html* を開きます。

## マニュアル・セットに含まれる各マニュアルについて

SQL Anywhere のマニュアルは次の構成になっています。

- **『SQL Anywhere 11 - 紹介』** このマニュアルでは、データの管理および交換機能を提供する包括的なパッケージである SQL Anywhere 11 について説明します。SQL Anywhere を使用す

ると、サーバ環境、デスクトップ環境、モバイル環境、リモート・オフィス環境に適したデータベース・ベースのアプリケーションを迅速に開発できるようになります。

- **『SQL Anywhere 11 - 変更点とアップグレード』** このマニュアルでは、SQL Anywhere 11 とそれ以前のバージョンに含まれる新機能について説明します。
- **『SQL Anywhere サーバ - データベース管理』** このマニュアルでは、SQL Anywhere データベースを実行、管理、構成する方法について説明します。データベース接続、データベース・サーバ、データベース・ファイル、バックアップ・プロシージャ、セキュリティ、高可用性、Replication Server を使用したレプリケーション、管理ユーティリティとオプションについて説明します。
- **『SQL Anywhere サーバ - プログラミング』** このマニュアルでは、C、C++、Java、PHP、Perl、Python、および Visual Basic や Visual C# などの .NET 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。ADO.NET や ODBC などのさまざまなプログラミング・インタフェースについても説明します。
- **『SQL Anywhere サーバ - SQL リファレンス』** このマニュアルでは、システム・プロシージャとカタログ (システム・テーブルとビュー) に関する情報について説明します。また、SQL Anywhere での SQL 言語の実装 (探索条件、構文、データ型、関数) についても説明します。
- **『SQL Anywhere サーバ - SQL の使用法』** 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。
- **『Mobile Link - クイック・スタート』** このマニュアルでは、セッションベースのリレーショナル・データベース同期システムである Mobile Link について説明します。Mobile Link テクノロジーは、双方向レプリケーションを可能にし、モバイル・コンピューティング環境に非常に適しています。
- **『Mobile Link - クライアント管理』** このマニュアルでは、Mobile Link クライアントを設定、構成、同期する方法について説明します。Mobile Link クライアントには、SQL Anywhere または Ultra Light のいずれかのデータベースを使用できます。また、dbmlsync API についても説明します。dbmlsync API を使用すると、同期を C++ または .NET のクライアント・アプリケーションにシームレスに統合できます。
- **『Mobile Link - サーバ管理』** このマニュアルでは、Mobile Link アプリケーションを設定して管理する方法について説明します。
- **『Mobile Link - サーバ起動同期』** このマニュアルでは、Mobile Link サーバ起動同期について説明します。この機能により、Mobile Link サーバは同期を開始したり、リモート・デバイス上でアクションを実行することができます。
- **『QAnywhere』** このマニュアルでは、モバイル・クライアント、ワイヤレス・クライアント、デスクトップ・クライアント、およびラップトップ・クライアント用のメッセージング・プラットフォームである、QAnywhere について説明します。
- **『SQL Remote』** 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・システムについて説明します。このシステムによって、SQL Anywhere の統合データベースと複数の SQL Anywhere リモート・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。

- 『Ultra Light - データベース管理とリファレンス』 このマニュアルでは、小型デバイス用 Ultra Light データベース・システムの概要を説明します。
- 『Ultra Light - C/C++ プログラミング』 このマニュアルでは、Ultra Light C および Ultra Light C++ のプログラミング・インタフェースについて説明します。Ultra Light を使用すると、ハンドヘルド・デバイス、モバイル・デバイス、埋め込みデバイスのデータベース・アプリケーションを開発し、これらのデバイスに配備できます。
- 『Ultra Light - M-Business Anywhere プログラミング』 このマニュアルは、Ultra Light for M-Business Anywhere について説明します。Ultra Light for M-Business Anywhere を使用すると、Palm OS、Windows Mobile、または Windows を搭載しているハンドヘルド・デバイス、モバイル・デバイス、または埋め込みデバイスの Web ベースのデータベース・アプリケーションを開発し、これらのデバイスに配備できます。
- 『Ultra Light - .NET プログラミング』 このマニュアルでは、Ultra Light.NET について説明します。Ultra Light.NET を使用すると、PC、ハンドヘルド・デバイス、モバイル・デバイス、または埋め込みデバイスのデータベース・アプリケーションを開発し、これらのデバイスに配備できます。
- 『Ultra Light J』 このマニュアルでは、Ultra Light J について説明します。Ultra Light J を使用すると、Java をサポートしている環境用のデータベース・アプリケーションを開発し、配備することができます。Ultra Light J は、BlackBerry スマートフォンと Java SE 環境をサポートしており、iAnywhere Ultra Light データベース製品がベースになっています。
- 『エラー・メッセージ』 このマニュアルでは、SQL Anywhere エラー・メッセージの完全なリストを示し、その診断情報を説明します。

## 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表記規則について説明します。

### オペレーティング・システム

SQL Anywhere はさまざまなプラットフォームで稼働します。ほとんどの場合、すべてのプラットフォームで同じように動作しますが、いくつかの相違点や制限事項があります。このような相違点や制限事項は、一般に、基盤となっているオペレーティング・システム (Windows、UNIX など) に由来しており、使用しているプラットフォームの種類 (AIX、Windows Mobile など) またはバージョンに依存していることはほとんどありません。

オペレーティング・システムへの言及を簡素化するために、このマニュアルではサポートされているオペレーティング・システムを次のようにグループ分けして表記します。

- **Windows** Microsoft Windows ファミリを指しています。これには、主にサーバ、デスクトップ・コンピュータ、ラップトップ・コンピュータで使用される Windows Vista や Windows XP、およびモバイル・デバイスで使用される Windows Mobile が含まれます。  
特に記述がないかぎり、マニュアル中に Windows という記述がある場合は、Windows Mobile を含むすべての Windows ベース・プラットフォームを指しています。



- **UNIX** 特に記述がないかぎり、マニュアル中に UNIX という記述がある場合は、Linux および Mac OS X を含むすべての UNIX ベース・プラットフォームを指しています。

## ディレクトリとファイル名

ほとんどの場合、ディレクトリ名およびファイル名の参照形式はサポートされているすべてのプラットフォームで似通っており、それぞれの違いはごくわずかです。このような場合は、Windows の表記規則が使用されています。詳細がより複雑な場合は、マニュアルにすべての関連形式が記載されています。

ディレクトリ名とファイル名の表記を簡素化するために使用されている表記規則は次のとおりです。

- **大文字と小文字のディレクトリ名** Windows と UNIX では、ディレクトリ名およびファイル名には大文字と小文字が含まれている場合があります。ディレクトリやファイルが作成されると、ファイル・システムでは大文字と小文字の区別が維持されます。

Windows では、ディレクトリおよびファイルを参照するとき、大文字と小文字は**区別されません**。大文字と小文字を混ぜたディレクトリ名およびファイル名は一般的に使用されますが、参照するときはすべて小文字を使用するのが通常です。SQL Anywhere では、*Bin32* や *Documentation* などのディレクトリがインストールされます。

UNIX では、ディレクトリおよびファイルを参照するとき、大文字と小文字は**区別されます**。大文字と小文字を混ぜたディレクトリ名およびファイル名は一般的に使用されません。ほとんどの場合は、すべて小文字の名前が使用されます。SQL Anywhere では、*bin32* や *documentation* などのディレクトリがインストールされます。

このマニュアルでは、ディレクトリ名に Windows の形式を使用しています。ほとんどの場合、大文字と小文字が混ざったディレクトリ名をすべて小文字に変換すると、対応する UNIX 用のディレクトリ名になります。

- **各ディレクトリおよびファイル名を区切るスラッシュ** マニュアルでは、ディレクトリの区切り文字に円記号を使用しています。たとえば、PDF 形式のマニュアルは *install-dir¥Documentation¥ja¥pdf* にあります。これは Windows の形式です。

UNIX では、円記号をスラッシュに置き換えます。PDF マニュアルは *install-dir/documentation/ja/pdf* にあります。

- **実行ファイル** マニュアルでは、実行ファイルの名前は、Windows の表記規則が使用され、*.exe* や *.bat* などの拡張子が付きます。UNIX では、実行ファイルの名前に拡張子は付きません。

たとえば、Windows でのネットワーク・データベース・サーバは *dbsrv11.exe* です。UNIX では *dbsrv11* です。

- **install-dir** インストール・プロセス中に、SQL Anywhere をインストールするロケーションを選択します。このロケーションを参照する環境変数 *SQLANY11* が作成されます。このマニュアルでは、そのロケーションを *install-dir* と表します。

たとえば、マニュアルではファイルを *install-dir¥readme.txt* のように参照します。これは、Windows では、*%SQLANY11%¥readme.txt* に対応します。UNIX では、*\$SQLANY11/readme.txt* または *\${SQLANY11}/readme.txt* に対応します。

*install-dir* のデフォルト・ロケーションの詳細については、「[SQLANY11 環境変数](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

- **samples-dir** インストール・プロセス中に、SQL Anywhere に含まれるサンプルをインストールするロケーションを選択します。このロケーションを参照する環境変数 SQLANYSAMP11 が作成されます。このマニュアルではそのロケーションを *samples-dir* と表します。

Windows エクスプローラ・ウィンドウで *samples-dir* を開くには、[スタート] - [プログラム] - [SQL Anywhere 11] - [サンプル・アプリケーションとプロジェクト] を選択します。

*samples-dir* のデフォルト・ロケーションの詳細については、「[SQLANYSAMP11 環境変数](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

## コマンド・プロンプトとコマンド・シェル構文

ほとんどのオペレーティング・システムには、コマンド・シェルまたはコマンド・プロンプトを使用してコマンドおよびパラメータを入力する方法が、1 つ以上あります。Windows のコマンド・プロンプトには、コマンド・プロンプト (DOS プロンプト) および 4NT があります。UNIX のコマンド・シェルには、Korn シェルおよび bash があります。各シェルには、単純コマンドからの拡張機能が含まれています。拡張機能は、特殊文字を指定することで起動されます。特殊文字および機能は、シェルによって異なります。これらの特殊文字を誤って使用すると、多くの場合、構文エラーや予期しない動作が発生します。

このマニュアルでは、一般的な形式のコマンド・ラインの例を示します。これらの例に、シェルにとって特別な意味を持つ文字が含まれている場合、その特定のシェル用にコマンドを変更することが必要な場合があります。このマニュアルではコマンドの変更について説明しませんが、通常、その文字を含むパラメータを引用符で囲むか、特殊文字の前にエスケープ文字を記述します。

次に、プラットフォームによって異なるコマンド・ライン構文の例を示します。

- **カッコと中カッコ** 一部のコマンド・ライン・オプションは、詳細な値を含むリストを指定できるパラメータを要求します。リストは通常、カッコまたは中カッコで囲まれています。このマニュアルでは、カッコを使用します。次に例を示します。

```
-x tcpip(host=127.0.0.1)
```

カッコによって構文エラーになる場合は、代わりに中カッコを使用します。

```
-x tcpip{host=127.0.0.1}
```

どちらの形式でも構文エラーになる場合は、シェルの要求に従ってパラメータ全体を引用符で囲む必要があります。

```
-x "tcpip(host=127.0.0.1)"
```

- **引用符** パラメータの値として引用符を指定する必要がある場合、その引用符はパラメータを囲むために使用される通常の引用符と競合する可能性があります。たとえば、値に二重引用符を含む暗号化キーを指定するには、キーを引用符で囲み、パラメータ内の引用符をエスケープします。

```
-ek "my ¥"secret¥" key"
```

多くのシェルでは、キーの値は my "secret" key のようになります。

- **環境変数** マニュアルでは、環境変数設定が引用されます。Windows のシェルでは、環境変数は構文 `%ENVVAR%` を使用して指定されます。UNIX のシェルでは、環境変数は構文 `$ENVVAR` または `${ENVVAR}` を使用して指定されます。

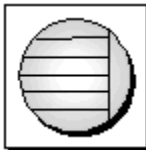
## グラフィック・アイコン

このマニュアルでは、次のアイコンを使用します。

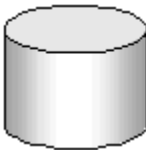
- クライアント・アプリケーション。



- SQL Anywhere などのデータベース・サーバ。



- データベース。ハイレベルの図では、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 サーバ、SQL Remote Message Agent などが挙げられます。



- プログラミング・インタフェース。

## ドキュメンテーション・チームへのお問い合わせ

このヘルプに関するご意見、ご提案、フィードバックをお寄せください。

SQL Anywhere ドキュメンテーション・チームへのご意見やご提案は、弊社までご連絡ください。頂戴したご意見はマニュアルの向上に役立たせていただきます。ぜひとも、ご意見をお寄せください。

### DocCommentXchange

DocCommentXchange を使用して、ヘルプ・トピックに関するご意見を直接お寄せいただくこともできます。DocCommentXchange (DCX) は、SQL Anywhere マニュアルにアクセスしたり、マニュアルについて議論するためのコミュニティです。DocCommentXchange は次の目的に使用できます (現在のところ、日本語はサポートされておりません)。

- マニュアルを表示する
- マニュアルの項目について明確化するために、ユーザによって追加された内容を確認する
- すべてのユーザのために、今後のリリースでマニュアルを改善するための提案や修正を行う

<http://dcx.sybase.com> を参照してください。

## 詳細情報の検索／テクニカル・サポートの依頼

詳しい情報やリソースについては、iAnywhere デベロッパー・コミュニティ (<http://www.iAnywhere.jp/developers/index.html>) を参照してください。

ご質問がある場合や支援が必要な場合は、次に示す Sybase iAnywhere ニュースグループのいずれかにメッセージをお寄せください。

ニュースグループにメッセージをお送りいただく際には、ご使用の SQL Anywhere バージョンのビルド番号を明記し、現在発生している問題について詳しくお知らせくださいますようお願いいたします。バージョンおよびビルド番号を調べるには、コマンド **dbeng11 -v** を実行します。

ニュースグループは、ニュース・サーバ [forums.sybase.com](http://forums.sybase.com) にあります。

以下のニュースグループがあります。

- [ianywhere.public.japanese.general](http://groups.google.com/group/sql-anywhere-web-development)

Web 開発に関する問題については、<http://groups.google.com/group/sql-anywhere-web-development> を参照してください。

**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也没有ありません。

iAnywhere のテクニカル・アドバイザー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。

---

# SQL Anywhere 11 の概要

この項では、SQL Anywhere 11 について説明します。また、SQL Anywhere 11 のデータ管理テクノロジーと交換テクノロジーについても説明します。

---

SQL Anywhere の概要 .....	3
データ管理テクノロジーの概要 .....	17
データ交換テクノロジーの概要 .....	39





---

# SQL Anywhere の概要

## 目次

SQL Anywhere 11 の概要 .....	4
SQL Anywhere 11 のコンポーネント .....	6
SQL Anywhere 11 の特長 .....	14
サポートされるプラットフォーム .....	15
アクセシビリティ有効化オプション .....	16

---

## SQL Anywhere 11 の概要

SQL Anywhere は、データの管理およびエンタープライズ・データの交換テクノロジーを提供する包括的なパッケージです。SQL Anywhere を使用すると、サーバ環境、デスクトップ環境、モバイル環境、リモート・オフィス環境に適したデータベース・ベースのアプリケーションを迅速に開発できるようになります。

次に、SQL Anywhere の特長を示します。

- **データ管理テクノロジー** SQL Anywhere は、独立系ソフトウェア・ベンダによりアプリケーションに導入された高性能なデータベース・サーバから、膨大な数のハンドヘルド・デバイスに導入されたモバイル・データベースに至るまで、多種多様なフロントエンド環境におけるオペレーションの課題に対応できるように設計された、エンタープライズ・クラスのデータベースを提供します。

[「リレーショナル・データベース・システム」 6 ページ](#)を参照してください。

- **データ交換テクノロジー** SQL Anywhere は、不安定な有線ネットワークやワイヤレス・ネットワークと、バックエンド・データベース、アプリケーション・サーバ、メッセージング・システムとの間で行われる複雑なデータの交換処理を行うデータ交換テクノロジーを提供します。また、SQL Anywhere のモバイル・メッセージングおよび同期テクノロジーにより、分散されたモバイル・コンピューティングでのセキュリティ保護されたメッセージ配信が保証されます。

[「データ交換テクノロジー」 6 ページ](#)を参照してください。

- **設計ツールと管理ツール** SQL Anywhere に用意されているツール・セットにより、データベース駆動型アプリケーションの設計および開発が向上し、データベースやデータ交換環境の管理が簡素化されています。

[「設計ツールと管理ツール」 8 ページ](#)を参照してください。

## フロントエンド環境での SQL Anywhere

SQL Anywhere テクノロジーは、10000 を超える企業でさまざまな方法で利用されています。SQL Anywhere の一般的な用途は次のとおりです。

- **クライアント／サーバ・アプリケーション** ユーザが 5 人、50 人、500 人、またはそれ以上であっても、SQL Anywhere はサーバ・アプリケーションの強力なデータベース・ソリューションとなって、購入後すぐに高性能を実現し、メンテナンスの必要性和コストを低く抑えることができます。

SQL Anywhere は、数百のアクティブ・ユーザ、数百ギガバイトのデータ、数百万のローをサポートするように簡単に拡張できます。また、使いやすさと管理機能によって、コストを低く抑えつつ、高いパフォーマンスを実現できます。

この配備モデルは、ユーザの大部分がネットワークに接続されている場合に最適です。

[「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」 31 ページ](#)を参照してください。

- **デスクトップ・アプリケーション** SQL Anywhere は、エンタープライズ・クラスの機能を提供しますが、エンタープライズ・データベースに特徴的な巨大になりすぎることはありません。信頼性とパフォーマンスが高く、メモリとシステム・リソースが効率的に使用されるので、デスクトップ・ユーザやラップトップ・ユーザはデータベースの存在を意識しなくて済みます。

企業は、既存のアプリケーションに SQL Anywhere を組み込んできました。これは、SQL Anywhere データベースが、広範に配備可能で最小限の管理ですむ環境で使用できるように設計されており、最小限のメモリとディスク領域しか必要としないためです。

「[デスクトップ・アプリケーションと組み込みデータベース](#)」 30 ページを参照してください。

- **リモート・オフィス・アプリケーション** SQL Anywhere のデータ交換アーキテクチャは、オフィス内で、または地理的に分散されているオフィスや従業員の間で、データを管理したり送信したりする際の課題に対応します。

業務を効率よく遂行するのに必要なデータをリモート・ワークに提供し、ビジネスの動向を理解するのに必要なクリティカルな情報を本社に提供するために、多くの企業が SQL Anywhere のデータベースとデータ交換テクノロジーを選択しています。

「[統合リモート・データベース](#)」 42 ページを参照してください。

- **モバイルおよびワイヤレス・アプリケーション** 業界をリードするモバイル・データベースとして認知されている SQL Anywhere を使用すると、モバイル・ワークは自分のデータとコーポレート・アプリケーションにアクセスできるようになります。接続やアプリケーションの種類にかかわらず、SQL Anywhere のデータ交換テクノロジーによって、モバイル・ワークは必要なときに必要な情報を得られるようになり、生産性を維持できます。モバイル・ワークはオフラインで情報にアクセスしてトランザクション・キューを追加していくことができるので、通信コストが削減され、アプリケーションとバッテリーのパフォーマンスが向上します。

企業は、ラップトップ、ハンドヘルド・デバイス、スマートフォンで実行されるデータおよびモバイル・アプリケーションの管理を、安心して SQL Anywhere に任せることができます。

次の項を参照してください。

- 「[Ultra Light の概要](#)」 『[Ultra Light データベース管理とリファレンス](#)』
- 「[SQL Anywhere for Windows Mobile](#)」 『[SQL Anywhere サーバ - データベース管理](#)』

## SQL Anywhere 11 のコンポーネント

次の各項では、SQL Anywhere に付属のリレーショナル・データベース・システム、データ交換テクノロジー、設計ツール、および管理ツールについて説明します。

### リレーショナル・データベース・システム

SQL Anywhere では、SQL Anywhere サーバと Ultra Light の 2 種類のデータベースを使用できます。

#### SQL Anywhere サーバ

SQL Anywhere サーバは、トランザクション処理のフル・サポート、参照整合性、マテリアライズド・ビュー、スナップショット・アイソレーション、データベースのミラーリングとサーバのクラスタリングによる高可用性、SQL および Java のストアド・プロシージャ、トリガ、ローレベルのロック、自動イベント・スケジュール、自動バックアップとリカバリなど、さまざまなエンタープライズ・クラスの機能を提供します。SQL Anywhere サーバは、数百のアクティブ・ユーザと数百ギガバイトのデータまで簡単に拡張できます。一方、占有容量は小さく、管理を自動化する機能が数多く用意されているので、サーバおよびデスクトップ・アプリケーションに埋め込んで客先やリモート・サイトに広く配備するデータベースとして最適です。

