

SAP SQL Anywhere 概要

SAPジャパン株式会社
2016年09月28日





SAP SQL Anywhere とは

SQL Anywhere とは、標準SQLに準拠した普通のRDBMSです。

SAP SQL Anywhere は、標準SQLに準拠し、スタンドアロンから、クライアント・サーバー型またはWebのシステムまで対応が可能な汎用RDBMS「SQL Anywhere」と、異種のRDBMS間のデータ同期およびデータ交換のテクノロジー「Mobile Link」などを含むスイート製品です。

「SQL Anywhere」は、使いやすく、**自動管理、自動チューニングの機能**を豊富に揃えているため、短期間のアプリケーション開発、手離れ良い運用を実現します。

自己学習機能（クエリ・オプティマイザ）、データベースエリアの自動拡張、データベース自動起動/停止、タスクスケジューリングとイベント処理、自動リカバリ機能、キャッシュサイズの自動拡張・縮小、移動中DBの断片化解消機能、クエリ間とクエリ内の2種類の並列処理、スナップショットアイソレーション、自動マルチプログラミングレベルチューニング、カラム統計の自己回復、その他複数の自動管理機能を装備

データベース管理者を置けない遠隔地のサーバーでも問題なく運用でき、またパッケージに含まれる「Mobile Link」を使うことで、ネットワークが不安定な遠隔地のSQL AnywhereデータベースサーバーとASE, Oracle, SQL Server, DB2, HANA など本社データベースサーバーとを随時データ同期させ、業務を止めないAlways Available なシステムを実現できます。

「SQL Anywhere」データベースのサブセットであるより小型の「Ultra Light」データベースを利用すれば、オフライン利用が可能なモバイルアプリケーションの開発も可能です。

含まれるコンポーネント



今回はSQL Anywhereに関する概要紹介です。
Ultra Light/Mobile Linkに関しては別の機会にて. . . .



SAP SQL Anywhere の主な機能

主な機能 : SQL Anywhere

DB機能

- 主要データタイプをフルサポート
- Transact-SQL と Watcom SQL の2種使用可
- 豊富なプログラミングインターフェース
- パラレル・クエリ（内部並列クエリ処理）
- OLAP系クエリー対応
- マテリアライズド・ビュー

可用性機能

- DBミラーリング機能(HA/DRクラスタ)
- 読み込み専用スケールアウト
- ログミラーリング機能
- ハードウェアリソース変動対応
- クラウド対応

運用系機能

- セルフチューニング ・ インメモリ・モード
- データベース全体/テーブルレベル/カラムレベル暗号化
- 通信暗号化 ・ カラムレベルのデータ圧縮
- 少リソース動作(最低20MBのメモリ)
- イベント(&ハンドリング)機能 ・ スケジューラー機能

SAP
SQL Anywhere



連携機能

- 統合 HTTP サーバー機能
- ODataプロデューサー
- リモートデータアクセス
- 外部認証（統合化認証）対応
- 外部環境プロシージャラー

SQL Anywhereの主な機能の紹介

- **データタイプをフルサポートしたDBMSです。**

- 標準タイプ

CHAR, LONG NVARCHAR, LONG VARCHAR, NCHAR, NTEXT, NVARCHAR, TEXT, UNIQUEIDENTIFIERSTR, VARCHAR, BIGINT, BIT, DECIMAL
DOUBLE, FLOAT, INTEGER, NUMERIC, REAL, SMALLINT, TINYINT, MONEY, SMALLMONEY, LONG
VARBIT, VARBIT, DATE, DATETIME, DATETIMEOFFSET, SMALLDATETIME, TIME
TIMESTAMP, TIMESTAMP WITH TIME ZONE, BINARY, IMAGE, LONG BINARY, UNIQUEIDENTIFIER, VARBINARY

- 空間エンジンによる空間データタイプ

ST_Point, ST_LineString, ST_Polygon, ST_CircularString, ST_CompoundCurve, ST_CurvePolygon, ST_Geometry, ST_GeomCollection, ST_MultiPoint, ST_MultiPolygon, ST_MultiLineString, ST_MultiSurface

- 全文検索：述語、ワイルドカード、類似検索を含む
 - OPENXML を介した XPATH クエリーによる XML タイプ

- **2つのSQL**

- 方言としては Transact-SQL と Watcom SQL の2種が使用可能
 - 1つのDB内で混在可能です。
 - プロシジャAはTransact SQL、プロシジャBはWATCOMということはOK
 - プロシジャAのコードの前半はTransact、後半がWATCOMはNG
 - SQL:2008準拠（1992,1999,2003）



提供プログラミングインターフェース

SAP SQL Anywhere では下記のアプリケーションプログラミングインタフェースを提供しています。

- **ADO.NET**
 - Microsoft .NET Framework バージョン 2.0、3.0、3.5、4.0、4.5に対応
 - Entity Framework 5.0 および 6.0対応 (Visual Studio 2015対応)
- **JDBC**
 - Type2とType4の2種類提供
- **ODBC**
- **OLE DB / ADO**
- **Embedded SQL**
- **SQL Anywhere データベース API for C/C++**
- **Open Client**
- **OData**
- **XS JavaScript**
- **Node.js**
- **Perl DBI**
- **Python**
- **Ruby**

SQL Anywhereの主な機能

- **動的パラメータ管理、セルフチューニング**

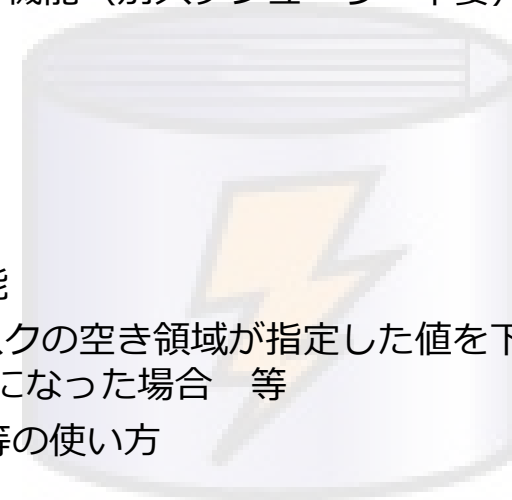
- キャッシュサイズ、同時実行タスク数等リソース関連のパラメータは自動調整を行うものが殆どです。
- 動作中のOSのメモリ使用状況を監視し、ユーザー要求処理時にメモリに余裕があるようであればキャッシュを追加、ユーザー処理要求が少ない状態でOSのメモリ使用状況が逼迫した状態であれば余剰を開放
- データベースのスループットを監視し、同時実行処理数を最も効率の良いスループットになるよう自動的に変動。（山登り法と放物線近似法を組み合わせて自動調整）