#### Ultra Light

小規模なデータ駆動型アプリケーションが必要な環境では、Ultra Light データベースが最適です。Ultra Light は、メモリやシステム要件を最小化してハンドヘルド・デバイスやその他のモバイル・デバイス (BlackBerry など) に配備できるように特に設計された、フル機能のリレーショナル・データベース管理システムです。Ultra Light は、完全なトランザクション処理、開発モデルの選択、他のデータベースとデータを交換するためにデータベースに直接組み込まれた同期クライアントを提供します。

「Ultra Light J」は、Java による Ultra Light のサブセットの実装で、BlackBerry スマートフォンと Java ME 用に設計されています。「[Ultra Light J 概要](#)」[『Ultra Light J』](#)を参照してください。

### データ交換テクノロジー

SQL Anywhere では、さまざまなオプションを用意して、既存のエンタープライズ・システムやモバイル・デバイスとのデータ交換を実現しています。たとえば、次のようなツールが用意されています。

- **Mobile Link - 同期** Mobile Link は、セッションベースの双方向同期を実現します。Mobile Link は、統合データベースと多数のリモート Ultra Light または SQL Anywhere データベースとの間のデータ交換、または中央の非リレーショナル・データ・ソースと多くの Ultra Light または SQL Anywhere データベースとの間のデータ交換に最適です。

Mobile Link の同期中に、リモート・データベースは Mobile Link サーバを使用した前回の同期後に行われた変更をアップロードします。Mobile Link サーバは、このデータを受信する

と、統合データベースを更新し、統合データベースの変更内容をリモート・データベースにダウンロードします。また、データベース間の接続が失われたときにデータベースのトランザクションの整合性を確保します。さらに、データの変更による競合を解決するためのメカニズムも提供します。

Mobile Link のファイル転送機能を使用すると、接続してデータを同期させるときに、同時にリモート・アプリケーションにファイルを転送できるので、新しいリモート・データベースを移植するときや、ソフトウェアをアップグレードするときに便利です。

また、Mobile Link では、リモート・データと中央のデータ・ソースとを同期させるときに、ダイレクト・ロー・ハンドリングを実行できます。同期可能なデータ・ソースには、アプリケーション、Web サーバ、Web サービス、アプリケーション・サーバ、テキスト・ファイル、スプレッドシート、非リレーショナル・データベース、または統合データベースとしてサポートされていない RDBMS を指定できます。

### ● QAnywhere - モバイル・エンタープライズ・メッセージングとモバイル Web サービス

ス QAnywhere では、分散されたモバイル・ユーザにセキュアで保証されたメッセージ配信を実現する、アプリケーション間の総合モバイル・メッセージング・ソリューションを提供します。Mobile Link サーバを拡張することで、QAnywhere はモバイル・アプリケーションとエンタープライズ・システム間で、確実にメッセージを送受信します。

QAnywhere は次の機能を備えています。

- スマートクライアント・アプリケーション用のセキュアな蓄積転送メッセージング
- ネットワーク・フォールト・トレラントな、信頼できる通信
- ネットワークに依存しない通信
- 接続頻度の少ないデバイスとメッセージベースのエンタープライズ・システム間におけるルールベースの保証されたメッセージ配信
- エンタープライズ・システムとモバイルのデータベース・アプリケーションを統合するためのオプションを拡張できる、JMS (Java Message Service) との簡単な統合
- モバイル Web サービス

### ● SQL Remote - レプリケーション

SQL Remote は、統合データベースと、一般的には数多くのモバイル・データベースを含む多数のリモート・データベースとの間で双方向同期を行うためのデータレプリケーション・テクノロジーです。

SQL Remote では、蓄積転送アーキテクチャをベースに、FTP や電子メールなどのファイルやメッセージの転送メカニズムを使用して、データを同期させます。

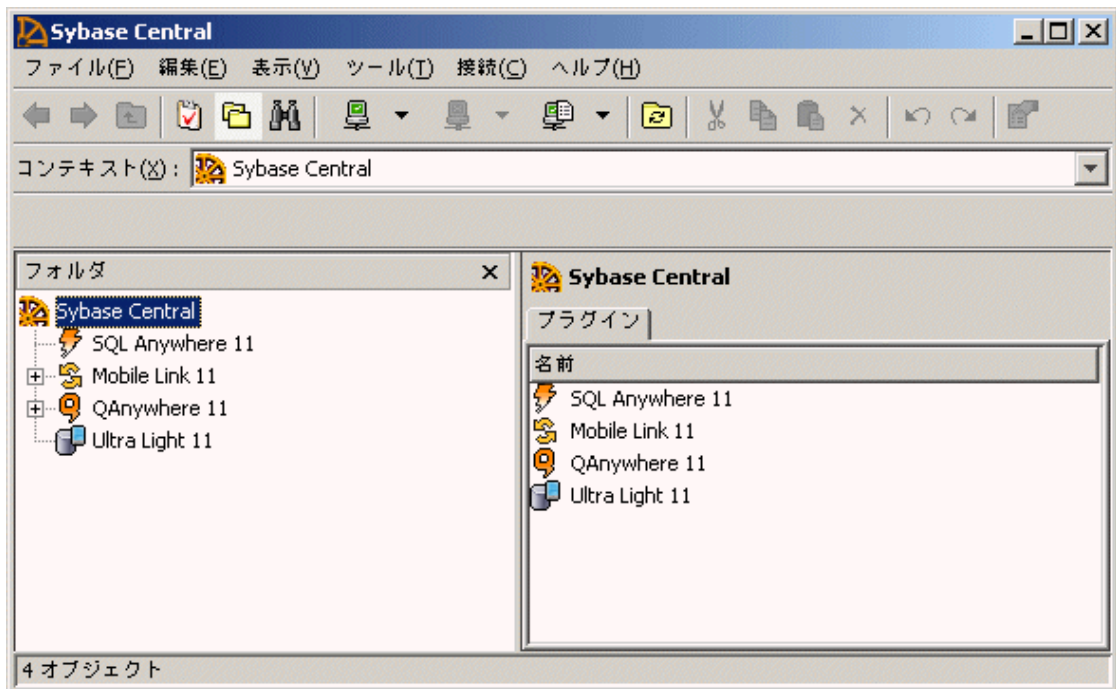
SQL Remote は、トランザクションの整合性を確保するので、多くのビジネス・アプリケーション、特に接続が不安定な環境で運用するアプリケーションに最適です。また、レプリケーション・システムのすべてのコンポーネントに必要なメモリとディスク領域は最小限に抑えられています。

「[SQL Remote の概要](#)」 [『SQL Remote』](#) を参照してください。

## 設計ツールと管理ツール

ここでは、SQL Anywhere に付属の設計ツールと管理ツールについて説明します。

- **Sybase Central - 集中制御と管理** Sybase Central は、データベースの設定、プロパティ、ユーティリティにグラフィカル・ユーザ・インタフェースからアクセスできる、統合データベース管理および開発ツールです。プラグインを使用することによって、SQL Anywhere サーバ、Mobile Link、QAnywhere、Ultra Light、その他の Sybase 製品の管理に Sybase Central を使用できます。たとえば、Mobile Link プラグインには、Mobile Link アプリケーションの開発とカスタマイズを支援する同期モデル作成ウィザードとモデル・モードが用意されています。



Sybase Central は、ルーチン・タスクの支援だけでなく、パフォーマンス統計、プロシージャ・プロファイリング、ストアド・プロシージャのデバッグ、さらにはイベントとスケジュール、Web サービス、接続プロファイルの管理機能を提供します。Sybase Central を使用すると、SQL 文をデータベース・サーバに送信することによって、または SQL Anywhere のユーティリティによって実行されたタスクを管理できます。[「Sybase Central の使用」](#) 『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

SQL Anywhere データベースの現在のパフォーマンスを分析し、モニタリングするのに役立つさまざまな Sybase Central ツールが用意されています。これらのツールには、プロシージャ・プロファイリング、グラフィカルなプラン、クエリの実行、パフォーマンス・モニタ、要求ロギング、タイミング・ユーティリティなどがあります。また、Sybase Central では次の機能も提供されます。

- **アプリケーション・プロファイリング・ウィザードを使用したアプリケーション・プロファイリング** Sybase Central のアプリケーション・プロファイリング・ウィザードを使用すると、次の操作を自動的に行うことができます。
  - ストアド・プロシージャ、関数、イベント、トリガのプロファイル
  - データベース・アプリケーションのパフォーマンス向上に役立つ推奨内容の受信
  - アプリケーション実行中のデータベース・アクティビティの取得

「[アプリケーション・プロファイリング](#)」『[SQL Anywhere サーバ - SQL の使用法](#)』を参照してください。
- **アプリケーション・プロファイリング・モードでの詳細なアプリケーション・プロファイリング** Sybase Central でデータベース・トレーシング・ウィザードとアプリケーション・プロファイリング・モードを使用することで、次の処理の全体的なパフォーマンスが向上します。
  - データベース・パフォーマンス・カウンタに基づいて、キャッシュ・サイズとインデックスを調整する。
  - デッドロックの発生時にデッドロックを特定する。
  - ロックしているアクティビティを確認する。
  - 実行プランを検査する。
  - 診断とトラブルシューティングのために、アプリケーションの各文をトレースする。

「[診断トレーシングを使用した詳細なアプリケーション・プロファイリング](#)」『[SQL Anywhere サーバ - SQL の使用法](#)』を参照してください。
- **インデックス・コンサルタントを使用した、インデックスの選択および最適化** インデックス・コンサルタントは、負荷を分析し、パフォーマンスを最適化するためのインデックスの選択方法を示す推奨内容を提供します。インデックス・コンサルタントは、Sybase Central または Interactive SQL から実行できます。「[インデックス・コンサルタント](#)」『[SQL Anywhere サーバ - SQL の使用法](#)』を参照してください。
- **Interactive SQL - SQL クエリ・エディタ** Interactive SQL は、SQL 文を実行し、データベースのデータを表示する、データベース・ユーティリティです。プランのグラフィカル表示などの組み込みクエリ・エディタやその他のツールを使用すると、クエリの分析、トラブルシューティング、最適化ができます。「[Interactive SQL の使用](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。
- **SQL Anywhere モニタ** SQL Anywhere モニタは、SQL Anywhere データベースや Mobile Link サーバの正常性や可用性に関する情報を示す、ブラウザベースの管理ツールです。データの常時収集、電子メールによる警告の通知、ブラウザベースのインタフェースがあり、複数のデータベースや Mobile Link サーバをモニタリングできます。「[SQL Anywhere モニタ](#)」『[SQL Anywhere サーバ - データベース管理](#)』と「[Mobile Link 用 SQL Anywhere モニタ](#)」『[Mobile Link - サーバ管理](#)』を参照してください。
- **Mobile Link モニタ - 同期のモニタ** Mobile Link モニタは、Mobile Link の同期のパフォーマンスに関する情報を示す、グラフィカルな管理ツールです。Mobile Link モニタは、開始時間と終了時間、アップロードまたはダウンロードされたデータ容量、正常に終了した同期、競



合が発生した同期など、発生したすべての同期に関する詳細情報と統計の概要を収集します。  
「[Mobile Link モニタ](#)」 「[Mobile Link - サーバ管理](#)」を参照してください。

- **データベース・ユーティリティ** SQL Anywhere には、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の、さまざまなユーティリティが用意されています。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。「[データベース管理ユーティリティ](#)」 「[SQL Anywhere サーバ - データベース管理](#)」を参照してください。
- **InfoMaker** InfoMaker はパーソナル・データ・アシスタントです。このツールを使用して、さまざまな方法でデータを処理できます。InfoMaker を使用すると、SQL Anywhere データベースのクエリを実行し、高度で効果的なカスタム・レポートを作成できます。「[InfoMaker について](#)」 72 ページを参照してください。
- **PowerDesigner Physical Data Model** Sybase PowerDesigner の Physical Data Model コンポーネントを使用すると、データベースの設計、生成、ドキュメント化、管理ができます。「[PowerDesigner Physical Data Model について](#)」 72 ページを参照してください。
- **DataWindow.NET** DataWindow.NET を使用すると、複雑なビジネス・ルールを取り入れ、高度なデータ提示が可能な、エンタープライズ・レベルの SQL Anywhere アプリケーションの構築と配備をすばやく実行できます。「[DataWindow .NET について](#)」 72 ページを参照してください。

## エディションとライセンス

SQL Anywhere には、さまざまな「エディション」があります。各エディションには、それぞれ特定のライセンスを別途必要とするコンポーネントが含まれます。また、データベース・サーバで使用する CPU 数に制約がある場合があります。エディションの詳細については、<http://www.sybase.jp/products/databasemanagement/sqlanywhere/editions> を参照してください。

別途ライセンスが必要なコンポーネントの詳細については、「[別途ライセンスが必要なコンポーネント](#)」 11 ページを参照してください。

### ライセンスと CPU 数

パーシット・ライセンスでは、-gt オプション、または実行しているエディションによる制限がデータベース・サーバにある場合を除き、コンピュータにあるすべての CPU がネットワーク・データベース・サーバで使用されます。CPU ベースのライセンスでは、-gt オプション、または実行している SQL Anywhere のエディションによる制限がデータベース・サーバにある場合を除き、ライセンスを受けた CPU 数までネットワーク・データベース・サーバで使用されます。

パーソナル・データベース・サーバは CPU が 1 つに制限されています。

### 参照

- 「[-gt サーバ・オプション](#)」 「[SQL Anywhere サーバ - データベース管理](#)」



## 別途ライセンスが必要なコンポーネント

次のコンポーネントは別途ライセンスが必要です。Sybase iAnywhere に注文してからインストールしてください。別途ライセンスが必要なコンポーネントの注文については、<http://www.iAnywhere.jp/sas/price.html> を参照してください。

また、SQL Anywhere には、それぞれ特定のライセンスを別途必要とするコンポーネントが含まれるさまざまなエディションがあります。「エディションとライセンス」 10 ページを参照してください。

### SQL Anywhere のセキュリティ・オプション

SQL Anywhere を使用すると、データベース・ファイルや、同期およびクライアント／サーバ通信のトランスポート・レイヤを強力に暗号化できます。

SQL Anywhere では次の強力な暗号化アルゴリズムをサポートしています。

	別途ライセンスが必要なセキュリティ・オプションに付属 <sup>1</sup>	SQL Anywhere に付属 <sup>2</sup>
データベースの暗号化	FIPS 認定の AES	AES
トランスポート・レイヤ・セキュリティ	FIPS 認定の RSA	RSA
トランスポート・レイヤ・セキュリティ	ECC	

<sup>1</sup> ECC テクノロジーまたは FIPS 認定テクノロジーを使用した強力な暗号化ソフトウェアは、別途注文する必要があります。

<sup>2</sup> AES および RSA による強力な暗号化は SQL Anywhere に付属しており、別途ライセンスは不要ですが、ライブラリは FIPS 認定ではありません。

セキュリティ・オプションでは、暗号化アルゴリズムを実装する Certicom DLL と、SQL Anywhere ソフトウェアと Certicom ライブラリとのインタフェースの役目を果たす追加 DLL が用意されています。

SQL Anywhere には、次のセキュリティ・オプションが用意されています。

- Windows オペレーティング・システム向け Certicom Security Builder GSE。  
詳細については、<http://csrc.nist.gov/cryptval/140-1/140val-all.htm> の 542 番を参照してください。
- Windows Mobile オペレーティング・システム向け Certicom Security Builder GSE。  
詳細については、<http://csrc.nist.gov/cryptval/140-1/140val-all.htm> の 316 番を参照してください。

暗号化の詳細については、「安全なデータの管理」 『SQL Anywhere サーバ - データベース管理』を参照してください。

### SQL Anywhere CAC 認証オプション

Ultra Light データベースについては、CAC 認証のライセンスを別途注文する必要があります。また、このオプションを使用するには、SQL Anywhere セキュリティ・オプションが必要です。

### SQL Anywhere の高可用性オプション

SQL Anywhere データベースでは、Veritas Cluster Server エージェント、またはフェールオーバー用にデータベースのミラーリングを使用するのに、別途ライセンスを注文する必要があります。

プラットフォームのサポートの詳細については、「[SQL Anywhere がサポートするプラットフォームおよびエンジニアリング・サポート状況](#)」を参照してください。

データベース・ミラーリングの詳細については、「[データベース・ミラーリングの概要](#)」  
『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

SQL Anywhere Veritas Cluster Server エージェントの詳細については、「[SQL Anywhere Veritas Cluster Server エージェントの使用](#)」  
『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

### SQL Anywhere のイン・メモリ・モード・オプション

SQL Anywhere データベースでイン・メモリ・モードを使用するには、ライセンスを別途注文する必要があります。

プラットフォームのサポートの詳細については、「[SQL Anywhere がサポートするプラットフォームおよびエンジニアリング・サポート状況](#)」を参照してください。

イン・メモリ・モードの詳細については、「[-im サーバ・オプション](#)」  
『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

### Mobile Link の高可用性オプション

Mobile Link の高可用性オプションを使用すると、共有ステータス・モードを使用する同一のサーバからなるサーバ・ファームに、複数の Mobile Link サーバをグループ化できます。Mobile Link を共有ステータス・モードで実行すると、同じリモート・データベースが同時に複数のサーバと同期されるのをブロックできるので、データ整合性が確保されます。Mobile Link の共有ステータス・サポートにより、サーバ起動同期の負荷分散とフェールオーバーも可能になります。

Mobile Link の高可用性オプションを使用すると、-ss オプションを指定して、複数の Mobile Link サーバを共有ステータス・モードで実行できます。「[-ss オプション](#)」  
『[Mobile Link - サーバ管理](#)』を参照してください。

サーバ・ファームでの Mobile Link の実行については、「[Mobile Link サーバをサーバ・ファームで実行する](#)」  
『[Mobile Link - サーバ管理](#)』を参照してください。

### Log Transfer Manager (LTM)

Sybase Replication Server 用の SQL Anywhere Replication Agent である LTM は、Sybase Replication Server のインストール環境にプライマリ・サイトとして参加しているすべての SQL Anywhere データベースに必要です。LTM にはライセンスが必要です。このライセンスは別途注文してください。

プラットフォームのサポートの詳細については、「[SQL Anywhere がサポートするプラットフォームおよびエンジニアリング・サポート状況](#)」を参照してください。

## SQL Anywhere 11 の特長

SQL Anywhere の特長を次に示します。

- **埋め込み性** SQL Anywhere は、他のアプリケーションに簡単に埋め込むことができます。高パフォーマンスでありながら、メモリの占有容量は非常に小さくて済みます。SQL Anywhere には、コンピュータ・リソースの最適化、パフォーマンス向上のためのセルフチューニング、リモートでのインストールとサポートの簡略化を可能にする機能など、フロントエンド環境での自己管理とメンテナンスを実現するためのさまざまな機能が備わっています。
- **相互運用性** SQL Anywhere は、Windows、Windows Mobile、Linux、Sun Solaris、HP-UX、IBM AIX、Macintosh など、多くのプラットフォームで使用できます。プラットフォーム間でデータベース・ファイルをコピーできるのは、SQL Anywhere だけです。さらに、SQL Anywhere は、小型デバイス用 Ultra Light データベース・テクノロジの使用により、BlackBerry、Embedded Linux、Windows Mobile 6、Java ME スマートフォンもサポートします。SQL Anywhere は、ODBC、JDBC、ADO.NET、PHP、Perl など、多くの一般的なデータベース・インタフェースをサポートしています。これにより、Microsoft Visual Studio、PowerBuilder、Eclipse、各種 Web ツールなどのさまざまなアプリケーション開発ツールを使用できます。ストアド・プロシージャは、C/C++、Java、.NET、または Perl で記述できます。
- **購入後すぐに実現できる性能** SQL Anywhere は、調整や管理なしに高性能を発揮できるように設計されています。動的キャッシュ・サイズ決定、統計の自動生成、高度なクエリ・最適化、クエリの並行処理、マテリアライズド・ビューなどの機能が備わっているので、SQL Anywhere は、高いパフォーマンスが必要とされるにもかかわらず常駐のデータベース管理者がいない環境に最適です。OLAP (オンライン分析処理) を備えているため、SQL Anywhere では単一の SQL 文内で複雑なデータ分析を実行できます。また、データベースでクエリの量を減らすことでパフォーマンスを向上しながら、結果の値を増やすことができます。
- **Web 操作** 組み込みの HTTP サーバと Web サービス・サポート、XML 機能、全文検索、PHP インタフェースを備えた SQL Anywhere は、Web サーバの背後にある Web ベースの環境で使用するデータベースとして最適です。
- **可動性** SQL Anywhere は、エンタープライズ・システムとの接続を確立できるかどうかにかかわらず、フロントエンドのシステムとデバイスで運用できる、エンタープライズ・クラスのデータベースを提供します。SQL Anywhere の同期テクノロジによって、ワイヤレス・ネットワークや有線ネットワークを経由して、バックエンドのデータベース、アプリケーション・サーバ、メッセージング・システムとのデータ交換を効率よく行えます。
- **セキュリティ** SQL Anywhere は、データベース・テーブル、ファイル、アプリケーションとデータベース間の通信ストリーム、Mobile Link の同期ストリームを 128 ビットの強力な暗号化で保護し、完全なエンドツーエンド・セキュリティを提供します。SQL Anywhere は、データへのアクセスを監査でき、組み込みのユーザ認証を備え、サード・パーティの認証システムと統合できます。また、別途ライセンスが必要なセキュリティ・オプションを使用すると、FIPS 認定の暗号化も使用できます。[「SQL Anywhere のセキュリティ・オプション」 11 ページ](#)を参照してください。

## サポートされるプラットフォーム

「SQL Anywhere がサポートするプラットフォームおよびエンジニアリング・サポート状況」ページには、SQL Anywhere がサポートするオペレーティング・システム・プラットフォームのリストと、SQL Anywhere 旧バージョンのエンジニアリング・サポート状況を表示します。

この Web サイトには、ソフトウェアの最新のビルドでのサポートが反映されており、現在使用中のソフトウェアとは情報が一致しないことがあります。<http://www.ianywhere.jp/sas/os.html>

このページから次の各ページにアクセスして、SQL Anywhere の各コンポーネントに関するプラットフォーム別の詳細情報を確認できます。

- **プラットフォーム別 SQL Anywhere コンポーネント** <http://www.ianywhere.jp/sas/os.html>
- **<http://www.ianywhere.jp/sas/os.html>** <http://www.ianywhere.jp/sas/os.html>
- **SQL Anywhere がサポートする Kerberos クライアント** <http://www.ianywhere.jp/sas/os.html>
- **SQL Anywhere がサポートする Listener プラットフォーム** <http://www.ianywhere.jp/sas/os.html>
- **SQL Anywhere の Mobile Link リダイレクタがサポートする Web サーバ** <http://www.ianywhere.jp/sas/os.html>

ソフトウェアの更新については、「ソフトウェア更新のチェック」『SQL Anywhere サーバ-データベース管理』を参照してください。

## アクセシビリティ有効化オプション

SQL Anywhere 11 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用意されています。このコンポーネントには **Sun Access Bridge** 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。画面リーダーなどのサード・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。

アクセシビリティ有効化オプションのプラットフォーム・サポートについては、<http://www.iAnywhere.jp/sas/os.html> を参照してください。

SQL Anywhere のアクセシビリティの詳細については、<http://www.sybase.com/accessibility> を参照してください。

---

# データ管理テクノロジーの概要

## 目次

データベース・システムの各部分 .....	18
リレーショナル・データベースの概念 .....	20
SQL Anywhere の内部構造 .....	25
SQL Anywhere データベースと Ultra Light データベースとの比較 .....	29
データベースの使用例 .....	30
多層コンピューティングのアーキテクチャ .....	32
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 .....	33
ETL 機能 .....	34
プログラミング・インタフェース .....	35

---

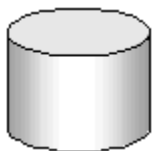
## データベース・システムの各部分

「リレーショナル・データベース管理システム」(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。

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
- データベース・サーバ
- アプリケーション・プログラミング・インタフェース (API)
- クライアント・アプリケーション

**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。



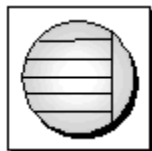
SQL Anywhere データベースは、通常は拡張子 *.db* が付いた 1 つのファイルです。Ultra Light データベースも、通常は拡張子 *.udb* が付いた 1 つのファイルです。SQL Anywhere には、ユーザが操作できるサンプル・データベースが SQL Anywhere サンプル・ディレクトリに含まれています。このデータベースは、*samples-dir¥demo.db* です。

*samples-dir* のデフォルト・ロケーションについては、「[サンプル・ディレクトリ](#)」『SQL Anywhere サーバ - データベース管理』を参照してください。

**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。データベースには、常にデータベース・サーバ経由でアクセスされます。

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。データベースを管理するのは、一度に 1 つのサーバだけです。ただし、SQL Anywhere データベース・サーバは、一度に複数のデータベースを管理できます。

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。



SQL Anywhere には、「パーソナル・サーバ」と「ネットワーク・サーバ」の 2 つのデータベース・サーバがあります。どちらのサーバも同じクエリ処理やその他の内部オペレーションを提供しており、唯一の違いは、サーバが受け入れる接続の数と種類です。



パーソナル・サーバが同じコンピュータ上で稼働するアプリケーションやユーザから受け入れることができる同時接続数は、最大 10 までです。このサーバは、同一コンピュータを使用するシングルユーザ向けです。

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およびマルチコンピュータ向けです。最大接続数は、ライセンス契約に応じて異なります。

[「2 種類の SQL Anywhere データベース・サーバ」 25 ページ](#)を参照してください。

**Ultra Light ランタイム・ライブラリ** Ultra Light では、データベース・サーバによく見られるデータベース管理システムはインプロセス・ランタイム・ライブラリとして実装されます。ランタイム・ライブラリとアプリケーションは、同じプロセスの一部です。

**プログラミング・インタフェース** アプリケーションは、ODBC、JDBC、OLE DB、ADO.NET、Embedded SQL などのプログラミング・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。

SQL Anywhere と Ultra Light でサポートされているプログラミング・インタフェースの完全なリストについては、[「プログラミング・インタフェース」 35 ページ](#)を参照してください。

各プログラミング・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 Windows 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。

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表では、クライアント・アプリケーションは次のように示します。

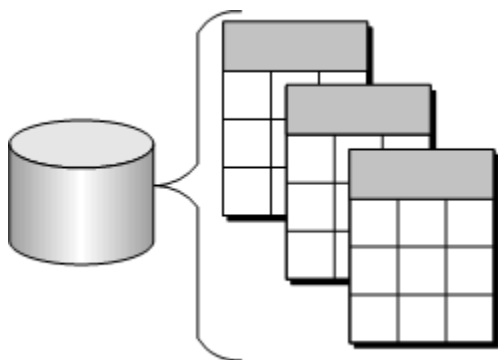


## リレーショナル・データベースの概念

次の各項では、基本的なリレーショナル・データベースのコンセプトについて簡単に説明します。テーブル、プライマリ・キーと外部キー、データベース・オブジェクトの定義についても説明します。

### 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。

EmployeeID	Surname	GivenName	Phone
102	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、dBase や FoxPro などの非リレーショナル・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、特定の種類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。

「[SQL Anywhere でのデータベースの作成](#)」 『[SQL Anywhere サーバ - SQL の使用法](#)』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最小限に抑えて効率的な方法で情報を保持できます。

## プライマリ・キー

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できません。

### 例

SQL Anywhere サンプル・データベース (*samples-dir\demo.db*) では、Employees テーブルに従業員の個人情報格納されています。このテーブルにはプライマリ・キー・カラム EmployeeID 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 番号の方が適しています。

SQL Anywhere サンプル・データベースの SalesOrderItems 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
- **LineID** 注文内での注文項目を識別する行番号
- **ProductID** 受注した製品を識別する製品 ID
- **Quantity** 受注した項目の数量
- **ShipDate** 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 LineID 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 LineID 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 LineID 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。

## 外部キー

1 つのテーブル内の情報は、「外部キー」によって他のテーブルの情報と関連付けられます。

### 例

SQL Anywhere サンプル・データベースには、従業員 (employee) 情報が入ったテーブルが 1 つと、部署 (department) 情報の入ったテーブルが 1 つあります。Departments テーブルには、次のカラムがあります。

- **DepartmentID** 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。
- **DepartmentName** 部署の名前
- **DepartmentHeadID** 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 Employees テーブルに入力しておく必要はありません。その代わりに Employees テーブルには、Departments テーブルの DepartmentID 値の 1 つと一致する番号の入ったカラムがあります。

Employees テーブルの DepartmentID カラムは、Departments 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、Employees 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」と呼びます。Departments 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」と呼びます。

## 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する必要がありますが、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。[「インデックス」](#) [『SQL Anywhere サーバ - SQL の使用法』](#)を参照してください。

- **テキスト・インデックス** テキスト・インデックスには、インデックス化されたすべてのカラム内の全単語の出現箇所の詳細な位置情報が含まれます。全文検索を実行する場合、テキスト・インデックスを使用して、一致するローが検索されます。このため、テキスト・インデックスを使用したクエリは、テーブル内のすべての値をスキャンする必要があるクエリよりも高速になる可能性があります。[「テキスト・インデックス」](#) [『SQL Anywhere サーバ - SQL の使用法』](#)と[「全文検索」](#) [『SQL Anywhere サーバ - SQL の使用法』](#)を参照してください。

- **ログイン・ポリシー** ログイン・ポリシーは、ユーザのデータベース接続を作成するときに適用されるルール・セットで構成されます。[「ログイン・ポリシーの管理の概要」](#) [『SQL Anywhere サーバ - データベース管理』](#)を参照してください。

- **ビュー** ビューは、一時的なテーブルです。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。

[「ビューの操作」](#) [『SQL Anywhere サーバ - SQL の使用法』](#)を参照してください。

- **マテリアライズド・ビュー** SQL Anywhere はマテリアライズド・ビューもサポートしています。マテリアライズド・ビューとは、ベース・テーブルとよく似ていて、結果セットが計算されてディスクに格納されるビューです。概念としては、マテリアライズド・ビューはビューでもあり (クエリ指定がある)、テーブルでもあります (永続的な実体化したローがある)。したがって、テーブルで実行する多くの操作は、マテリアライズド・ビューでも実行できます。たとえば、マテリアライズド・ビューに対して、インデックス構築やアンロードを実行できます。

マテリアライズド・ビューは、データベースが大きく、頻繁なクエリで大量のデータに対して集約操作とジョイン操作が発生し、かつ最新のデータにアクセスすることが重大な要件ではないような環境に最適です。[「マテリアライズド・ビューの操作」](#) [『SQL Anywhere サーバ - SQL の使用法』](#)を参照してください。

- **ストアド・プロシージャとトリガ** これらは、データベースに保持され、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・プロシージャは、パラメータと結果セットを受け入れることができます。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。

データベースには、Java クラスも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します。[「SQL Anywhere で使用する Java クラスの作成」](#) [『SQL Anywhere サーバ - プログラミング』](#)を参照してください。

[「プロシージャ、トリガ、バッチの使用」](#) [『SQL Anywhere サーバ - SQL の使用法』](#)を参照してください。

- **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。

[「ユーザ ID、権限、パーミッションの管理」](#) [『SQL Anywhere サーバ - データベース管理』](#)を参照してください。

これらの共通のデータベース・オブジェクトに加え、SQL Anywhere は次の高度な機能を提供します。

- イベント
- ドメイン
- パブリケーション
- Web サービス
- リモート・データ・アクセス
- メンテナンス・プラン

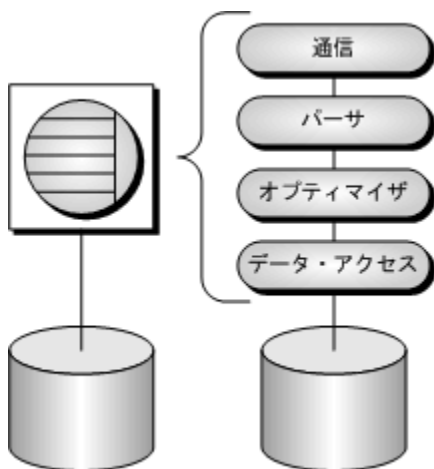
## SQL Anywhere の内部構造

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データベース・サーバとデータベースの対話方法をよく理解できます。

### データベース・サーバの内部構造

SQL 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。オブティマイザは、必要なデータをすばやく取得するのに最適な方法を選択します。
- 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。



### 2 種類の SQL Anywhere データベース・サーバ

SQL Anywhere には、「パーソナル・サーバ」と「ネットワーク・サーバ」の 2 つのデータベース・サーバがあります。



これらの2種類のサーバでは、同じ要求処理エンジンが使用され、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。ただし、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、複数のコンピュータ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。その他の相違点は次のとおりです。

- 「パーソナル・サーバ」は、同じコンピュータ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、同一コンピュータ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。

- Windows 上では *dbeng11.exe* (Windows Mobile では使用できません)。
- UNIX オペレーティング・システム上では *dbeng11*。

- 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。

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。

- Windows (Windows Mobile を含む) 上では *dbsrv11.exe*。このネットワーク・サーバは Windows Mobile 用にも提供されているので、デスクトップ・アプリケーションからモバイル・デバイス上のデータベースに接続できます。
- UNIX オペレーティング・システム上では *dbsrv11*。

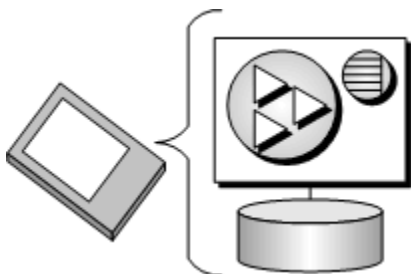
他の製品の一部として SQL Anywhere を入手した場合は、2つのデータベース・サーバのどちらかのバージョンが付属していないことがあります。同様に、すべてのオペレーティング・システムですべてのコンポーネントが利用できるわけではありません。たとえば、Windows Mobile で使用できるのはネットワーク・サーバだけで、パーソナル・サーバは使用できません。

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・サーバの実行の詳細については、「[データベース・サーバの実行](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

### Ultra Light の内部構造

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 を利用できます。Ultra Light では、サーバによって実行される機能は、一般的にランタイム・ライブラリに置かれています。ランタイム・ライブラリは、アプリケーションと結合され、同じプロセスの一部になっています。このため、データベースとアプリケーションは1対1の関係になっています。





複数のアプリケーションから同じデバイスの 1 つのデータベースに同時接続する必要がある配備の場合、ライブラリは独立した処理として存在する必要があります。このような場合は、Ultra Light データベース・エンジンが使用されます。

### その他の機能

- Ultra Light には、Mobile Link 同期テクノロジー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 および Mobile Link の統合の詳細については、「[Ultra Light クライアント](#)」『[Ultra Light データベース管理とリファレンス](#)』を参照してください。

- Ultra Light は、多く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 の概要](#)」『[Ultra Light データベース管理とリファレンス](#)』を参照してください。

## データベース ファイル

次の項からは、データベースを構成するファイルの種類（データベース・ファイル、トランザクション・ファイル、テンポラリ・ファイルなど）について簡単に説明します。SQL Anywhere と Ultra Light におけるこれらのファイルの実装の違いについても説明します。

### SQL Anywhere のデータベース・ファイル

SQL 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 台のコンピュータから別のコンピュータにコピーできます。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。

SQL 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トランザクション・ログとテンポラリ・ファイルの 2 つのファイルを使用します。

- **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、トランザクション・ログの名前は、拡張子

が `.db` ではなく `.log` であ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 の同期では、トランザクション・ログを使用して、他のデータベースに変更内容を同期させます。
- **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作成され、データベース・サーバが停止すると消去されます。その名前が示すように、テンポラリ・ファイルとはデータベース・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。

Ultra Light のテンポラリ・ファイルは、データベース・ファイルと同じディレクトリに保存されます。

「[TMP、TEMPDIR、TEMP 環境変数](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

### Ultra Light データベースの内部構造

Ultra Light データベースには、これまで説明した機能を備えていますが、次の例外があります。

- Ultra Light のデータベース・ファイルには、データベース内のデータの分散に関する情報が含まれません。
- Ultra Light は、別のログ・ファイルではなく、内部でトランザクションを追跡します。
- Ultra Light のテンポラリ・ファイルは、データベース・ファイルと同じディレクトリに保存されます。

「[Ultra Light のトランザクションとステータスの管理](#)」『[Ultra Light データベース管理とリファレンス](#)』を参照してください。

## SQL Anywhere データベースと Ultra Light データベースとの比較

SQL Anywhere と Ultra Light は、大規模な企業データベース・ソースからモバイル・データベースに至る、データ・ストレージとデータ・アクセスのニーズに対応しています。アプリケーションの設計時に、2 つの中から最適なデータベースを選択する必要があります。

- ターゲット・プラットフォームが UNIX または Mac OS X の場合は、SQL Anywhere データベースを使用してください。
- ターゲット・プラットフォームが Palm OS、BlackBerry、Embedded Linux、Windows Mobile 6、または Java ME スマートフォンの場合は、Ultra Light データベースを使用してください。
- ターゲット・プラットフォームが Windows Vista、Windows XP、または Linux の場合は、SQL Anywhere と Ultra Light のどちらも使用できます。SQL Anywhere の方が、フル機能を使用でき、追加で必要なメモリもそれほど問題にはならないので、よく使用されます。
- ターゲット・プラットフォームが Windows Mobile (Pocket PC やスマートフォンなどのデバイス) の場合、メモリの制約や、場合によってはアプリケーションで実行するタスクを考慮する必要があります。Windows Mobile 上では、SQL Anywhere は約 6 MB のメモリに加え、同期コンポーネント用に 2 MB のメモリを必要としますが、Ultra Light は 1 MB 以下のメモリしか必要とせず、同期機能も組み込まれています。しかし、Ultra Light はかなり小さいものの、SQL Anywhere と同じ機能 (複雑なクエリ、イベント、プロシージャ、トリガ、ビューなど) をサポートしているわけではありません。

総合的には、SQL Anywhere データベースと Ultra Light データベースを混在させ、Mobile Link を使用して同期させるソリューションを選択するのがベストかもしれません。

コアとなるデータベース・ソリューション (SQL Anywhere) と Ultra Light データベース・ソリューションの相違点の詳細については、「[Ultra Light の機能比較](#)」『[Ultra Light データベース管理とリファレンス](#)』を参照してください。

## データベースの使用例

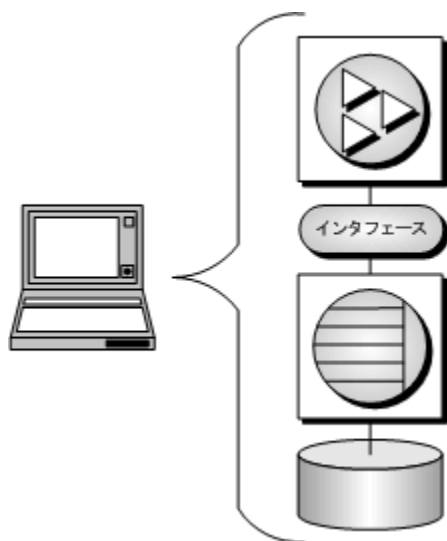
データベース・アプリケーションは、アプリケーションと同じコンピュータ上にあるデータベース・サーバに接続できます。ネットワーク・データベース・サーバの場合は、別のコンピュータ上にあるデータベース・サーバに接続できます。さらに、SQL Anywhere では、リモート・オフィスとモバイル・アプリケーション用に分散データベースを構築して、異なるコンピュータ上にある物理的に別個のデータベース間でデータを共有できます。

## デスクトップ・アプリケーションと組み込みデータベース

SQL 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、他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。

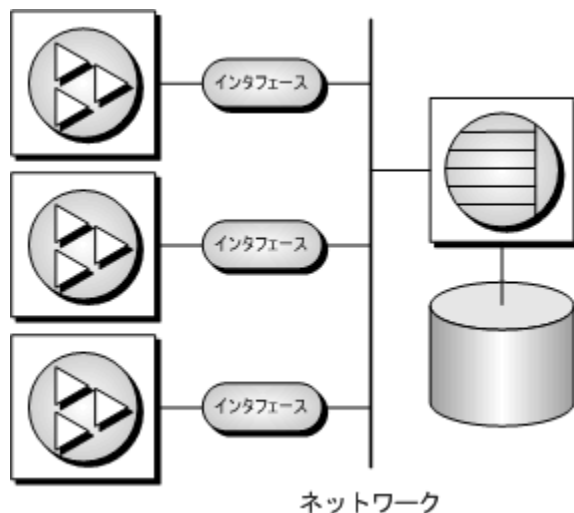
多くのリレーショナル・データベース管理システムでは、熟練したスタッフによる管理が求められます。SQL Anywhere データベースの特徴は、管理をまったく必要とせずに運用できることです。