- **スケジュール機能**

- DBMS自体が時間指定で決められたコマンド・プロシジャを実行するスケジューリング機能（別スケジューラー不要）
- バックアップ等の自動化
- バッチ処理の自動化、月次処理で集計テーブルを作成する等

- **イベント機能（&ハンドリング機能）**

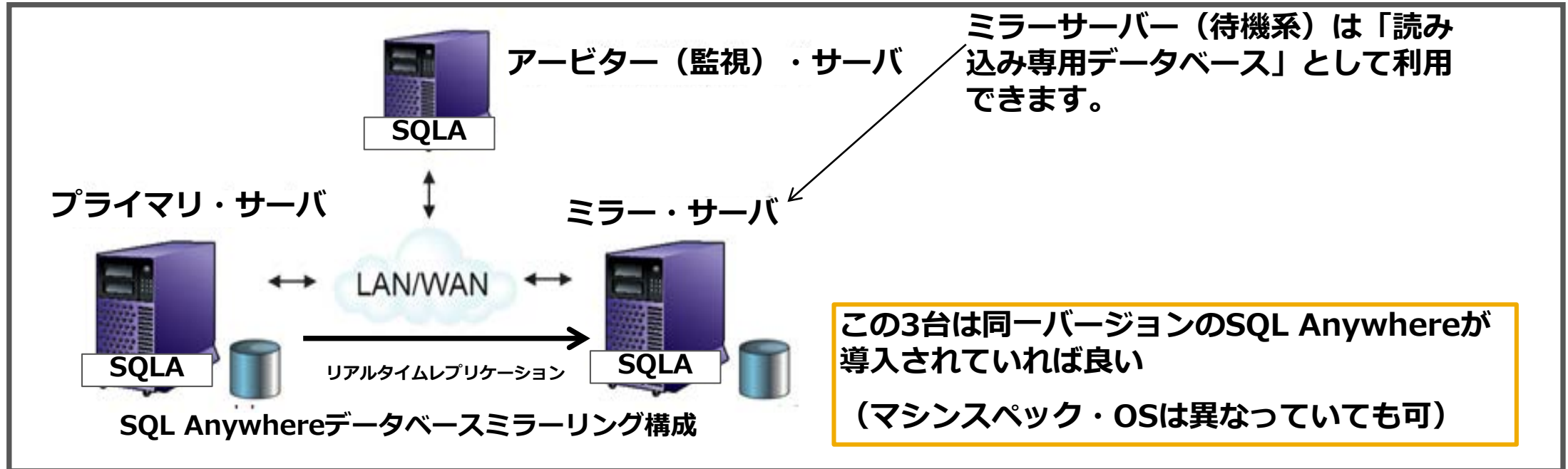
- 予め定めたイベントが発生した場合に決められたコマンド・プロシジャを実行する機能
- DBが起動・終了する際、ユーザーが接続・切断した際、DBファイルを配置したディスクの空き領域が指定した値を下回った際、DBのファイルサイズが指定したサイズに達した場合、指定した時間以上DBがアイドル状態になった場合 等
- アイドル状態になったらテーブルの断片化をチェックして必要であれば再編成 等の使い方



SQL Anywhereの主な機能

データベースミラーリング機能

SQL Anywhereの機能による高可用性クラスタ・DRクラスタ構築機能です。**特別なハードウェアや他のソフトウェア、クラスタリングソフトウェアは必要ありません。**主系・待機系としてDBが動作するマシン2台とそれら2台を監視する小さなプロセスが動作するマシンの計3台で動作します。



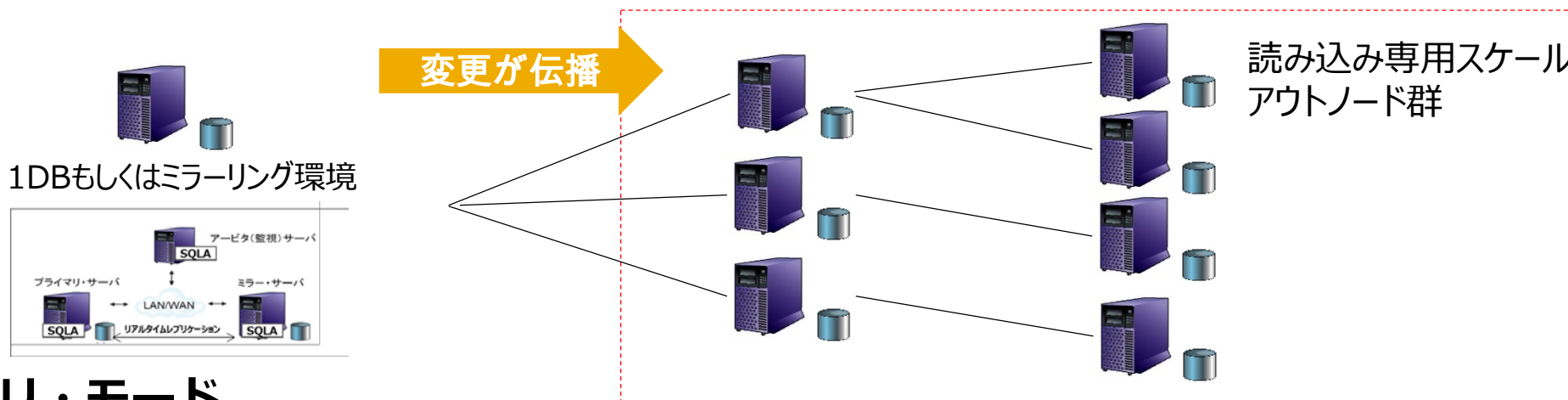
プライマリ→ミラー間はトランザクションログ送信によるリアルタイムレプリケーションです。

- ログの送信・反映タイミング違いの3モード

SQL Anywhereの主な機能

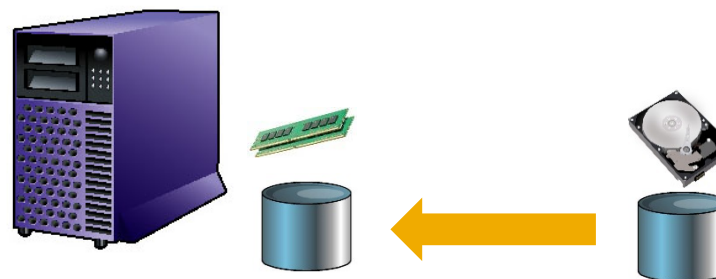
読み込み専用スケールアウト

- データベースを複製したノードを複数作成し、レポート作成やデータ参照など多数の要求に対する負荷軽減やデータ配布に使用できます。



インメモリ・モード

- データベースファイルをメモリ上に展開することによりI/Oを高速に
- 書き込み系処理に効果を発揮
- データの保全レベルを選択可
 - 完全インメモリ
 - 非同期遅延ディスク書き込み
 - ログのみトランザクションと同期的にディスク書き込み



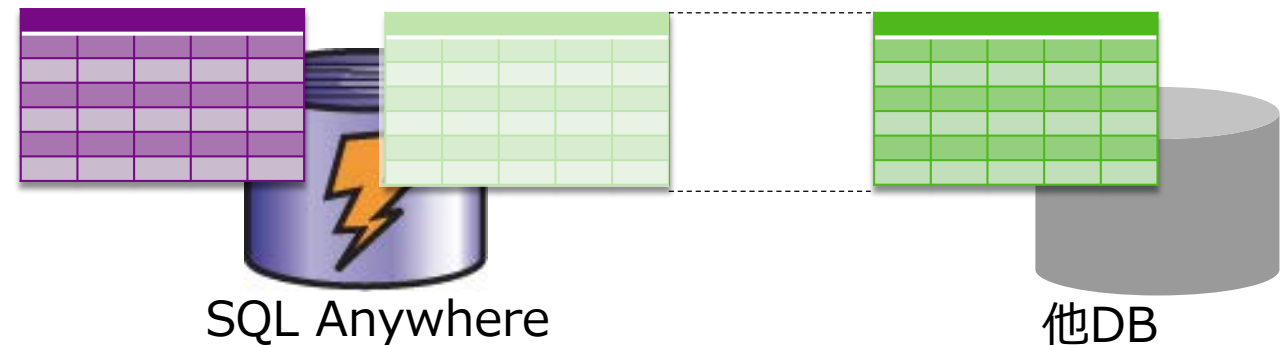
SQL Anywhereの主な機能

- **統合 HTTP サーバー機能**

- SAP SQL Anywhere サーバー自体をHTTPサーバとして利用可
- プロシジャでWebアプリケーションを組むことが可能
 - HTTPサーバのURLに返り値としてHTMLを返すプロシジャを割当て→アクセスで実行
 - 小型Web アプリケーションの自己完結型のデプロイメント
- SQL Anywhere単体でOData Producerに。

- **リモートデータアクセス**

- リンクテーブル、データベースリンク…等と呼ばれる他DBのテーブルを自DBにあるかのように見せる機能
 - SQL Anywhere、ASE、IQ、SAP HANA
 - IBM DB2 / Microsoft Access / Microsoft SQL Server
 - MySQL/Oracle Database /その他ODBC



SQL Anywhere サポートプラットフォーム

Windows

Windows 7 x86 and x64

Windows 8.x x86 and x64

Windows 10 x86 and x64

Windows Server 2003 x86 and x64

Windows Server 2008 x86 and x64

Windows Server 2008 R2 x64

Windows Server 2012 x64

Windows Server 2012 R2 x64

Windows Mobile for Pocket PC and Smartphone

Windows Mobile 6.x Classic, Professional and Standard (ARMv4i)

Windows CE 5.x and 6.x (ARMv4i)

Linux for x86 and x64

Red Hat Enterprise Linux 5/6/7

SUSE Linux Enterprise 11/12

Ubuntu 12.04

Linux for ARM

Raspberry Pi on Debian GNU/Linux

Olimex A20 on Debian GNU/Linux

Solaris

10 SPARC and x64

11 SPARC and x64

HP-UX

11i v2 IA64 on Itanium 2

11i v3 IA64 on Itanium 2

AIX

6.1 POWER

7.1 POWER

Mac OS X

10.9 x64

10.10 x64

参考) Ultra Light

Windows

Windows PC版はSQL Anywhereに準ずる

Windows Mobile 5 for Pocket PC and Smartphone
(Windows CE 5.0) ARMv4i

Windows Mobile 6.x Classic, Professional and Standard
(Windows CE 5.x) ARMv4i

Windows CE 6.0 ARMv4i

Windows Phone 8.1 ARM/x86(for emulator only)

Universal Windows Platform apps (UWP) ARM, x86, x64
(use WinRT components)

Android

4.1 or later ARM v7,x86, x86_64

iOS

Current version

Linux

SQL Anywhere に準ずる (x64)