SQL Anywhere パーソナル・データベース・サーバは、通常埋め込みアプリケーションとして使用します。埋め込み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コンピュータ上で動作するデータベース・サーバと接続しています。



## 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

SQL 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コンピュータをネットワーク・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・コンピュータ上にあります。



データベース・サーバとしては SQL 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には、追加の接続パラメータ (少なくともサーバ名と通信プロトコル) を指定する必要があります。

### 参照

- 「配備の種類」 『SQL Anywhere サーバ - プログラミング』
- 「SQL Anywhere データベース接続」 『SQL Anywhere サーバ - データベース管理』

## 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、WebLogic、または WebSphere) 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・ストアの他に、複数のデータベースにアクセスできます。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。最新の多層アプリケーションの多くは、Web サービスをベースにしたサービス指向アーキテクチャ (SOA) が使用されています。

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。

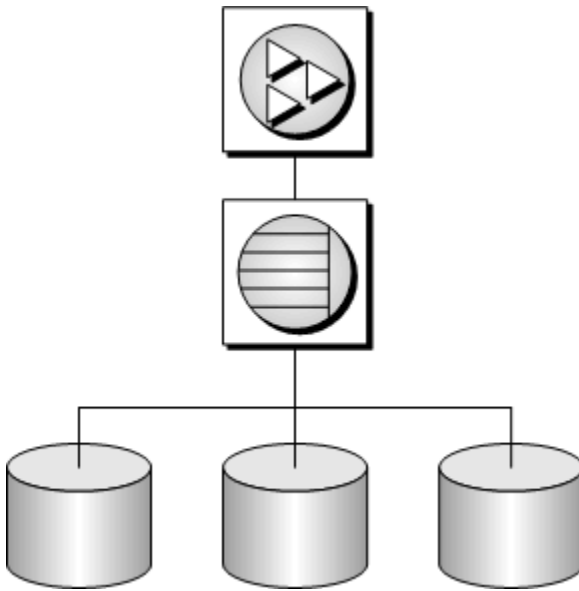
アプリケーション・サーバは、クライアント・アプリケーションにトランザクション論理も提供し、一連のオペレーションが複数のデータベース上で確実にアトミックに実行されるようにします。SQL Anywhere は、多層コンピューティングに最適であり、Microsoft 分散トランザクション・コーディネータ (DTC) で管理されている分散トランザクションにサーバとして使用できます。Sybase Enterprise Application Server と Microsoft Transaction Server のどちらも、DTC を使用してクライアント・アプリケーションにトランザクション・サービスを提供します。

SQL Anywhere では、Web サービスのサポートが組み込まれているので、多くの多層または SOA アプリケーションに適しています。

「3 層コンピューティングと分散トランザクション」 『SQL Anywhere サーバ - プログラミング』を参照してください。

## 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

SQL 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 や START DATABASE 文を使用してデータベースに接続しても開始できます。

### 参照

- 「SQL Anywhere データベース・サーバ」 『SQL Anywhere サーバ - データベース管理』
- 「DatabaseFile 接続パラメータ [DBF]」 『SQL Anywhere サーバ - データベース管理』
- 「START DATABASE 文」 『SQL Anywhere サーバ - SQL リファレンス』

## ほかのデータベースのデータへのアクセス

SQL Anywhere のリモート・データ・アクセス機能を使用して、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他の複数のデータベースにもアクセスできます。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。「リモート・データへのアクセス」 『SQL Anywhere サーバ - SQL の使用法』を参照してください。

## ETL 機能

ETL (Extract : 抽出、Transform : 変換、Load : ロード) とは、異なるデータ・ソースから大量のデータを抽出し、1 つのデータベースに統合するプロセスです。抽出フェーズでは、データは解析され、妥当性を評価されます。変換では、データは操作され、格納するためのフォーマットに変換されます。一般的に、変換では不要なカラムを削除し、計算値を計算し、データを統合できるように日付などの値を共通フォーマットに変換します。その後データは、組織のニーズに応じた頻度とスコープでデータベースにロードされます。

SQL Anywhere には、ETL をサポートするいくつかの機能があります。次に例を示します。

- **OPENSTRING 操作** FROM 句で OPENSTRING 操作を使用すると、クライアント側およびサーバ側のデータ・ソースからデータを変換およびロードできます。「[FROM 句](#)」『[SQL Anywhere サーバ - SQL リファレンス](#)』を参照してください。
- **openxml システム・プロシージャ** openxml システム・プロシージャを使用すると、XML ドキュメントからデータを抽出できます。「[openxml システム・プロシージャ](#)」『[SQL Anywhere サーバ - SQL リファレンス](#)』を参照してください。
- **MERGE 文** MERGE 文を使用すると、異なるソース・オブジェクトのデータをマージできます。「[MERGE 文](#)」『[SQL Anywhere サーバ - SQL リファレンス](#)』を参照してください。
- **プロキシ・テーブル** プロキシ・テーブルを使用すると、リモート・データベースのテーブル、ビュー、マテリアライズド・ビューなどのオブジェクトにアクセスできます。「[プロキシ・テーブルの操作](#)」『[SQL Anywhere サーバ - SQL の使用法](#)』を参照してください。
- **FROM 句のシステム・プロシージャ・コール** クエリの FROM 句でさまざまなシステム・プロシージャを使用することで、データをロードするためのデータ抽出やデータ変換を実行できます。SQL Anywhere で使用できるシステム・プロシージャのリストについては、「[システム・プロシージャ](#)」『[SQL Anywhere サーバ - SQL リファレンス](#)』を参照してください。



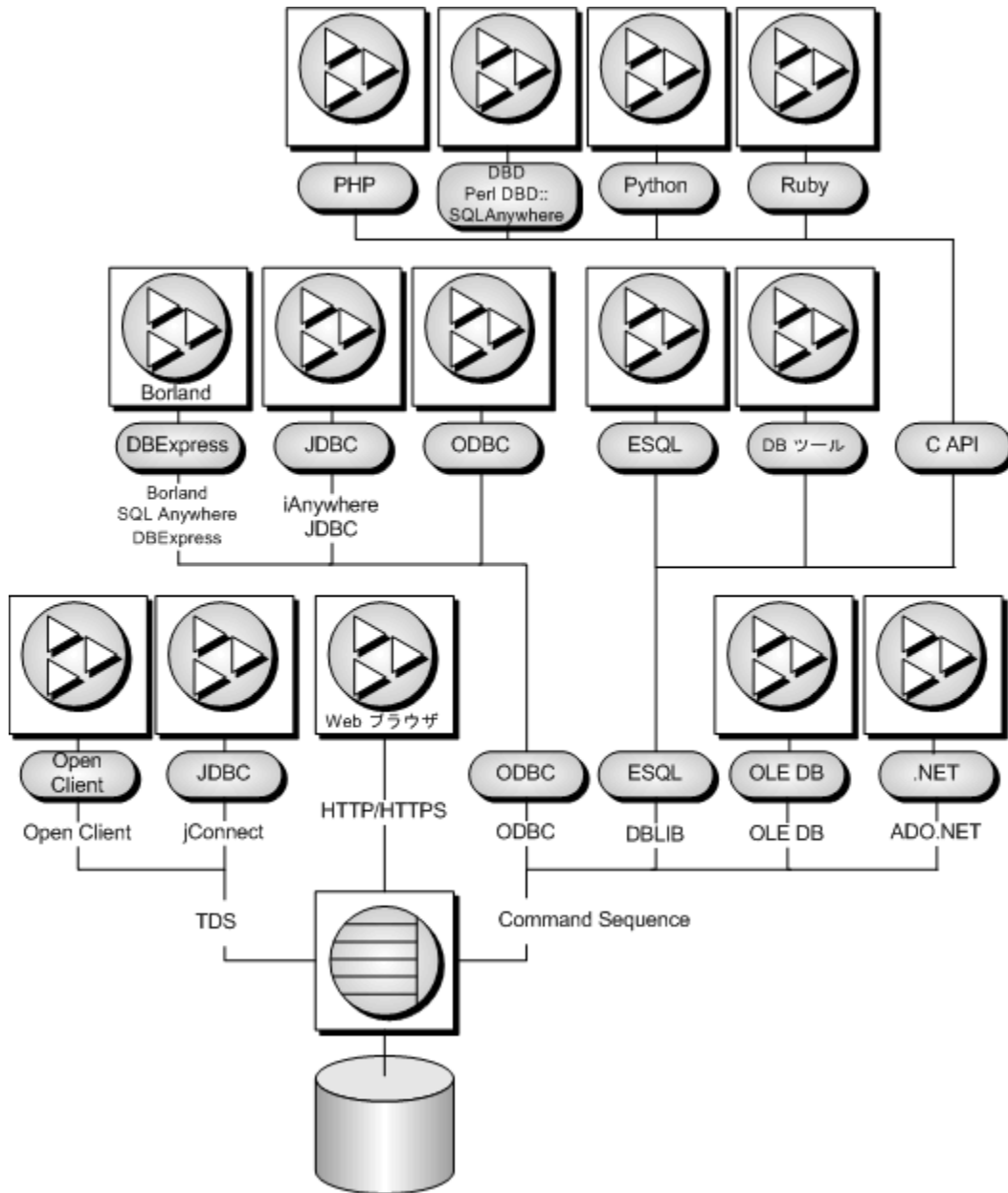
## プログラミング・インタフェース

SQL Anywhere では、さまざまなデータ・アクセス・プログラミング・インタフェースがサポートされており、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。

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、「[データベースの使用例](#)」 30 ページを参照してください。

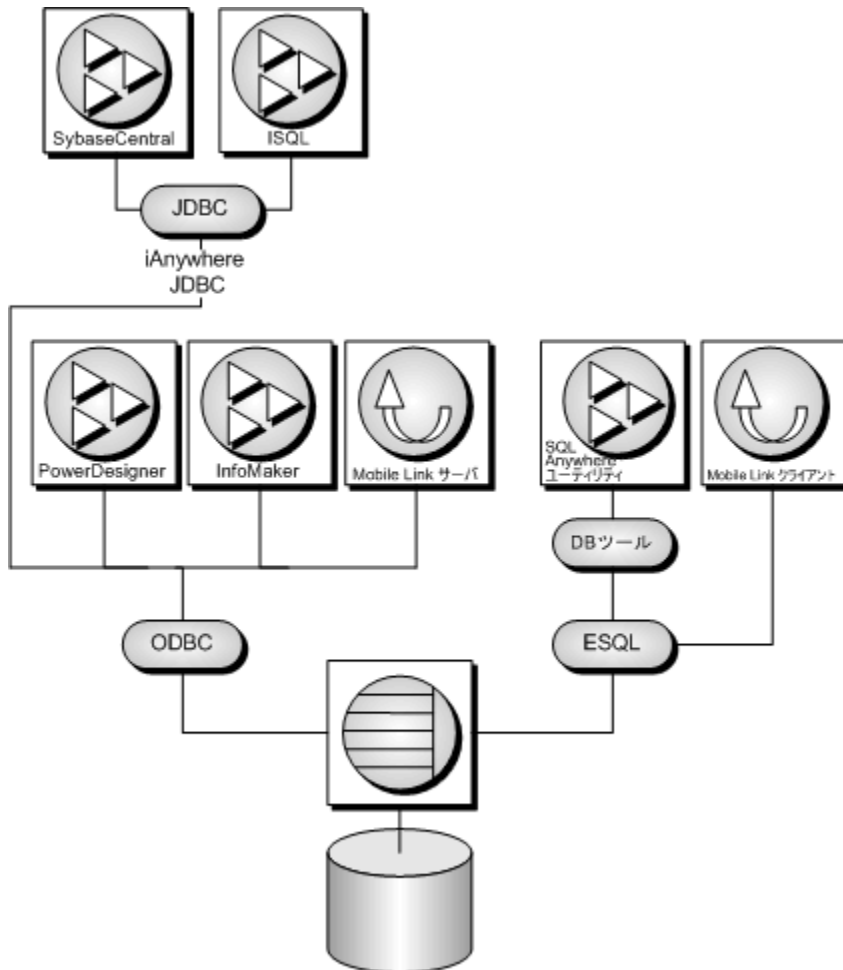
### サポートされるプログラミング・インタフェースとプロトコル

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 アプリケーション

SQL Anywhere 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。



## SQL Anywhere のプログラミング・インタフェース

SQL Anywhere のプログラミング・インタフェースの詳細については、次のリストを参照してください。

- 「SQL Anywhere の ODBC サポート」 『SQL Anywhere サーバ - プログラミング』
- 「SQL Anywhere の .NET サポート」 『SQL Anywhere サーバ - プログラミング』
- 「SQL Anywhere の OLE DB と ADO のサポート」 『SQL Anywhere サーバ - プログラミング』
- 「SQL Anywhere Embedded SQL」 『SQL Anywhere サーバ - プログラミング』
- 「SQL Anywhere の JDBC サポート」 『SQL Anywhere サーバ - プログラミング』
- 「Sybase Open Client のサポート」 『SQL Anywhere サーバ - プログラミング』
- 「SQL Anywhere の Perl DBI サポート」 『SQL Anywhere サーバ - プログラミング』
- 「SQL Anywhere の PHP サポート」 『SQL Anywhere サーバ - プログラミング』

## Ultra Light プログラミング・インタフェース

Ultra Light には、開発者がデータに簡単にアクセスできるプログラミング・インタフェースが複数用意されています。次に、Ultra Light のプログラミング・インタフェースのリストを示します。

- C/C++
- C/C++ を使用した Embedded SQL
- C# または VB.NET を使用した Ultra Light.NET
- JavaScript を使用した M-Business Anywhere

Ultra Light のプログラミング・インタフェースの詳細については、「[Ultra Light プログラミング・インタフェースの選択](#)」『[Ultra Light データベース管理とリファレンス](#)』を参照してください。

## 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。SQL Anywhere は、「Command Sequence」と「Tabular Data Stream (TDS)」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、接続ウィンドウが異なります。

**Command Sequence** このプロトコルは、SQL Anywhere、iAnywhere JDBC ドライバ、および Embedded SQL、ODBC、OLE DB、ADO.NET の各 API で使用されます。

**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。

---

# データ交換テクノロジーの概要

## 目次

同期テクノロジーの比較 .....	40
モバイル・エンタープライズ・メッセージング : QAnywhere .....	49

---

## 同期テクノロジーの比較

データ交換テクノロジーには、同期、レプリケーション、メッセージング、モバイル Web サービス・テクノロジーが含まれます。

データの「同期」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、同期システムの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、柔軟なアプリケーションのアーキテクチャが可能になります。

SQL Anywhere には、2つの同期テクノロジーが備わっています。

- **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一方向または双方向同期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ーです。さまざまな統合データベース・サーバをサポートし、その他のほぼすべてのデータ・ソースと同期できる API を提供します。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、Mobile Link はあらゆるモバイル・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** データベース・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ーです。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は最小限に抑えられているので、SQL Remote はモバイル・データベースに最適です。

次の表に、Mobile Link と SQL Remote の特徴を示します。

同期テクノロジー	データベースの数	接続	頻度	統合データベースの種類
Mobile Link	多い	随時	中	多種類
SQL Remote	多い	随時	低	SQL Anywhere

## Mobile Link の特徴

Mobile Link は、同期システム用に設計されたもので、以下の要件があります。

- **多数のリモート・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。膨大な数の同期を同時に処理できます。
- **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。
- **統合データベースのサポート** Mobile Link では、事実上どの種類のデータ・ソースも中央のデータ・ソースとして使用できます。リモートのデータ・ストアは、SQL Anywhere または Ultra Light データベースのいずれかである必要があります。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。

- **柔軟な同期スケジュール** アプリケーションは、秒、分、時間、または日単位の間隔で接続と同期を行います。

## SQL Remote の特徴

SQL Remote は、同期システム用に設計されたもので、以下の要件があります。

- **多数のリモート・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、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、システムにおいて、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、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、同時に複数のサイトのメッセージを準備することで、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、SQL Anywhere 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。

## データの同期の利点

### データ可用性

データ同期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。中央データベース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、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ことがありません。

### 応答時間

同期は、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。

## 同期テクノロジーの課題

どの同期テクノロジーの場合でも、同期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

同期システムの課題の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。

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、同期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、システムの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 でも、各トランザクションをレプリケートするかどうか選択できますが、デフォルトで Mobile Link では、リモート・データベースの複数のトランザクションが結合され、単一のトランザクションに適用されます。通常はこの仕組みによって、アップロードがより効率的になります。どちらの場合でも、Mobile Link ではトランザクションの整合性が維持されます。

### データの一貫性

同期システムのもう1つの課題は、システム全体のデータの一貫性を維持することです。同期システムは、システム全体として「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。

### 参照

- 「同期の方法」 『Mobile Link - サーバ管理』

## 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・データベースの間でデータの同期が行われます。

Mobile Link では、サポートされているいずれかの RDBMS を「統合データベース」として使用できます。一般的にコーポレート・サーバに置かれている統合データベースは、同期情報を追跡し、オプションとしてレプリケートされるデータを格納することができます。他の中央データは、非リレーショナル・データベース、Web サービス、テキスト・ファイルなど、任意の形式で保存されます。

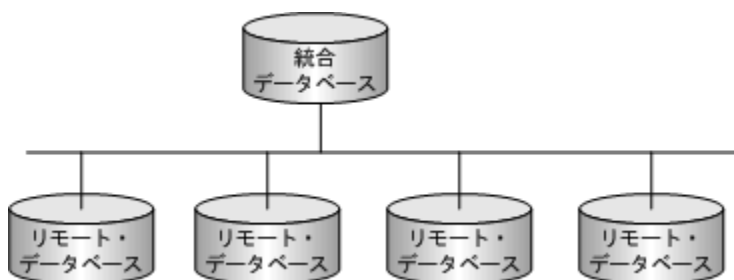
また、Mobile Link ではダイレクト・ロー・ハンドリングを実行できます。これにより、ERP (enterprise resource planning) システムやアプリケーション・サーバなど、リレーショナル・データベース以外の統合データ・ソースに対するデータの同期が実現されます。

SQL Remote では、同期されるすべてのデータが SQL Anywhere の統合データベースに含まれています。



「リモート・データベース」は、統合データベースと同じ場所でも、ハンドヘルド・デバイスのように物理的に離れた場所でも実行できます。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。

次の図は、小規模な同期システムを示したものです。



### リモート・ユーザ

一般的な同期システム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、中央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またはモバイル・デバイス上にあります。すべてのリモート・データベースは、中央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・コピーが中央データベースで保存されていれば、同期システム全体を、1つの分散データベースとみなすことができます。

中央データベースと同期する各リモート・サイトは、中央データベースのリモート・ユーザとみなされます。リモート・サイトがマルチユーザ・サーバの場合、サイト全体を中央データベースの単一のリモート・ユーザとみすことができます。

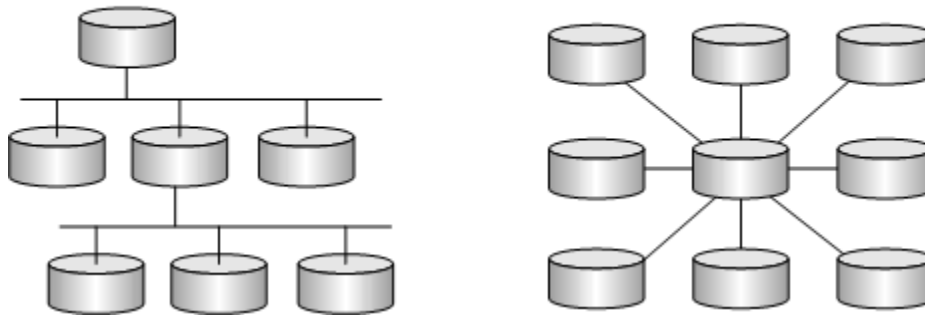
## 階層データベースの構成

「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・ピア型の同期など、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。

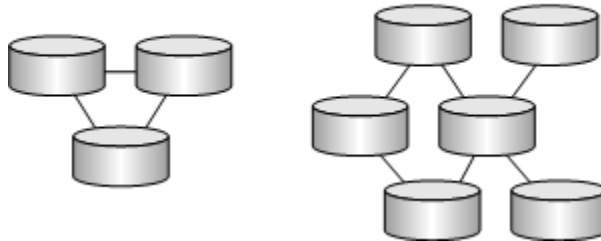
階層構成内で、別のデータベースと直接データを共有するデータベースはすべて、常にもう一方のデータベースの上位か下位に位置しています。