Mac OS X

10.9+ x86_64

<http://scn.sap.com/docs/DOC-35654>
を御参照下さい



SAP SQL Anywhere の便利な機能

簡単なセットアップ & 配布

SQL Anywhereエンジンの配布

SQL Anywhereエンジンの配布（お客様環境へのインストール）方法は通常のインストーラーを実行する方法の他に以下の方法が使用できます。

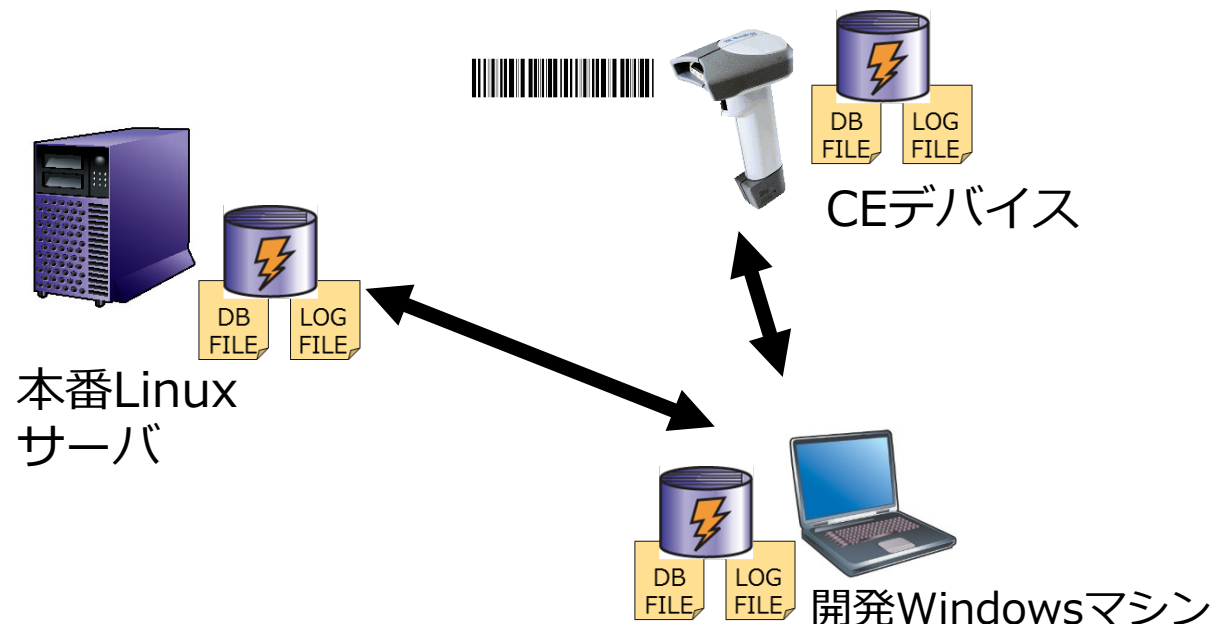
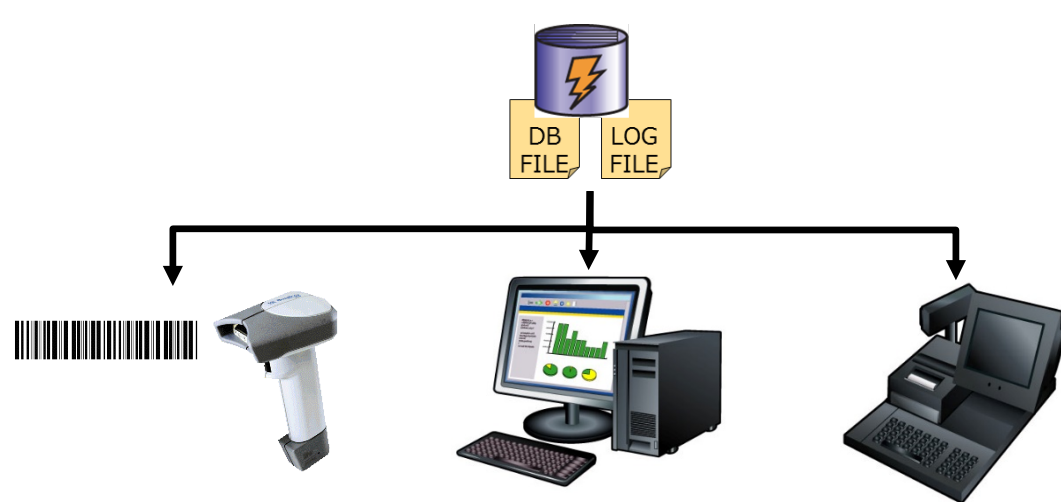
サイレントインストール	MSI/MSMインストーラー	ファイル配布
通常のインストールに使用する setup.exe に引数を付与してインストールすることで、画面にインストール中の表示をせずにインストールが可能です。	開発環境にて必要なコンポーネントのみを含んでいる MSI インストーラーの作成が可能です。 他の MSI インストーラーに含める形でインストールできるようにする MSM 形式での作成も可能で、アプリケーションのインストーラーへ統合する形で SQL Anywhere のインストールが可能です。 * Windows 版のみ	既にインストールされた環境（もしくは開発環境）にて必要なファイルをコピーし、それをそのまま配布することでもインストールが可能です。 必要なファイルをパスが通ったディレクトリへ配置して下さい。 アプリケーションのインストーラーの配布ファイルに含めてインストールが可能です。 Windows x64 版の最低限のファイルは 30MB 程です。

同一マシンに別バージョンの共存は問題ありません。

データベースファイルの配布

SQL Anywhereのデータベースを構成するファイルは「データファイル」と「ログファイル」の2種

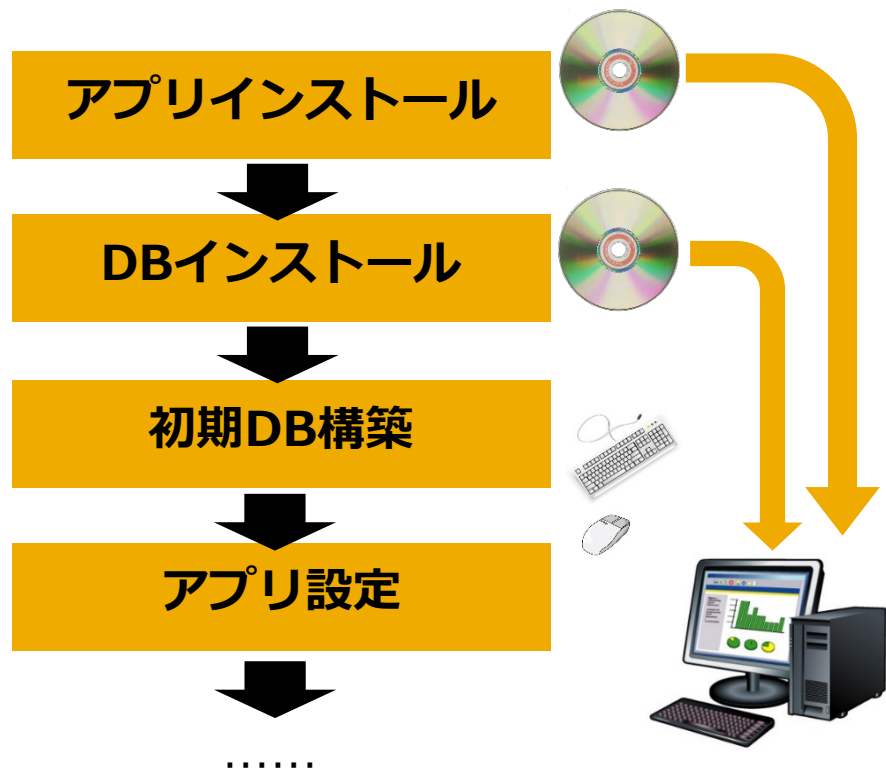
- データファイル：データを格納、ログファイル：トランザクションログを格納
- データファイルは最大13個
 - 最大1ファイル8TB (or OS / ファイルシステムの最大ファイルサイズ) *13個
- ファイルコピーでの配布が可能です。
- DBを作成したOSでないOSへのコピーも可能です。(DBファイルとログファイルはプラットフォームに依存しません)