階層データベースの構成



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。

階層以外の構成



Mobile Link や SQL Remote のシステムでは、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、同期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。

## 双方向の同期

すべての SQL Anywhere の同期テクノロジーで、双方向同期がサポートされています。中央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、中央データベースや他のリモート・データベースに伝達されます。Mobile Link では、アップロード専用の同期、ダウンロード専用の同期、双方向同期を実行できます。

SQL Remote と Mobile Link の両方で、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、同期システムの他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同期 : Mobile Link

「セッションベース」または「同期的」な同期スキームでは、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できます。リモート・サイトは、秒、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 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、サーバは中央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同期 : SQL Remote

SQL Remote は、メッセージを使用してデータベース間のデータを交換する、「非同期」の同期スキームを使用しています。メッセージは、通常ファイルまたは特殊な形式の電子メール・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、他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。

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。

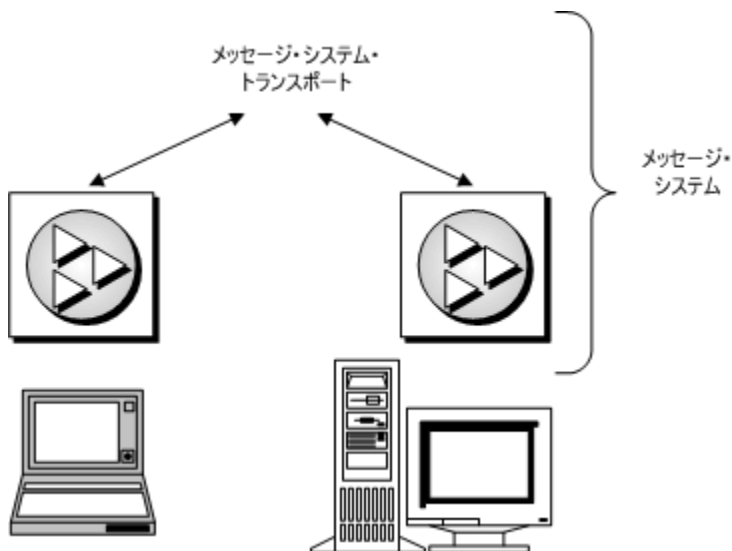
#### 蓄積転送方式を使用したメッセージ・サービス

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 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、インターネットのメール転送プロトコル (SMTP) 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・フォルダを開いてメールを読むまでそのメールを蓄積し、フォルダが開かれたときに、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同期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同期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

作業を確実にを行うために、メッセージベースの同期システムは、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、同期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。

### 接続ベースの同期

同期テクノロジーの中には、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。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、同期はわずか数秒程度のタイムラグで確実に行われます。

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ーが適しているのは、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・コンピュータがモバイルの場合や、接続頻度が低いような場合は、セッションベース (Mobile Link) やメッセージベース (SQL Remote) のテクノロジーの方が柔軟性の高いソリューションとなります。

SQL Anywhere を使用して接続ベースの同期システムを作成する場合は、Sybase Replication Server を使用する必要があります。[「Replication Server を使用したデータのレプリケート」](#)  
[『SQL Anywhere サーバ - データベース管理』](#)を参照してください。

## 同期テクノロジーの選択

SQL Anywhere の各同期テクノロジーは、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ー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ーを選択できます。

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

通常の同期環境では、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同期システムを調整しなければならない場合もあります。

SQL 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。.NET と Java 用の Mobile Link サーバ API を使用して、アプリケーション・サーバ、Web サーバ、テキスト・ファイル、その他のデータベース製品など、任意のデータ・ソースと同期できます。

SQL Remote システムでは、中央データベースは SQL Anywhere である必要があります。

### リモート・データベース・システム

また、SQL Anywhere の同期テクノロジーでは、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 では、SQL Anywhere データベースと Ultra Light データベースをリモート・データベースとして使用できます。

SQL Remote では SQL Anywhere リモート・データベースを使用できます。

### 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、つまり、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に適しています。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。

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、同期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、同期は 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、それ以上頻繁には同期はできません。

Mobile Link と SQL Remote はどちらも、数時間または数日に 1 回程度の頻度で同期が発生する状況を基本的に想定して設計されていますが、数秒に 1 度の頻度で同期するのに使用することもできます。

### リモート・サイトの数

Mobile Link と SQL Remote は、どちらも多数のリモート・ユーザがいる環境でも問題なく動作します。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。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。

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

デフォルトでは、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。トランザクションの順序を維持したり、個別にアップロードするように選択することもできます。どちらの場合も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。コミットされていないデータは絶対に同期されないで、データの整合性は保たれます。

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。

### 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同期の長所です。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的が達成されます。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。

## モバイル・エンタープライズ・メッセージング : QAnywhere

QAnywhere では、エンタープライズ・メッセージングをモバイル・アプリケーションに拡張できます。エンタープライズ・メッセージングは、ビジネス・アプリケーション間でデータを交換するのによく使われる、有効な方法です。QAnywhere は、使用されているエンタープライズ・メッセージング・システムと組み合わせさせて、モバイル・デバイス間やモバイル・デバイスとエンタープライズ間にメッセージを提供します。QAnywhere は、異種モバイル環境で情報の接続と統合を行う包括的な蓄積転送メッセージング・ソリューションです。

QAnywhere は、リモート・アプリケーションとモバイル・アプリケーションに、セキュアで保証されたメッセージ配信機能を提供します。QAnywhere によってネットワークの速度と信頼性の低下という課題が自動的に対処されるので、接続性、通信、セキュリティに関わる問題の代わりに、アプリケーションの機能に集中的に取り組むことができます。QAnywhere の蓄積転送テクノロジーによって、ネットワークに接続されていない場合でも、アプリケーションは常に使用可能になります。

QAnywhere は、実証済みの Mobile Link 同期テクノロジーをベースにしているので、占有容量が小さくてすみ、設定と管理の負荷が低くなっています。また、このデータ同期およびメッセージングの共通のインフラストラクチャによって、管理要件を大幅に削減し、配備を簡略化できます。

QAnywhere には、次のような特性があります。

- モバイル・メッセージング・アプリケーションを構築するための強力な柔軟なプログラミング・モデルを使用した、包括的なメッセージング・インタフェース
- バックエンドの JMS ベース・エンタープライズ・システムに接続するコネクタ
- 圧縮およびトランザクション機能を持つ、信頼性が高く効率的なメッセージ配信
- 128 ビットの暗号化を使用したセキュアなメッセージの保管と転送
- ネットワークに依存しない通信
- 配信待機中のメッセージの Push 通知
- グラフィカルな管理ツール

### QAnywhere の用途

QAnywhere は、次の用途に使用されます。

- **バックエンドのエンタープライズ・アプリケーション・サーバとメッセージング・システムをモバイル・アプリケーションに拡張する** QAnywhere を使用して、Java Messaging Service (JMS) をサポートするバックエンド・システムと簡単に統合できるモバイル・アプリケーションを開発できます。
- **モバイル・エンタープライズ・メッセージング機能を既存のデータ同期システムに追加する** QAnywhere は、Mobile Link 同期テクノロジーをベースにしているので、両方の製品を 1 つのシステムに簡単に統合できます。また、共通のインフラストラクチャによって、管理要件を大幅に削減し、配備を簡略化できます。



QAnywhere と Mobile Link は、複数の方法で結合して、データの移動や変更を制御できます。たとえば、リモート・アプリケーションに対してデータを同期し、そのデータを使用して注文を作成し、メッセージを中間層ビジネス・ロジック・アプリケーションに送信して処理させることができます。

- **ネットワークに依存しない通信を提供する** QAnywhere のメッセージはネットワーク・プロトコルに依存しないので、受信側アプリケーションは、通信に使用しているネットワーク・プロトコルが送信側と同じでなくても、QAnywhere メッセージを受信できます。
- **随時接続環境での通信の実現する** メッセージングは蓄積転送機能を備えています。つまり、送信先アプリケーションにネットワークを介して接続できない場合でも、メッセージを作成しておくと、後でネットワークが使用可能になったときにそのメッセージが配信されます。
- **ルールベースの、条件に応じたメッセージ転送を使用する** QAnywhere では、メッセージの転送時期や配信時期を決定する規則を指定できます。たとえば、メッセージのプロパティやネットワークの転送コストなどを規則に追加できます。
- **モバイル Web サービスを作成する** モバイル Web サービスは、QAnywhere テクノロジーを使用して、Web サービスをモバイル環境に拡張します。[「モバイル Web サービス」 50 ページ](#) を参照してください。

[QAnywhere](#) を参照してください。

## モバイル Web サービス

### Web サービス

Web サービスは、異なるプラットフォームや言語で実行されるアプリケーションによるデータの対話や交換を実現します。Web サービス・システムでは、各アプリケーションが 1 つのインタフェースを使用して、アプリケーションの情報を Web サーバに変換します。たとえば、SQL Anywhere は独自の組み込み Web サービス・サーバまたは外部の Web サーバ経由で Web サービスの要求を送受信できます。この機能によって、他のアプリケーションが SQL Anywhere データベース内部に保存されている情報にアクセスできます。また、Web サービスは、サービス指向アーキテクチャ (SOA) で使用されます。

### モバイル Web サービス

SQL Anywhere のモバイル Web サービスによって、使用している Web サービスをモバイル環境に拡張できます。モバイル Web サービスは、Web サービスの機能と利点と、SQL Anywhere が持つ先進的なモバイル・テクノロジーを組み合わせ提供します。モバイル Web サービスを使用すると、モバイル・アプリケーションがオフラインであっても、Web サービス要求を作成し、後で転送できるように要求をキューイングできます。モバイル Web サービスでは、要求と応答を確実に配信するために、QAnywhere メッセージング・テクノロジーを使用しています。これにより、開発者は、接続された環境で作業している場合と同様に、Web サービスの開発やアクセスに集中できます。QAnywhere を使用すると、モバイル環境における要求と応答の転送、認証、直列化が簡単になります。

また、モバイル Web サービスには、次の特性があります。



- セキュアな通信を実現するための、HTTP と HTTPS の両方をサポートする Web サービス・コネクタ
- 開発を簡略化するプロキシ・クラスを生成する機能

#### モバイル Web サービスを使用する場合

モバイル Web サービスは、次の場合に使用されます。

- Web サービスにモバイル・アプリケーションからアクセスするとき
- 異なるシステム間のネットワーク接続が常時利用できないとき

「[モバイル Web サービス](#)」 [『QAnywhere』](#) を参照してください。

---

# サンプル・データベース

この項では、SQL Anywhere 11 のサンプル・データベースのスキーマについて説明します。サンプル・データベースを実際に使ってみて、SQL Anywhere 11 への理解を深めてください。

---

SQL Anywhere サンプル・データベース .....	55
CustDB サンプル・データベース・アプリケーション .....	61



---

# SQL Anywhere サンプル・データベース

## 目次

サンプル・データベースについて .....	56
サンプル・データベースの再作成 .....	58

---

## サンプル・データベースについて

このマニュアルでは、一貫性を保ち、わかりやすくするために、多くの例で SQL Anywhere のサンプル・データベース (*demo.db*) が使用されています。このファイルは、SQL Anywhere サンプル・ディレクトリにインストールされる *samples-dir\demo.db* です。

*samples-dir* のデフォルト・ロケーションについては、「[サンプル・ディレクトリ](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

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

ユーザ ID : **DBA**

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

### 警告

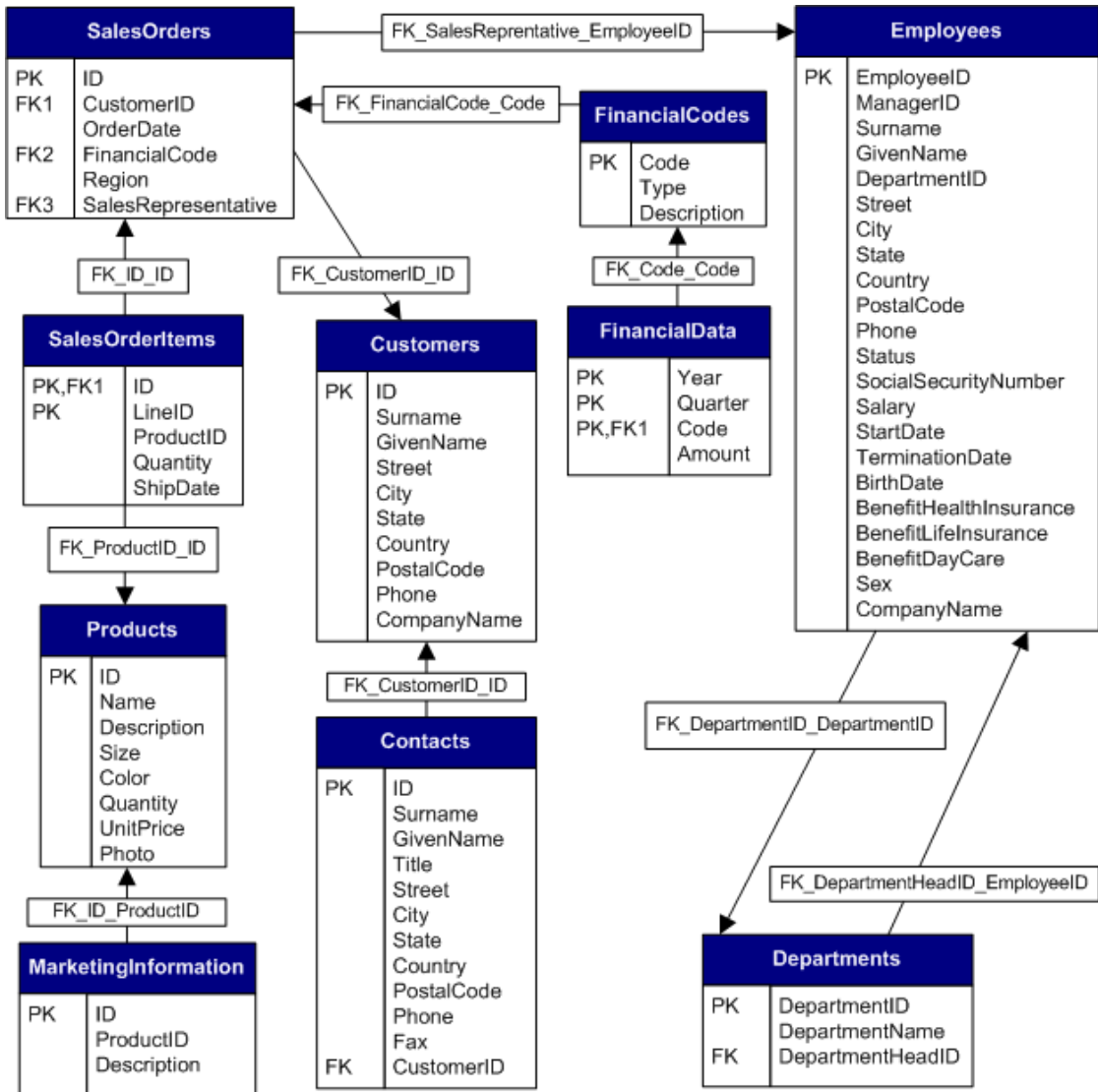
サンプル・データベースでは、常に同じユーザ ID とパスワードが使用されるので、DBA のユーザ ID とパスワードを変更して、データベースへのアクセスを制限することをおすすめします。  
「[パスワードの変更](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

サンプル・データベースでは、ODBC データ・ソース **SQL Anywhere 11 Demo** を使用します。

サンプル・データベースは、スポーツ衣料品を販売する中規模な企業の例です。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データ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。

*demo.db* に接続する手順については、「[チュートリアル：サンプル・データベースの使用](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。



## サンプル・データベースの再作成

機能をテストしたり、SQL Anywhere マニュアルのチュートリアルを完了したりすると、サンプル・データベースが変更され、以後のチュートリアルやテストが正常に行えない場合があります。このような場合には、サンプル・データベースを元の状態にリストアできます。また、サンプル・データベースの現在の状態を保持する必要がある場合は、別の名前を使って元の状態のサンプル・データベースを再作成できます。ここでは、両方の方法について説明します。

### ◆ サンプル・データベースを再作成するには、次の手順に従います。

1. コマンド・プロンプトで、サンプル・ディレクトリに移動します。

*samples-dir* のデフォルト・ロケーションについては、「[SQLANYSAMP11 環境変数](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

2. 次のコマンドを実行して、*demo.db* を削除します。

```
dberase demo.db
```

3. 選択の確定と、トランザクション・ログの削除を確認するプロンプトが表示されたら、**y** と入力します。
4. 次のコマンドを実行して、新しい空のサンプル・データベースを作成します。

```
dbinit demo.db
```

5. コマンド・プロンプトで、*install-dir\scripts* ディレクトリに移動します。

*install-dir* のデフォルト・ロケーションについては、「[SQLANY11 環境変数](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

6. 次のコマンドを実行して、オブジェクトとデータを含んだ新しいサンプル・データベースをロードします。

```
dbisql -c "DSN=SQL Anywhere 11 Demo" mkdemo.sql
```

### ◆ 別の名前を使ってサンプル・データベースを再作成するには、次の手順に従います。

1. コマンド・プロンプトで、サンプル・ディレクトリに移動します。

*samples-dir* のデフォルト・ロケーションについては、「[SQLANYSAMP11 環境変数](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

2. 次のコマンドを実行して、*mydemo.db* というデータベースを作成します。

```
dbinit mydemo.db
```

3. 次のコマンドを実行して、データベースを起動します。

```
dbeng11 mydemo.db
```

4. コマンド・プロンプトで、*install-dir\scripts* ディレクトリに移動します。

*install-dir* のデフォルト・ロケーションについては、「[SQLANY11 環境変数](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。



5. 次のコマンドを実行して、デモ・データベースの作成に使用するオブジェクトやデータを含んだ *mydemo.db* をロードします。

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

---

---

# CustDB サンプル・データベース・アプリケーション

## 目次

CustDB サンプル・データベースについて .....	62
------------------------------	----

---

## CustDB サンプル・データベースについて

CustDB サンプル・アプリケーションは、Ultra Light および Mobile Link アプリケーションの開発方法を学ぶのに便利なツールです。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。

CustDB サンプル・アプリケーションは、2つの部分から構成されています。

- **Ultra Light** Ultra Light では、CustDB は、Ultra Light でサポートされている任意のプラットフォームを使用して、Ultra Light でサポートされている任意のデバイス上に配備できます。Ultra Light の CustDB アプリケーションを作成するのに使用したソース・コードはすべて参照できます。また、このサンプルを実行することもできます。CustDB サンプル・アプリケーションは、Mobile Link 同期で処理できるように設定されています。

Ultra Light の CustDB サンプル・アプリケーションは、*samples-dir¥UltraLite¥CustDB¥*にあります。

*samples-dir* のデフォルト・ロケーションについては、「[サンプル・ディレクトリ](#)」『[SQL Anywhere サーバ - データベース管理](#)』を参照してください。