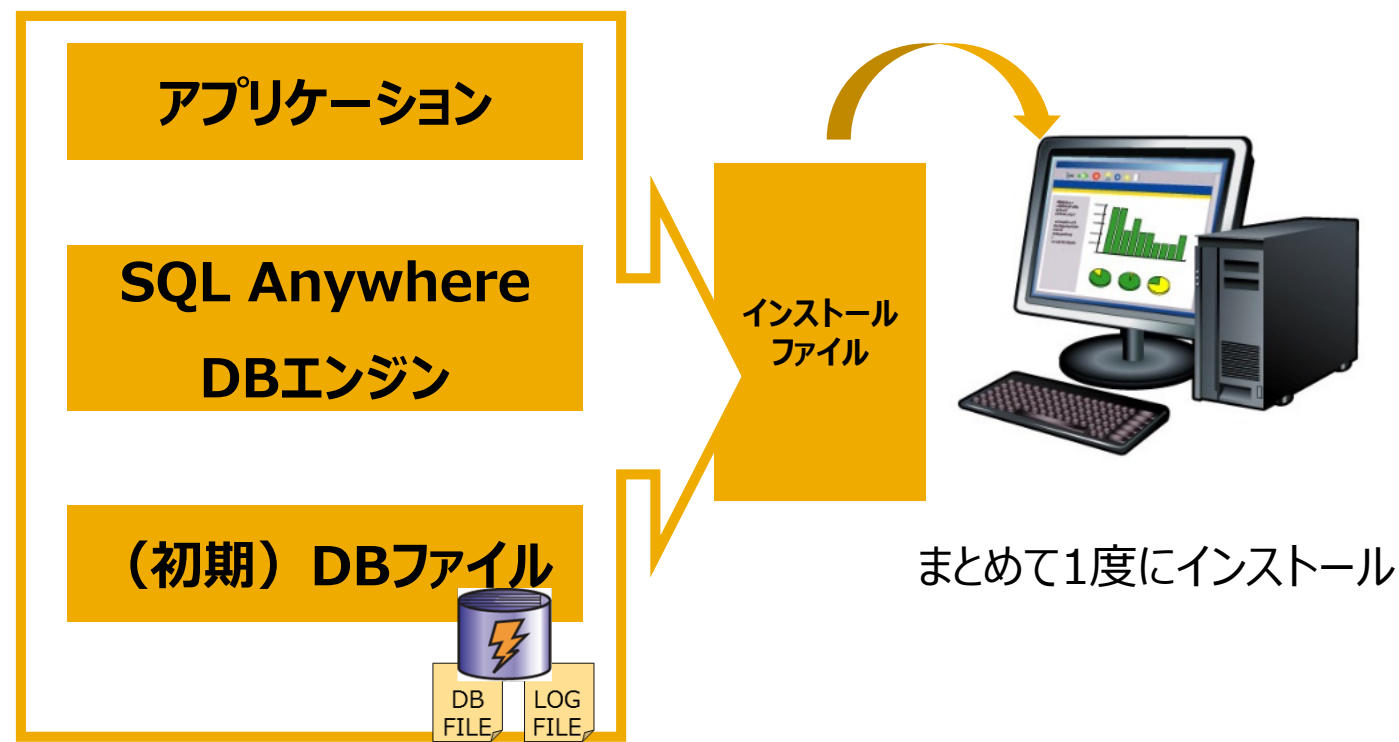
まとめて配布出来ます！

- テーブルやインデックス、プロシジャ、初期データを含んだデータベースファイルをアプリケーションのインストーラーファイルに含めて配布すればそれだけで使用可能です。
- データベースにテーブルなどを作成するスクリプトの作成の手間を省くことが可能です。

普通のDB+アプリのシステムなら



SQL Anywhereなら



ツール類の機能の御紹介

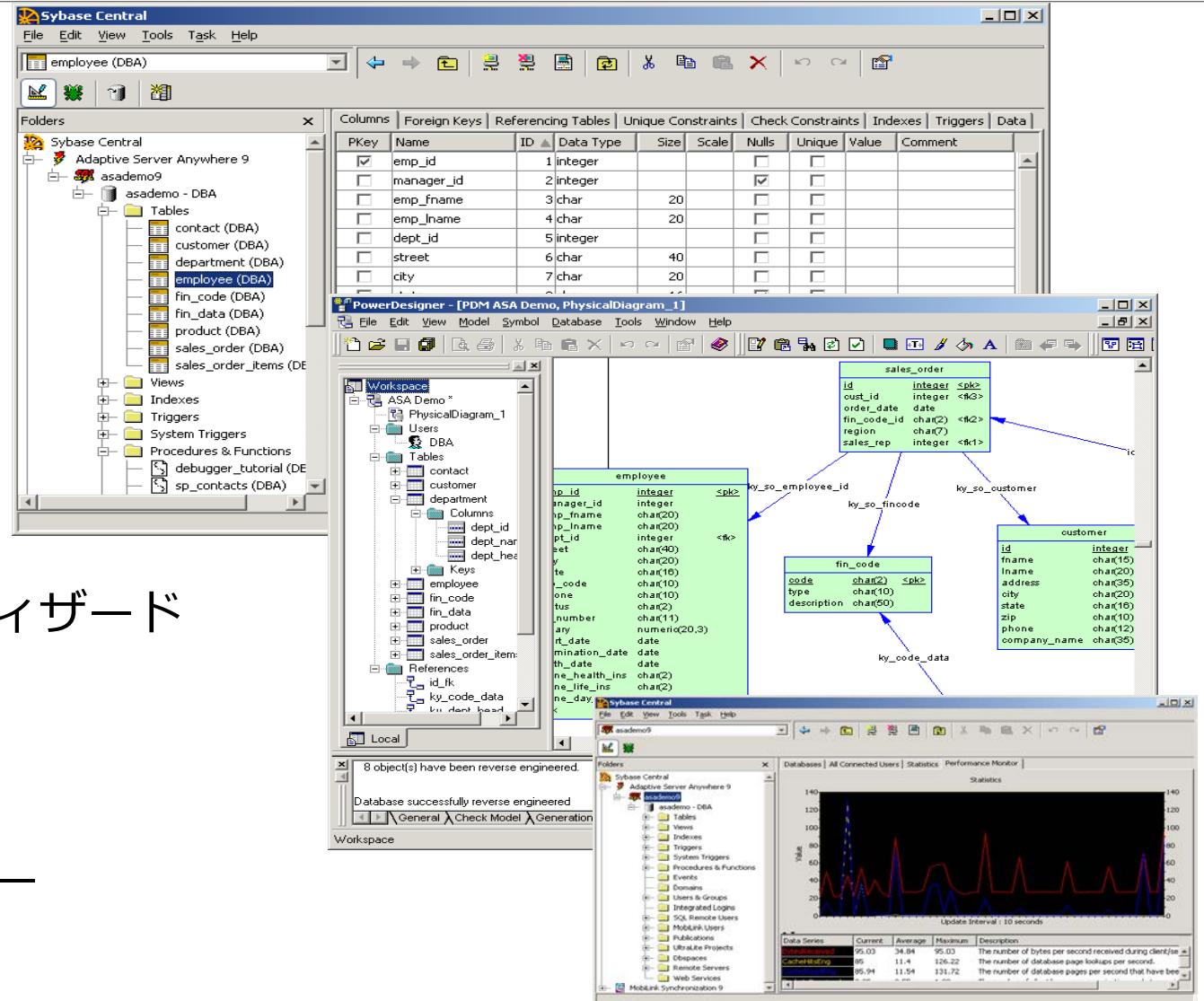
使いやすい設計・管理ツールを同梱

- **管理とモニタリング**

- Sybase Central(SQL Central)
- データベースモニター
- 監査

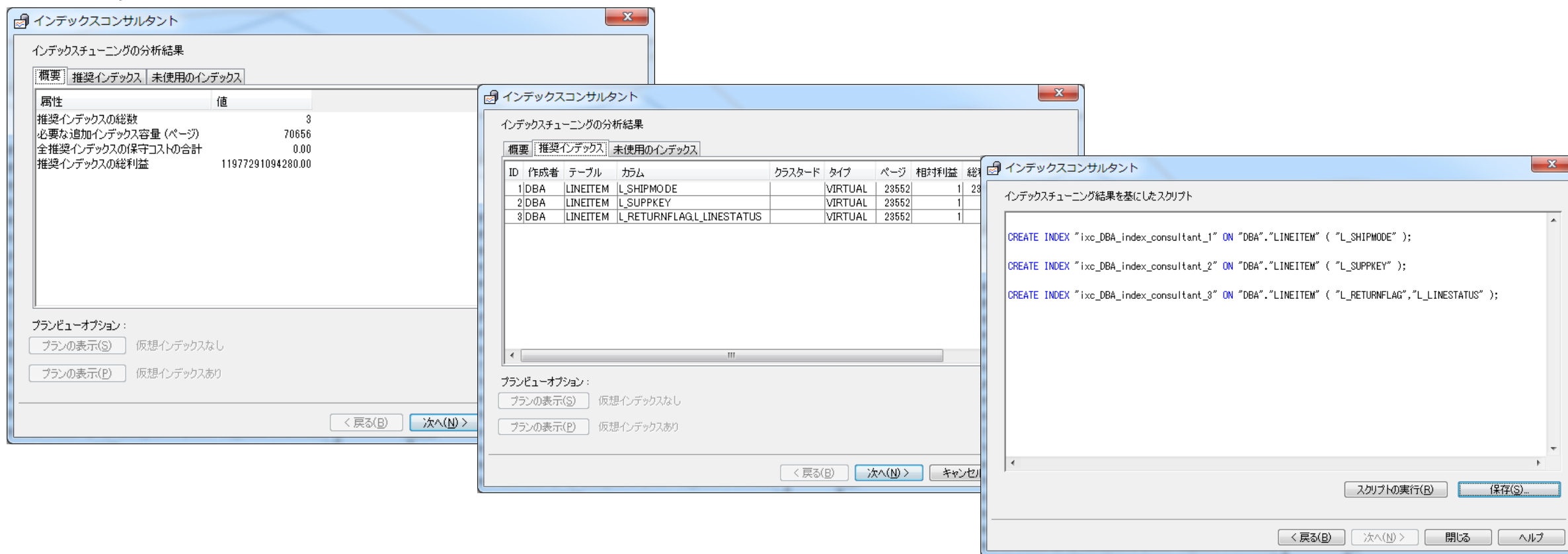
- **設計と開発**

- アプリケーションプロファイリングウィザード
- Interactive SQL (dbisql)
- データベースオブジェクトデバッガー
- インデックスコンサルタント
- ストアドプロシージャプロファイラー