「[Ultra Light CustDB サンプル](#)」『[Ultra Light データベース管理とリファレンス](#)』を参照してください。

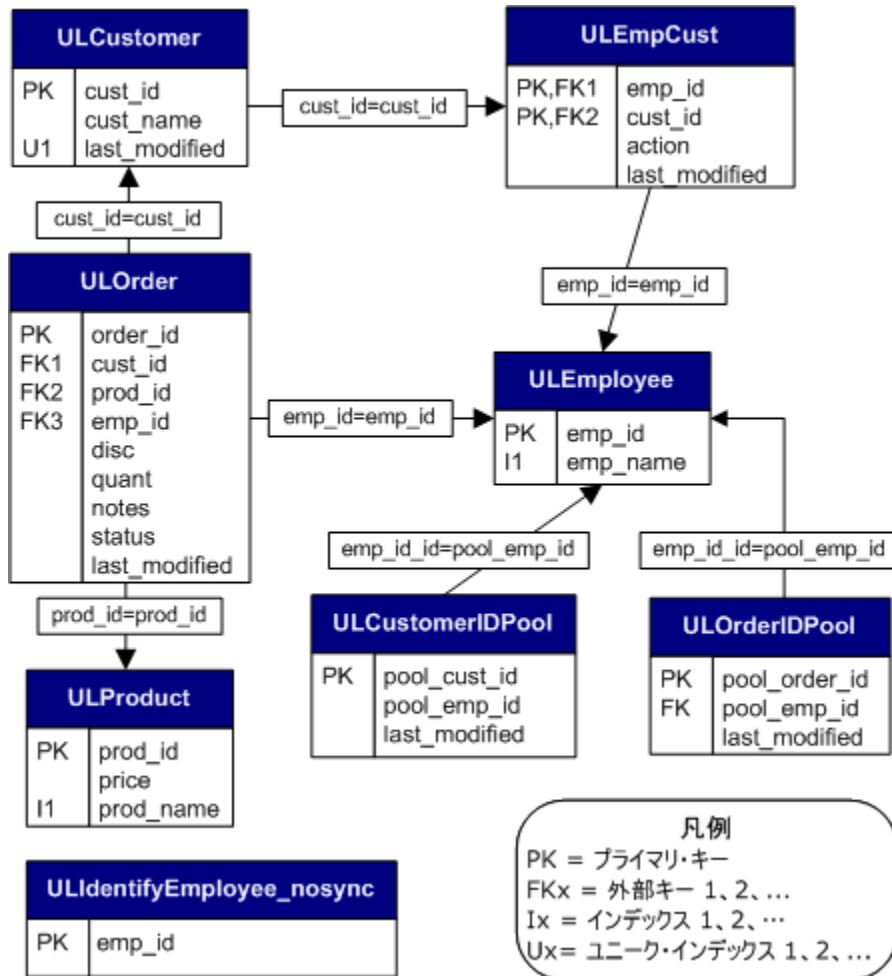
- **Mobile Link** Mobile Link の機能について詳しく知りたい場合は、サンプル同期論理が組み込まれている CustDB 統合データベースを使用できます。この統合データベースと Ultra Light の CustDB サンプル・リモート・データベースを使用して、サンプル同期システムを実行できます。

Mobile Link 統合 CustDB データベースは、サポートされているリレーショナル・データベース (SQL Anywhere、Adaptive Server Enterprise、Oracle、Microsoft SQL Server、DB2) に対してスクリプト・ファイルを実行することで作成されます。これらの設定ファイルは、*samples-dir¥MobiLink¥CustDB¥*にあります。

CustDB 統合データベースでは、ODBC データ・ソース **SQL Anywhere 11 CustDB** を使用します。

「[Mobile Link CustDB サンプルの解説](#)」『[Mobile Link - クイック・スタート](#)』を参照してください。

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。



---

# SQL Anywhere 11 のクイック・スタート

この項では、SQL Anywhere の主なテクノロジーの入門チュートリアルを提供します。データベースに精通し、このソフトウェアの実行をすぐに開始したいユーザが対象です。

---

SQL Anywhere 11 のクイック・スタート .....	67
----------------------------------	----





---

# SQL Anywhere 11 のクイック・スタート

## 目次

概要 .....	68
SQL Anywhere サーバのクイック・スタート .....	70
その他のアプリケーション .....	72

---

## 概要

このマニュアルのチュートリアル的大部分では、SQL Anywhere 11 のサンプル・データベース (*demo.db*) または CustDB サンプル・データベース・アプリケーション (*custdb.db*) が使用されています。

*demo.db* の詳細については、「[SQL Anywhere サンプル・データベース](#)」 55 ページを参照してください。

*custdb.db* の詳細については、次の項を参照してください。

- 「[Mobile Link CustDB サンプルの解説](#)」 『[Mobile Link - クイック・スタート](#)』
- 「[Ultra Light CustDB サンプル](#)」 『[Ultra Light データベース管理とリファレンス](#)』

### クイック・スタートの内容

次に、データベースとデータ交換テクノロジーの入門マニュアルとチュートリアルのリストを示します。

- **SQL Anywhere サーバ** 「[SQL Anywhere サーバのクイック・スタート](#)」 70 ページを参照してください。

この項では、SQL Anywhere サーバ・データベースの接続と管理に関連するいくつかのタスクを説明します。

- **Ultra Light** 「[Ultra Light CustDB サンプル](#)」 『[Ultra Light データベース管理とリファレンス](#)』を参照してください。

この章では、Ultra Light サンプル・データベース (*custdb.db*) を使用して次の操作を示します。

- Ultra Light リモート・データベースへのログインと移植
- クライアント・アプリケーションの使用
- Ultra Light データベースと統合データベースの同期
- Mobile Link 同期スクリプトの参照

- **Mobile Link** [Mobile Link - クイック・スタート](#)を参照してください。

このマニュアルでは、Mobile Link の概要を説明し、SQL Anywhere データベースを使用する同期システムの設定プロセスについて理解するための多くのチュートリアルを提供します。

- **QAnywhere** 「[QAnywhere のクイック・スタート](#)」 『[QAnywhere](#)』を参照してください。

この章には、QAnywhere メッセージングの設定と実行に必要なタスクの概要が記載されています。

- **SQL Remote** 「[SQL Remote の概要](#)」 『[SQL Remote](#)』を参照してください。

この章では、SQL Remote を紹介し、Sybase Central を使用して簡単な 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。

### その他のアプリケーション

SQL Anywhere には、物理データ・モデル、レポート、アプリケーションを作成するための次のアプリケーションも含まれています。

- **PowerDesigner Physical Data Model**   「[PowerDesigner Physical Data Model について](#)」 72 ページを参照してください。

PowerDesigner Physical Data Model を使用すると、データベースの物理データ・モデルを作成して、SQL Anywhere データベースに変換することができます。

- **InfoMaker**   「[InfoMaker について](#)」 72 ページを参照してください。

InfoMaker を使用すると、レポートをすばやく作成できます。

- **DataWindow .NET**   「[DataWindow .NET について](#)」 72 ページを参照してください。

DataWindow .NET を使用すると、アプリケーションを開発できます。

# SQL Anywhere サーバのクイック・スタート

## 概要

SQL Anywhere では、Sybase Central、Interactive SQL、コマンド・ライン・ユーティリティの 3 つの方法でシステムを管理できます。多くの場合、どのツールを使用するかは好みによります。データベースへの接続など、タスクの多くは、どの方法でも実行できます。

この項に示されているチュートリアルと概要部分では、3 つの方法すべてを使用して、SQL Anywhere データベースを作成および管理します。

Sybase Central と Interactive SQL の概要については、「[設計ツールと管理ツール](#)」 8 ページを参照してください。

## データベース・サーバの起動

- [「チュートリアル：サンプル・データベースの使用」](#) 『SQL Anywhere サーバ - データベース管理』

このチュートリアルでは、データベース・サーバの起動、データベース・サーバ・メッセージ・ウィンドウの表示、データベース・サーバの停止について説明します。

- **その他の情報** データベースまたはデータベース・サーバを起動するときに使用する最も一般的なオプションのリストについては、「[一般的なオプション](#)」 『SQL Anywhere サーバ - データベース管理』を参照してください。

サーバ・オプションの完全なリストは、「[SQL Anywhere データベース・サーバ](#)」 『SQL Anywhere サーバ - データベース管理』を参照してください。

または、次のコマンドを実行します。

```
dbeng11 -?
```

- **詳細情報** データベース・サーバの起動と停止の詳細については、「[データベース・サーバの実行](#)」 『SQL Anywhere サーバ - データベース管理』を参照してください。

SQL Anywhere でのデータベースとサーバの命名方法の詳細については、「[サーバとデータベースの命名](#)」 『SQL Anywhere サーバ - データベース管理』を参照してください。

## データベースへの接続

- [「SQL Anywhere データベース接続の例」](#) 『SQL Anywhere サーバ - データベース管理』

この項では、ODBC データ・ソースの作成方法と、そのデータ・ソースを Sybase Central で使用してデータベースに接続する方法について説明します。

- **詳細情報** SQL Anywhere サンプル・データベースへの接続、組み込みデータベースへの接続、ネットワーク経由での接続の例の詳細については、「[SQL Anywhere データベース接続](#)」 『SQL Anywhere サーバ - データベース管理』を参照してください。

ODBC データ ソースの詳細については、「[ODBC データ・ソースの作成](#)」 『SQL Anywhere サーバ - データベース管理』を参照してください。

## データベースの管理

### ● 「Sybase Central の使用」 『SQL Anywhere サーバ - データベース管理』

この項では、Sybase Central の概要と、Sybase Central を使用して次の操作を実行する方法を説明します。

- データベースへの接続
- 接続プロファイルの作成
- データベースの検索
- ER (実体関連) 図の表示
- データベースの正常性と統計情報のモニタリング

## データベースの作成

### ● 「チュートリアル : SQL Anywhere データベースの作成」 『SQL Anywhere サーバ - SQL の使用法』

このチュートリアルでは、Sybase Central を使用して次の操作を実行する方法を説明します。

- データベース・ファイルの作成
- データベースへの接続
- テーブルの作成
- テーブル間の関係の作成

## SQL 文の使用

### ● 「Interactive SQL の使用」 『SQL Anywhere サーバ - データベース管理』

この項では、Interactive SQL の概要と、次の手順を説明します。

- 複数の文の実行
- Interactive SQL コマンドのキャンセル
- テーブル、カラム、プロシージャの検索
- SQL 文の印刷
- コマンドの再呼び出しとログ
- スクリプト・ファイルの作成と実行

## その他のアプリケーション

次の項からは、PowerDesigner Physical Data Model、InfoMaker、および DataWindow .NET で利用可能なマニュアルやリソースについて説明します。

### PowerDesigner Physical Data Model について

SQL Anywhere 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 Physical Data Model が含まれています。このモジュールは、グラフィカルに表現されたデータベース・スキーマを使用してデータベースの生成や修正を行うことができます。これにより、テーブル、カラム、インデックス、キー、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 Physical Data Model には、ビデオによるチュートリアルを含む包括的なマニュアルが付属しています。PowerDesigner の詳細については、[www.sybase.com/products/modelingmetadata/powerdesigner](http://www.sybase.com/products/modelingmetadata/powerdesigner) を参照してください。

SQL Anywhere データベース設計の詳細については、「SQL Anywhere でのデータベースの作成」『SQL Anywhere サーバ - SQL の使用法』を参照してください。

### InfoMaker について

InfoMaker は、レポート作成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 には包括的なマニュアルが付属しています。InfoMaker の詳細については、[www.sybase.com/products/development/infomaker](http://www.sybase.com/products/development/infomaker) を参照してください。

### DataWindow .NET について

DataWindow .NET を使用すると、リッチ・クライアント、スマート・クライアント、モバイル、ワイヤレスのどのアプリケーションであっても、データ駆動型アプリケーションの作成が簡単になり、促進されます。また、生産性の高い 4GL 開発環境が含まれています。

DataWindow .NET には包括的なマニュアルが付属しています。DataWindow .NET の詳細については、[www.sybase.com/products/development/datawindownet](http://www.sybase.com/products/development/datawindownet) を参照してください。

# 用語解説

---

用語解説 .....	75
------------	----





---

# 用語解説

---

## 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。バージョン 10.0.0 で、Adaptive Server Anywhere は SQL Anywhere サーバに、SQL Anywhere Studio は SQL Anywhere にそれぞれ名前が変更されました。

参照：「[SQL Anywhere](#)」 80 ページ。

## Carrier

Mobile Link システム・テーブルまたは Notifier プロパティ・ファイルに保存される Mobile Link オブジェクトで、システム起動同期で使用される通信業者に関する情報が含まれます。

参照：「[サーバ起動同期](#)」 85 ページ。

## 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 つのデータベースは 13 個までのファイルに保管されます (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](#)」 89 ページ。

## DBA 権限

ユーザに、データベース内の管理作業を許可す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 \(DBA\)](#)」 89 ページ。

## EBF

Express Bug Fix の略です。Express Bug Fix は、1 つ以上のバグ・フィックスが含まれる、ソフトウェアのサブセットです。これらのバグ・フィックスは、更新のリリース・ノートにリストされます。バグ・フィックス更新を適用できるのは、同じバージョン番号を持つインストール済みのソフトウェアに対してだけです。このソフトウェアについては、ある程度のテストが行われているとはいえ、完全なテストが行われたわけではありません。自分自身でソフトウェアの妥当性を確かめるまでは、アプリケーションとともにこれらのファイルを配布しないでください。

## 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。SQL 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](#)」 [97 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。

## iAnywhere JDBC ドライバ

iAnywhere JDBC ドライバでは、pure Java である jConnect JDBC ドライバに比べて何らかの有利なパフォーマンスや機能を備えた JDBC ドライバが提供されます。ただし、このドライバは pure Java ソリューションではありません。iAnywhere JDBC ドライバは一般に推奨されるドライバです。

参照：

- 「[JDBC](#)」 [77 ページ](#)
- 「[jConnect](#)」 [77 ページ](#)

## 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、SQL 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返す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。iAnywhere JDBC ドライバは一般に推奨されるドライバです。

参照：

- [「JDBC」 77 ページ](#)
- [「iAnywhere JDBC ドライバ」 76 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。推奨 JDBC ドライバは、iAnywhere JDBC ドライバです。

参照：

- [「jConnect」 77 ページ](#)
- [「iAnywhere JDBC ドライバ」 76 ページ](#)

## Listener

Mobile Link サーバ起動同期に使用される、dblsn という名前のプログラムです。Listener はリモート・デバイスにインストールされ、Push 通知を受け取ったときにデバイス上でアクションが開始されるように設定されます。

参照：[「サーバ起動同期」 85 ページ](#)。

## LTM

LTM (Log Transfer Manager) は、Replication Agent と呼ばれます。Replication Server と併用することで、LTM はデータベース・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 Sybase Replication Server に送信します。

参照：[「Replication Server」 80 ページ](#)。

## Mobile Link

Ultra Light と SQL Anywhere のリモート・データベースを統合データベースと同期させるために設計された、セッションベース同期テクノロジーです。

参照：

- 「統合データベース」 104 ページ
- 「同期」 104 ページ
- 「Ultra Light」 81 ページ

## 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。SQL 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 サーバ

Mobile Link 同期を実行する、mlsrv11 という名前のコンピュータ・プログラムです。

## Mobile Link システム・テーブル

Mobile Link の同期に必要なシステム・テーブルです。Mobile Link 設定スクリプトによって、Mobile Link 統合データベースにインストールされます。

## Mobile Link モニタ

Mobile Link の同期をモニタするためのグラフィカル・ツールです。

## Mobile Link ユーザ

Mobile Link ユーザは、Mobile Link サーバに接続するのに使用されます。Mobile Link ユーザをリモート・データベースに作成し、統合データベースに登録します。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。

## Notifier

Mobile Link サーバ起動同期に使用されるプログラムです。Notifier は Mobile Link サーバに統合されており、統合データベースに Push 要求がないか確認し、Push 通知を送信します。

参照：

- 「サーバ起動同期」 85 ページ
- 「Listener」 77 ページ

## 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、SQL Anywhere がサポートするインタフェースの 1 つです。

---

## 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。

## PDB

Palm のデータベース・ファイルです。

## 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。SQL Anywhere には、PowerDesigner の Physical Data Model コンポーネントが付属します。

## PowerJ

Java アプリケーション開発に使用する Sybase 製品です。

## Push 通知

QAnywhere では、メッセージ転送を開始するよう QAnywhere クライアントに対して指示するために、サーバから QAnywhere クライアントに配信される特殊なメッセージです。Mobile Link サーバ起動同期では、Push 要求データや内部情報を含むデバイスに Notifier から配信される特殊なメッセージです。

参照：

- [「QAnywhere」 79 ページ](#)
- [「サーバ起動同期」 85 ページ](#)

## Push 要求

Mobile Link サーバ起動同期において、Push 通知をデバイスに送信する必要があるかどうかを判断するために Notifier が確認する、結果セット内の値のローです。

参照：[「サーバ起動同期」 85 ページ](#)。

## QAnywhere

アプリケーション間メッセージング (モバイル・デバイス間メッセージングやモバイル・デバイスとエンタープライズの間のメッセージングなど) を使用すると、モバイル・デバイスや無線デバイスで動作しているカスタム・プログラムと、集中管理されているサーバ・アプリケーションとの間で通信できます。

## QAnywhere Agent

QAnywhere では、クライアント・デバイス上で動作する独立のプロセスのことです。クライアント・メッセージ・ストアをモニタリングし、メッセージを転送するタイミングを決定します。

## REMOTE DBA 権限

SQL Remote では、Message Agent (dbremote) で必要なパーミッションのレベルを指します。Mobile Link では、SQL 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 (dbremote) または同期クライアントがこの権限のある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。Message Agent (dbremote) または同期クライアント (dbmlsync) から接続しない場合、このユーザ ID にはパーミッションは追加されません。

参照：「[DBA 権限](#)」 75 ページ。

## Replication Agent

参照：「[LTM](#)」 77 ページ。

## Replication Server

SQL 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・テクノロジーです。Replication Server は、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。

参照：「[LTM](#)」 77 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-2003 です。SQL は、公認されてはいませんが、Structured Query Language の略です。

## SQL Anywhere

SQLAnywhere 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。SQL Anywhere は、SQL Anywhere RDBMS、Ultra Light RDBMS、Mobile Link 同期ソフトウェア、その他のコンポーネントを含むパッケージの名前でもあります。

## 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データ・レプリケーション・テクノロジーです。統合データベースとリモート・データベースは、SQL Anywhere である必要があります。

---

## SQL ベースの同期

Mobile Link では、Mobile Link イベントを使用して、テーブル・データを Mobile Link でサポートされている統合データベースに同期する方法のことです。SQL ベースの同期では、SQL を直接使用したり、Java と .NET 用の Mobile Link サーバ API を使用して SQL を返すことができます。

## 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」 87 ページ](#)
- [「SQL」 80 ページ](#)
- [「データベース管理システム \(DBMS\)」 89 ページ](#)

## Sybase Central

SQL 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

小型デバイス、モバイル・デバイス、埋め込みデバイス用に最適化されたデータベースです。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。

## Ultra Light ランタイム

組み込みの Mobile Link 同期クライアントを含む、インプロセス・リレーショナル・データベース管理システムです。Ultra Light ランタイムは、Ultra Light の各プログラミング・インタフェースで使用するライブラリと、Ultra Light エンジンの両方に含まれます。

## Windows

Windows Vista、Windows XP、Windows 200x などの、Microsoft Windows オペレーティング・システムのファミリのことです。

## Windows CE

[「Windows Mobile」 81 ページ](#)を参照してください。

## Windows Mobile

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。

## アーティクル

Mobile Link または SQL Remote 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」 97 ページ](#)
- [「パブリケーション」 92 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

完全に処理されるかまったく処理されないことが保証される 1 つのトランザクションです。エラーによってアトミックなトランザクションの一部が処理されなかった場合は、データベースが一貫性のない状態になるのを防ぐために、トランザクションがロールバックされます。

## 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。

## イベント・モデル

Mobile Link では、同期を構成する、`begin_synchronization` や `download_cursor` などの一連のイベントのことです。イベントは、スクリプトがイベント用に作成されると呼び出されます。

## 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」 91 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。



---

## ウィンドウ

分析関数の実行対象となるローのグループです。ウィンドウには、ウィンドウ定義内のグループ化指定に従って分割されたデータの、1つ、複数、またはすべてのローが含まれます。ウィンドウは、入力現在のローについて計算を実行する必要があるローの数や範囲を含むように移動します。ウィンドウ構成の主な利点は、追加のクエリを実行しなくても、結果をグループ化して分析する機会が増えることです。

## エージェント ID

参照：[「クライアント・メッセージ・ストア ID」 84 ページ](#)。

## エンコード

文字コードとも呼ばれます。エンコードは、文字セットの各文字が情報の1つまたは複数のバイトにマッピングされる方法のことで、一般的に16進数で表現されます。UTF-8はエンコードの例です。

参照：

- [「文字セット」 105 ページ](#)
- [「コード・ページ」 85 ページ](#)
- [「照合」 102 ページ](#)

## 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」 81 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。SQL 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」 84 ページ](#)
- [「カーソル位置」 83 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」 83 ページ](#)
- [「カーソル結果セット」 84 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」 83 ページ](#)
- [「カーソル位置」 83 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」 80 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。

## クライアント・メッセージ・ストア

QAnywhere では、メッセージを保管するリモート・デバイスにある SQL Anywhere データベースのことです。

## クライアント・メッセージ・ストア ID

QAnywhere では、Mobile Link リモート ID のことです。これによって、クライアント・メッセージ・ストアがユニークに識別されます。

## 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」 90 ページ](#)
- [「ローカル・テンポラリ・テーブル」 98 ページ](#)

---

## ゲートウェイ

Mobile Link システム・テーブルまたは Notifier プロパティ・ファイルに保存される Mobile Link オブジェクトで、システム起動同期用のメッセージの送信方法に関する情報が含まれます。

参照：「[サーバ起動同期](#)」 85 ページ。

## コード・ページ

コード・ページは、文字セットの文字を数値表示 (通常 0 ～ 255 の整数) にマッピングするエンコードです。Windows Code Page 1252 などのコード・ページがあります。このマニュアルの目的上、コード・ページとエンコードは同じ意味で使用されます。

参照：

- 「[文字セット](#)」 105 ページ
- 「[エンコード](#)」 83 ページ
- 「[照合](#)」 102 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。

## サーバ・メッセージ・ストア

QAnywhere では、サーバ上のリレーショナル・データベースです。このデータベースは、メッセージを、クライアント・メッセージ・ストアまたは JMS システムに転送されるまで一時的に格納します。メッセージは、サーバ・メッセージ・ストアを介して、クライアント間で交換されます。

## サーバ管理要求

XML 形式の QAnywhere メッセージです。サーバ・メッセージ・ストアを管理したり、QAnywhere アプリケーションをモニタリングするために QAnywhere システム・キューに送信されます。

## サーバ起動同期

Mobile Link サーバから Mobile Link 同期を開始する方法です。

## 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

Mobile Link 同期では、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。

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。

参照：

- [「パブリケーション」 92 ページ](#)
- [「Mobile Link ユーザ」 78 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

SQL 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」 86 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、SQL Anywhere がジョイン条件を生成します。

参照：

- [「ジョイン」 86 ページ](#)
- [「生成されたジョイン条件」 103 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の制御します。

参照：[「イベント・モデル」 82 ページ](#)。

## スクリプト・バージョン

Mobile Link では、同期を作成するために同時に適用される、一連の同期スクリプトです。

## スクリプトベースのアップロード

Mobile Link では、ログ・ファイルを使用した方法の代わりとなる、アップロード処理のカスタマイズ方法です。

## ストアド・プロシージャ

ストアド・プロシージャは、データベースに保存され、データベース・サーバに対する一連の操作やクエリを実行するために使用される SQL 命令のグループです。

## スナップショット・アイソレーション

読み込み要求を発行するトランザクション用のデータのコミットされたバージョンを返す、独立性レベルの種類です。SQL Anywhere では、スナップショット、文のスナップショット、読み込み専用文のスナップショットの 3 つのスナップショットの独立性レベルがあります。スナップショット・アイソレーションが使用されている場合、読み込み処理は書き込み処理をブロックしません。

参照：[「独立性レベル」 105 ページ](#)。

## セキュア機能

データベース・サーバが起動されたときに、そのデータベース・サーバで実行されているデータベースでは使用できないように -sf オプションによって指定される機能です。

## 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。

## ダイレクト・ロー・ハンドリング

Mobile Link では、テーブル・データを Mobile Link でサポートされている統合データベース以外のソースに同期する方法のことです。アップロードとダウンロードの両方をダイレクト・ロー・ハンドリングで実装できます。

参照：

- 「[統合データベース](#)」 104 ページ
- 「[SQL ベースの同期](#)」 81 ページ

## 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。

## チェックサム

データベース・ページを使用して記録されたデータベース・ページのビット数の合計です。チェックサムを使用すると、データベース管理システムは、ページがディスクに書き込まれるときに数が一致しているかを確認することで、ページの整合性を検証できます。数が一致した場合は、ページが正常に書き込まれたとみなされます。

## 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・キューブ

同じ結果を違う方法でグループ化およびソートされた内容を各次元に反映した、多次元の結果セットです。データ・キューブは、セルフジョイン・クエリと相関サブクエリを必要とするデータの複雑な情報を提供します。データ・キューブは OLAP 機能の一部です。

## 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](#)」 99 ページ
- 「[プライマリ・キー](#)」 94 ページ
- 「[データベース管理システム \(DBMS\)](#)」 89 ページ
- 「[リレーショナル・データベース管理システム \(RDBMS\)](#)」 97 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。SQL 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](#)」 75 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 \(RDBMS\)](#)」 97 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 \(DBA\)](#)」 89 ページ
- 「[SYS](#)」 81 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ンネルです。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」 89 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みます。

参照：「ドメイン」 90 ページ。

## データ操作言語 (DML)

データベース内のデータの操作に使う SQL 文のサブセットです。DML 文は、データベース内のデータを検索、挿入、更新、削除します。

## データ定義言語 (DDL)

データベース内のデータの構造を定義するときに使う SQL 文のサブセットです。DDL 文は、テーブルやユーザなどのデータベース・オブジェクトを作成、変更、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。

## デバイス・トラッキング

Mobile Link サーバ起動同期において、デバイスを特定する Mobile Link のユーザ名を使用して、メッセージのアドレスを指定できる機能です。

参照：「サーバ起動同期」 85 ページ。

## 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」 98 ページ
- 「グローバル・テンポラリ・テーブル」 84 ページ

## ドメイン

適切な位置に精度や小数点以下の桁数を含み、さらにオプションとしてデフォルト値や CHECK 条件などを含んでいる、組み込みデータ型のエイリアスです。ドメインには、通貨データ型のように SQL Anywhere が事前に定義したものもあります。ユーザ定義データ型とも呼ばれます。

参照：「データ型」 90 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。SQL 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はありません。トランザクションは、データに加えた変更を永久的なものにする 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。

### 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」 91 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」 98 ページ](#)
- [「文レベルのトリガ」 105 ページ](#)
- [「整合性」 102 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」 92 ページ](#)。

### ネットワーク・プロトコル

TCP/IP や HTTP などの通信の種類です。

## 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。

## ハッシュ

ハッシュは、インデックスのエントリをキーに変換する、インデックスの最適化のことです。インデックスのハッシュの目的は、必要なだけの実際のロー・データをロー ID に含めることで、インデックスされた値を特定するためのローの検索、ロード、アンパックという負荷の高い処理を避けることです。

## 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、データベース・サーバ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

Mobile Link または SQL Remote では、同期されるデータを識別するデータベース・オブジェクトのことです。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ティクルから構成されています。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」 97 ページ](#)
- [「アティクル」 82 ページ](#)
- [「パブリケーションの更新」 92 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」 97 ページ](#)
- [「パブリケーション」 92 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](#)」 97 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](#)」 102 ページ
- 「[ユーザ定義データ型](#)」 96 ページ

## ヒストグラム

ヒストグラムは、カラム統計のもっとも重要なコンポーネントであり、データ分散を表します。SQL Anywhere は、ヒストグラムを維持して、カラムの値の分散に関する統計情報をオプティマイザに提供します。

## ビット配列

ビット配列は、一連のビットを効率的に保管するのに使用される配列データ構造の種類です。ビット配列は文字列に似てますが、使用される要素は文字ではなく 0 (ゼロ) と 1 になります。ビット配列は、一般的にブール値の文字列を保持するのに使用されます。

## 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。

## ファイル定義データベース

Mobile Link では、ダウンロード・ファイルの作成に使用される SQL Anywhere データベースのことです。

参照：「[ファイルベースのダウンロード](#)」 93 ページ。

## フェールオーバー

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。フェールオーバーは自動的に発生します。

## 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。

参照：「[外部キー](#)」 [99 ページ](#)。

## 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](#)」 [102 ページ](#)
- 「[検査制約](#)」 [101 ページ](#)
- 「[外部キー制約](#)」 [100 ページ](#)
- 「[一意性制約](#)」 [99 ページ](#)
- 「[整合性](#)」 [102 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 SQL Anywhere)で表示されます。

参照：「[Sybase Central](#)」 [81 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ており、システム障害やメディア障害が発生した場合の保護として機能します。

参照：「[インクリメンタル・バックアップ](#)」 [82 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](#)」 [95 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」 90 ページ](#)
- [「ビュー」 93 ページ](#)

## ポーリング

Mobile Link サーバ起動同期において、Mobile Link Listener などのライト・ウェイト・ポーラが Notifier から Push 通知を要求する方法です。

参照：[「サーバ起動同期」 85 ページ](#)。

## ポリシー

QAnywhere では、メッセージ転送の発生時期を指定する方法のことです。

## マテリアライズド・ビュー

計算され、ディスクに保存されたビューのことです。マテリアライズド・ビューは、ビュー (クエリ指定を使用して定義される) とテーブル (ほとんどのテーブルの操作をそのテーブル上で実行できる) の両方の特性を持ちます。

参照：

- [「ベース・テーブル」 95 ページ](#)
- [「ビュー」 93 ページ](#)

## ミラー・ログ

参照：[「トランザクション・ログ・ミラー」 91 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、他のデータの特徴と内容について記述しています。

参照：[「スキーマ」 87 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。SQL Anywhere では、FILE、FTP、SMTP 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」 97 ページ](#)
- [「FILE」 76 ページ](#)

## メッセージ・ストア

QAnywhere では、メッセージを格納するクライアントおよびサーバ・デバイスのデータベースのことです。

参照：

- [「クライアント・メッセージ・ストア」 84 ページ](#)
- [「サーバ・メッセージ・ストア」 85 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ようになります。

参照：

- [「レプリケーション」 97 ページ](#)
- [「統合データベース」 104 ページ](#)

## メッセージ・ログ

データベース・サーバや Mobile Link サーバなどのアプリケーションからのメッセージを格納できるログです。この情報は、メッセージ・ウィンドウに表示されたり、ファイルに記録されたりすることもあります。メッセージ・ログには、情報メッセージ、エラー、警告、MESSAGE 文からのメッセージが含まれます。

## メンテナンス・リリース

メンテナンス・リリースは、同じメジャー・バージョン番号を持つ旧バージョンのインストール済みソフトウェアをアップグレードするための完全なソフトウェア・セットです (バージョン番号のフォーマットは、メジャー、マイナー、パッチ、ビルドです)。バグ・フィックスとその他の変更については、アップグレードのリリース・ノートにリストされます。

## ユーザ定義データ型

参照：[「ドメイン」 90 ページ](#)。

## ライト・ウェイト・ポーラ

Mobile Link サーバ起動同期において、Mobile Link サーバからの Push 通知をポーリングするデバイス・アプリケーションです。

参照：[「サーバ起動同期」 85 ページ](#)。

---

## リダイレクタ

クライアントと Mobile Link 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ー・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 では、Ultra Light クライアントの開発に使用される SQL Anywhere データベースです。開発中は、1 つの SQL 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 ID

SQL Anywhere と Ultra Light データベース内のユニークな識別子で、Mobile Link によって使用されます。リモート ID は NULL に初期設定されていますが、データベースの最初の同期時に GUID に設定されます。

## リモート・データベース

Mobile Link または SQL Remote 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：

- [「同期」 104 ページ](#)
- [「統合データベース」 104 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 \(DBMS\)」 89 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・テクノロジーを提供しています。

## 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・システムに必要なパススルー文、情報があります。

参照：

- [「レプリケーション」 97 ページ](#)
- [「パブリケーションの更新」 92 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」 97 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」 91 ページ](#)
- [「文レベルのトリガ」 105 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」 90 ページ](#)
- [「グローバル・テンポラリ・テーブル」 84 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フレーズを指します。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)" 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」 91 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。ロール名は独自に作成することもできます。

参照：[「外部キー」 99 ページ](#)。



---

## ログ・ファイル

SQL 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」 91 ページ](#)
- [「トランザクション・ログ・ミラー」 91 ページ](#)
- [「フル・バックアップ」 94 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。SQL 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」 105 ページ](#)
- [「同時性 \(同時実行性\)」 105 ページ](#)
- [「整合性」 102 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。

## 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」 100 ページ](#)
- [「プライマリ・キー制約」 94 ページ](#)
- [「制約」 102 ページ](#)

## 解析ツリー

クエリを代数で表現したものです。

## 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」 94 ページ](#)
- [「外部テーブル」 100 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」 102 ページ](#)
- [「検査制約」 101 ページ](#)
- [「プライマリ・キー制約」 94 ページ](#)
- [「一意性制約」 99 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。SQL 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」 86 ページ](#)
- [「内部ジョイン」 105 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」 99 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、SQL 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。

---

## 競合解決

Mobile Link では、競合解決は 2 人のユーザが別々のリモート・データベースの同じローを変更した場合にどう処理するかを指定するロジックのことです。

## 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」 102 ページ
- 「外部キー制約」 100 ページ
- 「プライマリ・キー制約」 94 ページ
- 「一意性制約」 99 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。

## 作成者 ID

Ultra Light の Palm OS アプリケーションでは、アプリケーションが作成されたときに割り当てられる ID のことです。

## 参照元オブジェクト

テーブルなどのデータベースの別のオブジェクトをオブジェクト定義が直接参照する、ビューなどのオブジェクトです。

参照：「外部キー」 99 ページ。

## 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」 94 ページ
- 「外部キー」 99 ページ

## 参照先オブジェクト

ビューなどの別のオブジェクトの定義で直接参照される、テーブルなどのオブジェクトです。

参照：「プライマリ・キー」 94 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。SQL 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。

参照：

- [「文字セット」 105 ページ](#)
- [「コード・ページ」 85 ページ](#)
- [「エンコード」 83 ページ](#)

## 世代番号

Mobile Link では、リモート・データベースがデータをアップロードしてからダウンロード・ファイルを適用するようにするためのメカニズムのことです。

参照：[「ファイルベースのダウンロード」 93 ページ](#)。

## 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」 101 ページ](#)
- [「外部キー制約」 100 ページ](#)
- [「プライマリ・キー制約」 94 ページ](#)
- [「一意性制約」 99 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](#)」 101 ページ。

## 正規化

データベース・スキーマ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。

## 正規表現

正規表現は、文字列内で検索するパターンを定義する、一連の文字、ワイルドカード、演算子です。

## 生成されたジョイン条件

自動的に生成される、ジョインの結果に対する制限です。キーとナチュラルの2種類があります。キー・ジョインは、KEY JOIN を指定したとき、またはキーワード JOIN を指定したが、CROSS、NATURAL、または ON を使用しなかった場合に生成されます。キー・ジョインの場合、生成されたジョイン条件はテーブル間の外部キー関係に基づいています。ナチュラル・ジョインは NATURAL JOIN を指定したときに生成され、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。

参照：

- 「[ジョイン](#)」 86 ページ
- 「[ジョイン条件](#)」 87 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。

## 接続起動同期

Mobile Link のサーバ起動同期の1つの形式で、接続が変更されたときに同期が開始されます。

参照：「[サーバ起動同期](#)」 85 ページ。

## 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 句で定義した代替名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。

参照：[「レプリケーション」 97 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 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。

## 転送ルール

QAnywhere では、メッセージの転送を発生させる時期、転送するメッセージ、メッセージを削除する時期を決定する論理のことです。

## 統合データベース

分散データベース環境で、データのマスタ・コピーを格納するデータベースです。競合や不一致が発生した場合、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。

参照：

- [「同期」 104 ページ](#)
- [「レプリケーション」 97 ページ](#)

## 統合化ログイン

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。

## 動的 SQL

実行される前に作成したプログラムによって生成される SQL です。Ultra Light の動的 SQL は、占有容量の小さいデバイス用に設計された変形型です。

## 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」 77 ページ](#)
- [「SQL Remote」 80 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、SQL 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」 91 ページ](#)
- [「独立性レベル」 105 ページ](#)

## 独立性レベル

あるトランザクションの操作が、同時に処理されている別のトランザクションの操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。SQL Anywhere では、スナップショット、文のスナップショット、読み込み専用文のスナップショットの 3 つのスナップショットの独立性レベルがあります。

参照：[「スナップショット・アイソレーション」 87 ページ](#)。

## 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」 86 ページ](#)
- [「外部ジョイン」 100 ページ](#)

## 物理インデックス

インデックスがディスクに保存されるときの実際のインデックス構造です。

## 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」 91 ページ](#)
- [「ロー・レベルのトリガ」 98 ページ](#)

## 文字セット

文字セットは記号、文字、数字、スペースなどから成ります。"ISO-8859-1" は文字セットの例です。Latin1 と呼ばれます。

参照：

- [「コード・ページ」 85 ページ](#)
- [「エンコード」 83 ページ](#)
- [「照合」 102 ページ](#)

## 文字列リテラル

文字列リテラルとは、一重引用符 (') で囲まれ、シーケンスで並べられた文字のことです。

## 論理インデックス

物理インデックスへの参照 (ポインタ) です。ディスクに保存される論理インデックス用のインデックス構造はありません。



---

# 索引

## 記号

- .NET 同期論理
  - サポートされるプラットフォーム, 15
- 2 方向の同期
  - 同期テクノロジーの概要, 44
- 508 条
  - アクセシビリティ有効化モジュール, 16
- 32 ビット
  - サポートされるバージョン, 15

## A

- ActiveSync
  - サポートされるプラットフォーム, 15
- AIX
  - サポートされるバージョン, 15
  - サポートされるプラットフォーム, 15
- API
  - 説明, 35
- Apple (参照 Mac OS X)
- ARMV4T
  - サポートされるプラットフォーム, 15
- ARM V4T モード
  - サポートされるプラットフォーム, 15
- ARM チップ
  - サポートされるプラットフォーム, 15
- ARM プロセッサ
  - サポートされるプラットフォーム, 15

## B

- BlackBerry
  - SQL Anywhere データベースはサポート対象外, 29
  - Ultra Light データベースのサポート, 29

## C

- Caldera
  - サポートされるバージョン, 15
- Carrier
  - 用語定義, 75
- CE (参照 Windows Mobile)
- Certicom
  - 暗号化ソフトウェアの注文, 11
- Command Sequence 通信プロトコル

- 図, 35
- 説明, 38
- components\_platforms.html
  - ロケーション, 15
- CPU 数
  - ライセンス, 10
- cusdb.db
  - 説明, 62
- CustDB
  - 説明, 62
- CustDB アプリケーション
  - 説明, 62

## D

- DataWindow .NET
  - 説明, 72
- Data Window .NET (参照 DataWindow .NET)
- DBA 権限
  - 用語定義, 75
- DBMS
  - 用語定義, 89
- dbremote
  - サポートされるプラットフォーム, 15
- DB 領域
  - 用語定義, 75
- DCX
  - 説明, vi
- DDL
  - 用語定義, 90
- demo.db ファイル
  - 再作成, 58
  - 説明, 56
- DML
  - 用語定義, 90
- DocCommentXchange (DCX)
  - 説明, vi
- DOS
  - サポートされるプラットフォーム, 15

## E

- EBF
  - 用語定義, 75
- ECC
  - 暗号化ソフトウェアの注文, 11
- Embedded SQL
  - 用語定義, 76
- ETL

サポートされている機能, 34  
説明, 34

## F

### FILE

用語定義, 76

### FILE メッセージ・タイプ

用語定義, 76

### FIPS

暗号化ソフトウェアの注文, 11

### FTP メッセージ・タイプ

SQL Remote がサポートする Windows プラット  
フォーム, 15

## G

### grant オプション

用語定義, 76

## H

### Hewlett Packard HP-UX

(参照 HP-UX)

### HP-UX

サポートされるバージョン, 15

サポートされるプラットフォーム, 15

## I

### iAnywhere JDBC ドライバ

用語定義, 76

### iAnywhere デベロッパー・コミュニティ

ニュースグループ, xii

### IBM AIX

(参照 AIX)

### InfoMaker

説明, 72

用語定義, 76

### install-dir

マニュアルの使用法, ix

### Interactive SQL

サポートされるプラットフォーム, 15

用語定義, 76

## J

### JAR ファイル

用語定義, 76

### Java ME

サポートされるバージョン, 15

### Java クラス

用語定義, 77

### Java 同期論理

サポートされるプラットフォーム, 15

### jConnect

用語定義, 77

### JDBC

用語定義, 77

## L

### Linux

サポートされるバージョン, 15

サポートされるプラットフォーム, 15

### Listener

用語定義, 77

### LTM

別途ライセンスが必要なコンポーネント, 12

用語定義, 77

## M

### Macintosh

(参照 Mac OS X)

サポートされるバージョン, 15

サポートされるプラットフォーム, 15

### Mac OS X

Ultra Light はサポート対象外, 29

サポートされるバージョン, 15

サポートされるプラットフォーム, 15

### Mandrake

サポートされるバージョン, 15

### Message Agent

サポートされるプラットフォーム, 15

### MIPS チップ

サポートされるプラットフォーム, 15

### Mobile Link

SQL Remote と Mobile Link の比較, 39

サポートされるプラットフォーム, 15

用語定義, 77

### Mobile Link がサポートしているプラットフォーム

説明, 15

### Mobile Link クライアント

用語定義, 78

### Mobile Link サーバ

用語定義, 78

### Mobile Link システム・テーブル

用語定義, 78

### Mobile Link 同期

---

機能の比較, 47  
サポートされるプラットフォーム, 15  
Mobile Link 統合データベース  
サポートされる RDBMS, 15  
Mobile Link モニタ  
用語定義, 78  
Mobile Link ユーザ  
用語定義, 78