SQLチューニングツール インデックスコンサルタント

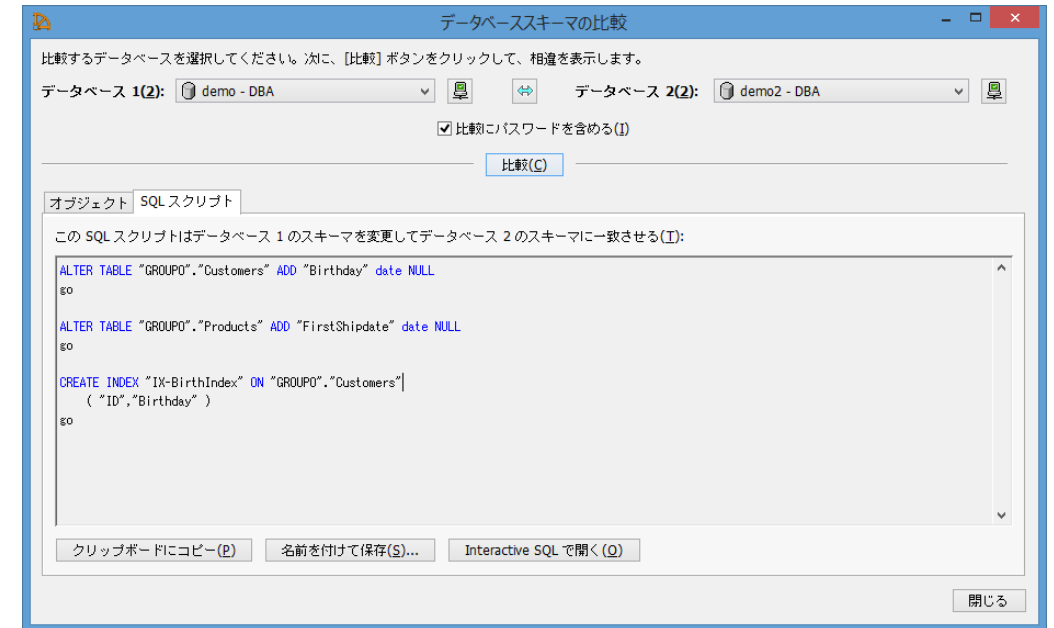
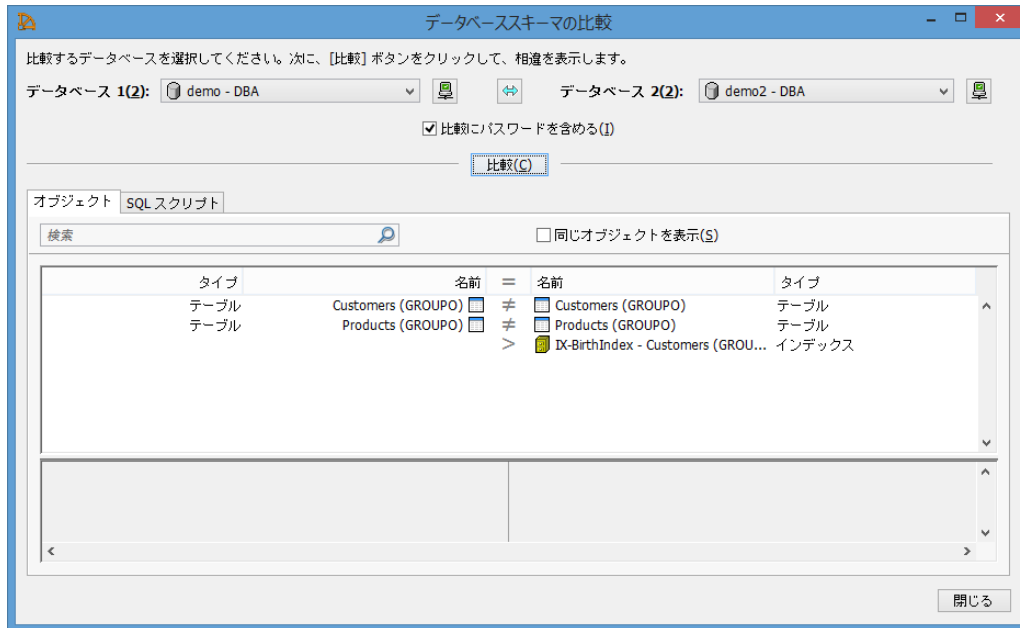
指定したSQL、もしくは実行中のDBのアクティビティをプロファイリングした内容から「このインデックスがあれば処理が高速化出来る」と判断したものを提案する機能です。



データベーススキーマの比較機能 (スキーマ差分生成機能)

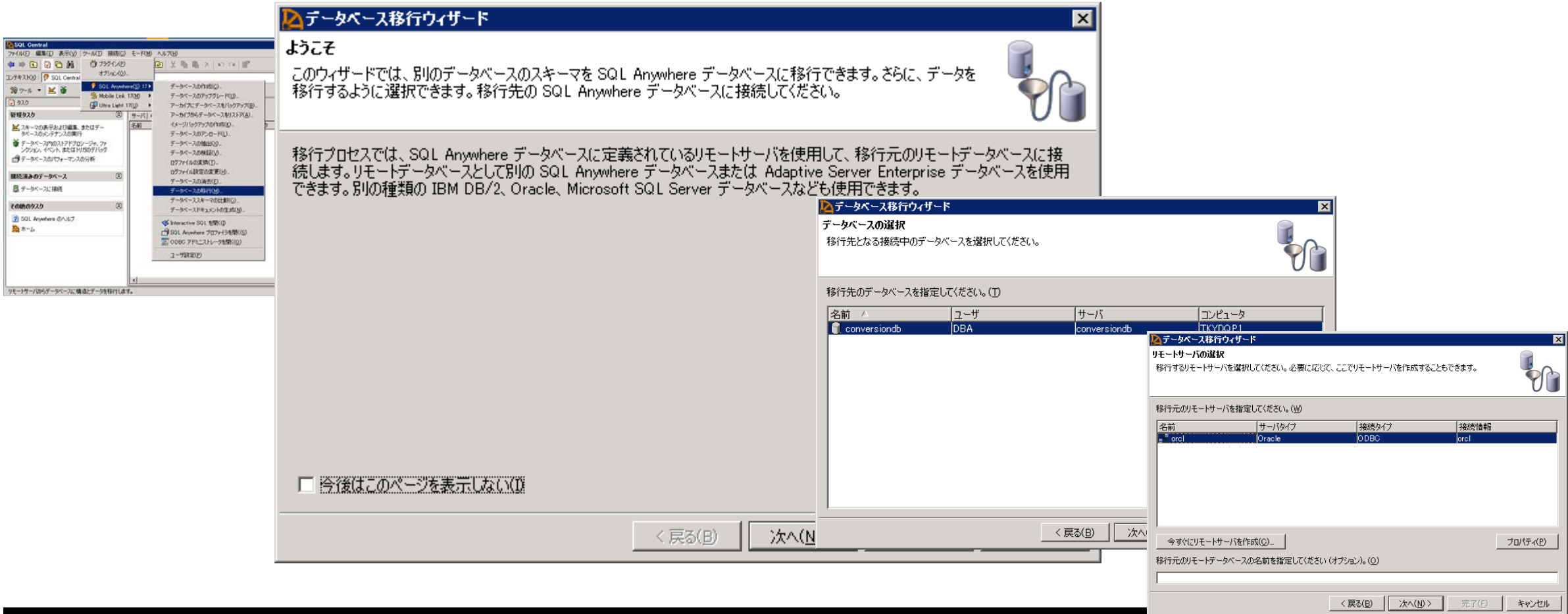
継続開発の際に、データベースのスキーマが変更される場合があります。SQL Centralにはスキーマの比較機能があり、2つのDBの間でスキーマを分析・比較し、違いを出力する事が出来ます。