## N

Notifier  
用語定義, 78  
n 層コンピューティング  
概要, 32

## O

ODBC  
用語定義, 78  
ODBC アドミニストレータ  
用語定義, 79  
ODBC データ・ソース  
SQL Anywhere 11 CustDB, 62  
SQL Anywhere 11 Demo, 56  
用語定義, 79  
OLE DB ドライバ  
サポートされるプロセッサ, 15

## P

Palm Computing Platform  
Ultra Light 開発環境, 15  
Palm OS  
サポートされるバージョン, 15  
PDB  
用語定義, 79  
PDF  
マニュアル, vi  
Pocket PC  
Windows Mobile でサポートされるプラットフォーム, 15  
PowerDesigner  
説明, 72  
用語定義, 79  
PowerJ  
用語定義, 79  
Push 通知  
用語定義, 79  
Push 要求

用語定義, 79

## Q

QAnywhere  
サポートされるプラットフォーム, 15  
用語定義, 79  
利点, 49  
QAnywhere Agent  
用語定義, 80  
QAnywhere がサポートしているプラットフォーム  
説明, 15

## R

RDBMS  
用語定義, 97  
Red Hat  
サポートされるバージョン, 15  
REMOTE DBA 権限  
用語定義, 80  
Replication Agent  
サポートされるプラットフォーム, 15  
別途ライセンスが必要なコンポーネント, 12  
用語定義, 80  
Replication Server  
用語定義, 80

## S

samples-dir  
マニュアルの使用方法, ix  
SMTP メッセージ・タイプ  
サポートされるプラットフォーム, 15  
SOA  
サービス指向アーキテクチャの説明, 50  
Solaris  
サポートされるバージョン, 15  
サポートされるプラットフォーム, 15  
SQL  
用語定義, 80  
SQL Anywhere  
Ultra Light との比較, 29  
クイック・スタート, 70  
コンポーネント, 6  
サポートされるプラットフォーム, 15  
使用目的, 4  
説明, 4  
特長, 14  
内部, 25

マニュアル, vi  
用語定義, 80  
SQL Anywhere がサポートしているプラットフォーム  
説明, 15  
SQL Anywhere サンプル・データベース  
再作成, 58  
説明, 56  
SQL Anywhere データベース  
Ultra Light データベースとの比較, 29  
SQL Anywhere データベースと Ultra Light データベースとの比較  
説明, 29  
SQL Anywhere の概要  
クイック・スタート, 68  
紹介, 4  
設計ツールと管理ツール, 8  
データベース・テクノロジー, 6  
同期テクノロジー, 6  
メッセージング・テクノロジー, 6  
SQL Remote  
Mobile Link と SQL Remote の比較, 39  
サポートされるプラットフォーム, 15  
用語定義, 80  
SQL Remote がサポートしているプラットフォーム  
説明, 15  
SQL 文  
用語定義, 81  
SQL ベースの同期  
用語定義, 81  
Sun Solaris  
(参照 Solaris)  
SuSE  
サポートされるバージョン, 15  
Sybase Central  
サポートされるプラットフォーム, 15  
用語定義, 81  
SYS  
用語定義, 81

**T**  
Tabular Data Stream 通信プロトコル  
図, 35  
説明, 38  
TDS 通信プロトコル  
(参照 Tabular Data Stream 通信プロトコル)

Thumb モード  
サポートされるモード, 15  
TurboLinux  
サポートされるバージョン, 15

## U

Ultra Light  
SQL Anywhere との比較, 29  
サポートされるプラットフォーム, 15  
用語定義, 81  
Ultra Light がサポートしているプラットフォーム  
説明, 15  
Ultra Light データベース  
SQL Anywhere データベースとの比較, 29  
Ultra Light ランタイム  
用語定義, 81  
UNIX  
サポートされるバージョン, 15  
サポートされるプラットフォーム, 15

## V

V4T モード  
ARM プロセッサ, 15  
VCS エージェント  
別途ライセンスが必要なコンポーネント, 12  
Veritas Cluster Server エージェント  
別途ライセンスが必要なコンポーネント, 12  
Vista  
サポートされるプラットフォーム, 15

## W

Windows  
サポートされるバージョン, 15  
サポートされるプラットフォーム, 15  
用語定義, 81  
Windows 2000  
サポートされるプラットフォーム, 15  
Windows Mobile  
OLE DB サポート, 15  
サポートされるバージョン, 15  
サポートされるプラットフォーム, 15  
サポートされるプロセッサ, 15  
用語定義, 81  
Windows Vista  
サポートされるプラットフォーム, 15  
Windows XP

---

サポートされるプラットフォーム, 15

## X

x86 チップ

サポートされるプラットフォーム, 15

XScale プロセッサ

サポートされるプラットフォーム, 15

## あ

アイコン

ヘルプでの使用, xi

アクセシビリティ

アクセシビリティ有効化モジュール, 16

アップロード

用語定義, 82

アトミック・トランザクション

用語定義, 82

暗号化

暗号化ソフトウェアの注文, 11

アンロード

用語定義, 82

アーティクル

用語定義, 82

## い

一意性制約

用語定義, 99

イベント・モデル

用語定義, 82

インクリメンタル・バックアップ

用語定義, 82

インストール

サポートされるプラットフォーム, 15

インデックス

用語定義, 82

イン・メモリ・モード

別途ライセンスが必要なコンポーネント, 12

## う

ウィンドウ (OLAP)

用語定義, 83

## え

エディション

詳細情報の参照先, 10

別途ライセンスが必要なコンポーネントのバンドル, 10

エンコード

用語定義, 83

エンタープライズ・メッセージング

QAnywhere の説明, 49

エンティティ

関係, 20

エージェント ID

用語定義, 83

## お

応答時間

データの同期の利点, 41

オブジェクト・ツリー

用語定義, 83

オペレーティング・システム

サポートされるプラットフォーム, 15

オンライン・マニュアル

PDF, vi

## か

解析ツリー

用語定義, 99

階層データ構造

階層データベース構成, 43

開発プラットフォーム

サポート・プラットフォーム, 15

外部キー

説明, 22

定義, 21

用語定義, 99

外部キー制約

用語定義, 100

外部ジョイン

用語定義, 100

外部テーブル

用語定義, 100

外部ログイン

用語定義, 100

カラム

説明, 20

環境変数

コマンド・シェル, x

コマンド・プロンプト, x

関係

エンティティ, 20

管理ツール

サポートされるプラットフォーム, 15

カーソル  
  用語定義, 83  
カーソル位置  
  用語定義, 83  
カーソル結果セット  
  用語定義, 84  
カーネル  
  サポートされるバージョン, 15

## き

競合  
  用語定義, 100  
競合解決  
  用語定義, 101  
強力な暗号化  
  暗号化ソフトウェアの注文, 11  
キー  
  外部, 22  
  説明, 21  
  プライマリ, 21  
キー・ジョイン  
  用語定義, 103

## く

クイック・スタート  
  SQL Anywhere 11, 68  
クエリ  
  用語定義, 84  
組  
  概要, 20  
組み込みデータベース  
  サンプル・アプリケーション, 4  
  定義済み, 30  
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, 31  
  用語定義, 84  
クライアント／サーバ限定版  
  サポートされるプラットフォーム, 15  
クライアント・メッセージ・ストア  
  用語定義, 84  
クライアント・メッセージ・ストア ID  
  用語定義, 84  
グローバル・テンポラリ・テーブル  
  用語定義, 84

## け

検査制約  
  用語定義, 101  
検証  
  用語定義, 101  
限定リリース  
  サポートされるプラットフォーム, 15  
ゲートウェイ  
  用語定義, 85

## こ

高可用性  
  別途ライセンスが必要なコンポーネント, 12  
購入  
  別途ライセンスが必要なコンポーネント, 11, 12  
コマンド・シェル  
  引用符, x  
  カッコ, x  
  環境変数, x  
  中カッコ, x  
  表記規則, x  
コマンド・ファイル  
  用語定義, 85  
コマンド・プロンプト  
  引用符, x  
  カッコ, x  
  環境変数, x  
  中カッコ, x  
  表記規則, x  
コード・ページ  
  用語定義, 85

## さ

作成者 ID  
  用語定義, 101  
サブクエリ  
  用語定義, 86  
サブスクリプション  
  用語定義, 86  
サポート  
  ニュースグループ, xii  
サポートされるプラットフォーム  
  Mobile Link, 15  
  QAnywhere, 15  
  SQL Anywhere サーバ, 15

SQL Remote, 15  
Ultra Light, 15  
説明, 15  
参照先オブジェクト  
用語定義, 101  
参照整合性  
用語定義, 101  
参照元オブジェクト  
用語定義, 101  
3 層コンピューティング  
概要, 32  
サンプル・データベース  
custdb.db のスキーマ, 62  
custdb.db の説明, 62  
demo.db について, 56  
再作成, 58  
サーバ管理要求  
用語定義, 85  
サーバ起動同期  
用語定義, 85  
サーバ・メッセージ・ストア  
用語定義, 85  
サービス  
用語定義, 85  
サービス指向アーキテクチャ  
説明, 50

## し

識別子  
用語定義, 102  
システム・オブジェクト  
用語定義, 86  
システム・テーブル  
用語定義, 86  
システムの稼働条件  
サポートされるプラットフォーム, 15  
システム・ビュー  
用語定義, 86  
述部  
用語定義, 102  
ジョイン  
用語定義, 86  
ジョイン条件  
用語定義, 87  
ジョイン・タイプ  
用語定義, 86  
照合

用語定義, 102  
詳細情報の検索／テクニカル・サポートの依頼  
テクニカル・サポート, xii  
常時利用可能  
データの同期の利点, 41  
常時利用可能なコンピューティング  
SQL Anywhere の特長, 14

## す

スキーマ  
PowerDesigner を使用したデータベース設計,  
72  
用語定義, 87  
スクリプト  
用語定義, 87  
スクリプト・バージョン  
用語定義, 87  
スクリプトベースのアップロード  
用語定義, 87  
スタンドアロン・アプリケーション  
定義済み, 30  
ストアド・プロシージャ  
用語定義, 87  
スナップショット・アイソレーション  
用語定義, 87  
スマートフォン  
SQL Anywhere データベースのサポート, 29  
Ultra Light データベースのサポート, 29

## せ

正規化  
用語定義, 103  
正規表現  
用語定義, 103  
整合性  
用語定義, 102  
生成されたジョイン条件  
用語定義, 103  
制約  
用語定義, 102  
セキュア機能  
用語定義, 87  
セキュリティ・オプション  
説明, 11  
世代番号  
用語定義, 102  
設計

- PowerDesigner の使用, 72
- セッションベースの同期
  - 同期テクノロジーの概要, 45
  - 用語定義, 88
- 接続 ID
  - 用語定義, 103
- 接続起動同期
  - 用語定義, 103
- 接続プロファイル
  - 用語定義, 103
- 接続ベースの同期
  - 同期テクノロジーの概要, 46

## そ

- 関連名
  - 用語定義, 103
- 双方向の同期
  - 同期テクノロジーの概要, 44
- 属性
  - 関係, 20

## た

- ダイレクト・ロー・ハンドリング
  - 用語定義, 88
- ダウンロード
  - 用語定義, 88
- 多層コンピューティング
  - 概要, 32
- ターゲット・プラットフォーム
  - Ultra Light, 15

## ち

- チェックサム
  - 用語定義, 88
- チェックポイント
  - 用語定義, 88
- 蓄積転送
  - SQL Remote の同期, 45
- 抽出
  - 用語定義, 104
- 抽出・変換・ロード
  - ETL, 34
- チュートリアル
  - SQL Anywhere サーバ, 70
  - デモ・データベースの再作成, 58

## つ

- 通信ストリーム
  - 用語定義, 104
- 通信プロトコル
  - SQL Anywhere, 38
- ツール
  - 設計ツールと管理ツール, 8

## て

- テクニカル・サポート
  - ニュースグループ, xii
- デッドロック
  - 用語定義, 90
- デバイス・トラッキング
  - 用語定義, 90
- デベロッパー・コミュニティ
  - ニュースグループ, xii
- デモ・データベース
  - 再作成, 58
  - 説明, 56
- 展開
  - サポートされるプラットフォーム, 15
- 転送ルール
  - 用語定義, 104
- 伝達方法
  - 同期テクノロジーの概要, 45
- テンポラリ・テーブル
  - 用語定義, 90
- テンポラリ・ファイル
  - 概要, 27
- データ・ウェアハウス
  - ETL 機能, 34
- データ型
  - 用語定義, 90
- データ可用性
  - データの同期の利点, 41
- データ・キューブ
  - 用語定義, 88
- データ交換
  - 説明, 6
- データ操作言語
  - 用語定義, 90
- データの一貫性
  - 同期テクノロジーの課題, 42
- データベース
  - SQL Anywhere と Ultra Light の比較, 29
  - オブジェクト, 23



クライアント・アプリケーション, 19  
言語インタフェース, 19  
コンポーネント, 18  
サーバ, 18  
同期, 39  
ファイル, 27  
用語定義, 88  
リレーショナル, 20  
レプリケート, 39  
データベース・オブジェクト  
説明, 23  
用語定義, 89  
データベース管理者  
用語定義, 89  
データベース・サーバ  
説明, 18  
内部, 25  
パーソナルとネットワークの違い, 18  
用語定義, 89  
データベース所有者  
用語定義, 89  
データベース設計のモデル化  
PowerDesigner の使用, 72  
データベース接続  
用語定義, 89  
データベース・テーブル  
説明, 20  
データベース同期  
説明, 39  
データベースの暗号化  
暗号化ソフトウェアの注文, 11  
データベースの同期  
説明, 39  
データベースのレプリケート  
説明, 39  
データベース・ファイル  
概要, 27  
用語定義, 89  
データベース・ミラーリング  
別途ライセンスが必要なコンポーネント, 12  
データベース名  
用語定義, 89  
データベース・レプリケーション  
説明, 39  
テーブル  
外部キー, 22  
説明, 20

特性, 20

## と

同期  
Mobile Link 機能の比較, 47  
SQL Remote と Mobile Link, 39  
SQL Remote と Mobile Link の比較, 40  
用語定義, 104  
同期テクノロジの課題  
説明, 42  
同期テクノロジの選択  
説明, 47  
統合化ログイン  
用語定義, 104  
統合データベース  
サポートされる RDBMS, 15  
用語定義, 104  
同時性 (同時実行性)  
用語定義, 105  
動的 SQL  
用語定義, 104  
特長  
SQL Anywhere, 14  
独立性レベル  
用語定義, 105  
トピック  
グラフィック・アイコン, xi  
ドメイン  
用語定義, 90  
トラブルシューティング  
ニュースグループ, xii  
トランザクション  
用語定義, 91  
トランザクション単位の整合性  
用語定義, 91  
トランザクション・テクノロジ  
同期テクノロジの課題, 42  
トランザクション・ログ  
概要, 27  
用語定義, 91  
トランザクション・ログ・ミラー  
用語定義, 91  
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, 11  
トリガ  
用語定義, 91

## な

### 内部

- データベース・サーバ, 25
- データベース・ファイル, 27

### 内部ジョイン

- 用語定義, 105

### ナチュラル・ジョイン

- 用語定義, 103

## に

### ニュースグループ

- テクニカル・サポート, xii

## ね

### ネットワーク・サーバ

- 概要, 18
- 用語定義, 91

### ネットワーク・データベース・サーバ (参照 ネットワーク・サーバ)

- ネットワーク・プロトコル
- 用語定義, 91

## は

### バグ

- フィードバックの提供, xii

### パッケージ

- 用語定義, 92

### ハッシュ

- 用語定義, 92

### パフォーマンス統計値

- 用語定義, 92

### パブリケーション

- 用語定義, 92

### パブリケーションの更新

- 用語定義, 92

### パブリッシャ

- 用語定義, 93

### ハンドヘルド PC

- Windows Mobile でサポートされるプラットフォーム, 15

### パーソナル・サーバ

- 概要, 18
- 用語定義, 92

### パーソナル・データベース・サーバ (参照 パーソナル・サーバ)

## ひ

### ビジネス・ルール

- 用語定義, 93

### ヒストグラム

- 用語定義, 93

### ビット配列

- 用語定義, 93

### ビュー

- 用語定義, 93

### 表記規則

- コマンド・シェル, x
- コマンド・プロンプト, x
- マニュアル, viii
- マニュアルでのファイル名, ix

## ふ

### ファイル共有メッセージ・タイプ

- SQL Remote がサポートする Windows プラットフォーム, 15

### ファイル定義データベース

- 用語定義, 93

### ファイルベースのダウンロード

- 用語定義, 93

### フィードバック

- エラーの報告, xii
- 更新のご要望, xii
- 提供, xii

- マニュアル, xii

### フェールオーバー

- 用語定義, 94

### 複数のデータベース

- 単一サーバでの実行, 33

### 物理インデックス

- 用語定義, 105

### 物理データ・モデル

- PowerDesigner の使用, 72

### フロントエンド環境

- 説明, 4

### プライマリ・キー

- 定義, 21
- 用語定義, 94
- 例, 21

### プライマリ・キー制約

- 用語定義, 94

### プライマリ・テーブル

- 用語定義, 94

プラグイン・モジュール

用語定義, 94

プラットフォーム

サポートされるオペレーティング・システム,  
15

プラットフォームのサポート

説明, 15

フル・バックアップ

用語定義, 94

プロキシ・テーブル

用語定義, 94

プログラミング・インタフェース

説明, 35

プロセッサ数

ライセンス, 10

文レベルのトリガ

用語定義, 105

## へ

別途ライセンスが必要なコンポーネント

説明, 11

ヘルプ

テクニカル・サポート, xii

ヘルプへのアクセス

テクニカル・サポート, xii

ベース・テーブル

説明, 20

用語定義, 95

## ほ

ホスト・プラットフォーム

サポートされるプラットフォーム, 15

ポリシー

用語定義, 95

ポーリング

用語定義, 95

## ま

マテリアライズド・ビュー

ETL 機能, 34

用語定義, 95

マニュアル

SQL Anywhere, vi

表記規則, viii

マルチユーザ・データベース

定義済み, 31

## み

ミラーリング

別途ライセンスが必要なコンポーネント, 12

ミラー・ログ

用語定義, 95

## め

メタデータ

用語定義, 95

メッセージ・システム

用語定義, 95

メッセージ・ストア

用語定義, 96

メッセージ・タイプ

用語定義, 96

メッセージベースの同期

同期テクノロジの概要, 45

メッセージ・ログ

用語定義, 96

メンテナンス・リリース

用語定義, 96

## も

文字セット

用語定義, 105

文字列リテラル

用語定義, 106

モバイル Web サービス

利点, 50

モバイル・エンタープライズ・メッセージング

説明, 49

モバイル・コンピューティング

サンプル・アプリケーション, 4

## ゆ

ユーザ定義データ型

用語定義, 96

## よ

用語解説

SQL Anywhere の用語一覧, 75

## ら

ライセンス

CPU 数, 10

プロセッサ数, 10

別途ライセンスが必要なコンポーネント, 11

## り

リダイレクタ

サポートされる Windows プラットフォーム,  
15

用語定義, 97

リハビリテーション法

アクセシビリティ有効化モジュール, 16

リファレンス・データベース

用語定義, 97

リモート ID

用語定義, 97

リモート・データ・アクセス

サポートされるプラットフォーム, 15

リモート・データベース

用語定義, 97

リレーショナル・データベース

説明, 20

リレーショナル・データベース・システム

概念, 20

説明, 6

## れ

レプリケーション

SQL Remote と Mobile Link, 39

SQL Remote と Mobile Link の比較, 40

用語定義, 97

レプリケーションの頻度

用語定義, 98

レプリケーション・メッセージ

用語定義, 97

## ろ

ログ・ファイル

用語定義, 99

ロック

用語定義, 99

論理インデックス

用語定義, 106

ロー

説明, 20

ローカル・テンポラリ・テーブル

用語定義, 98

ロール

用語定義, 98

ロールバック・ログ

用語定義, 98

ロール名

用語定義, 98

ロー・レベルのトリガ

用語定義, 98

## わ

ワークグループ・コンピューティング  
(参照組み込みデータベース)

ワーク・テーブル

用語定義, 99