- この「違い」を合致させる為のDDLスクリプトを生成することが可能です。
- 違うならAlter文、存在しないならCreate文、消えたならDelete文…
- アプリのバージョンアップでスキーマ変更がかかる場合に、変更するための文を生成できます。

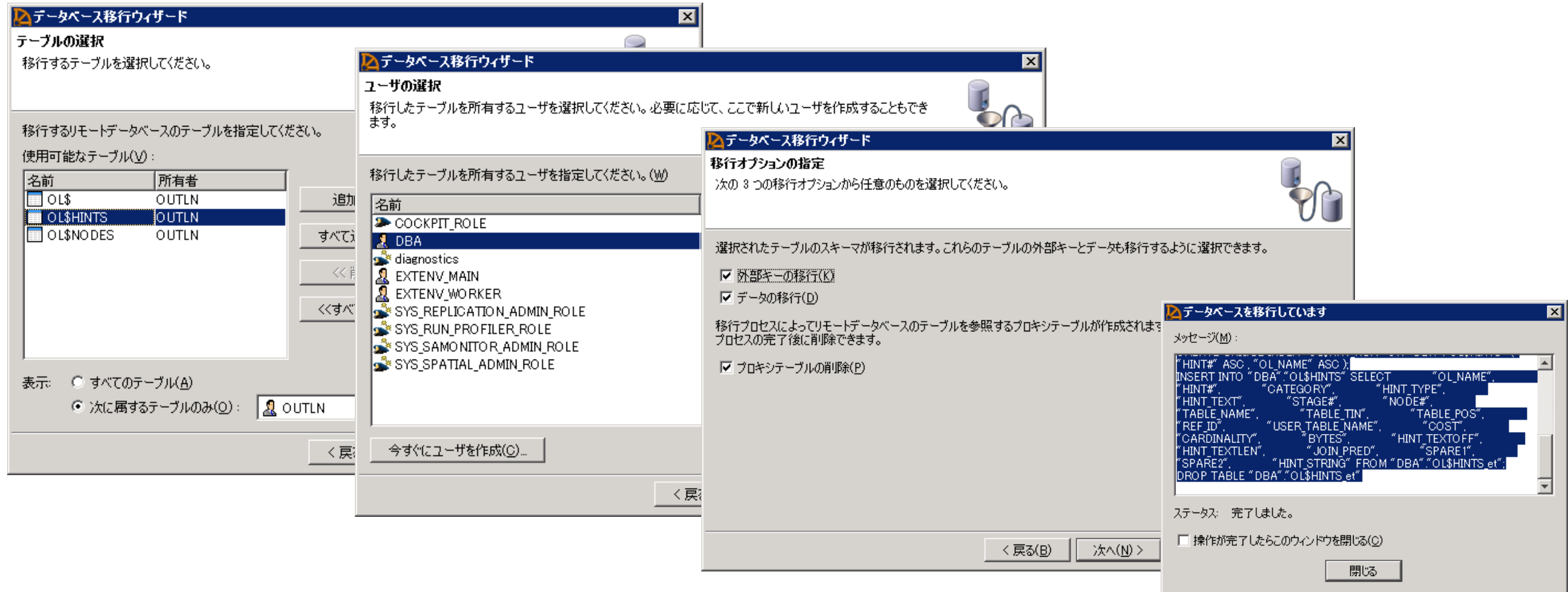


データベース移行機能

他社のDBのテーブル・インデックス・データをSQL Anywhereデータベースへ移行する機能がSQL Centralには含まれます。



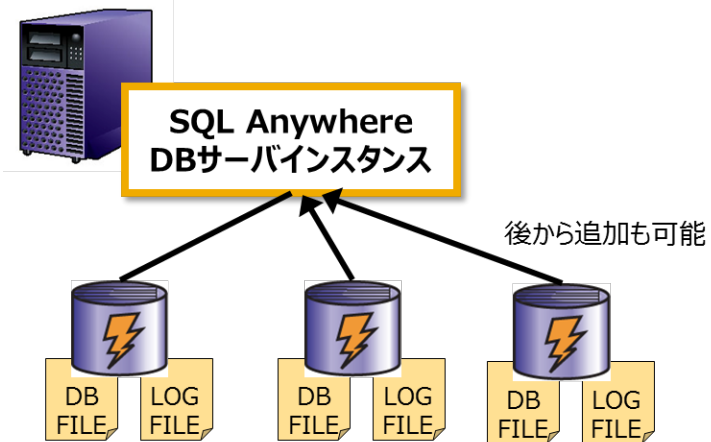
テーブル内の各カラムはSQL Anywhereへ互換性の有る型で移行されます。インデックス・外部キー、並びにデータも移行が可能です。



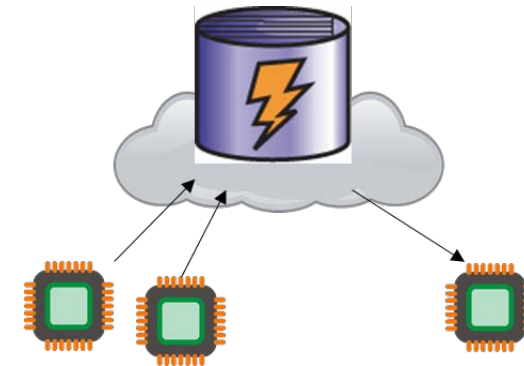
SQL Anywhereはクラウド環境をサポート

SQL Anywhereのサポートは“OS”を基準にしたものであり、その環境が仮想環境か物理環境かは問いません。

- クラウドサービスがPaas/Iaasとして提供している環境もサポート対象です。
 - SQL AnywhereはDBファイルとログファイルで構成されるDBをサーバインスタンス上で動作させるという構造です。
 - CPUコア数の動的変動のサポート
仮想環境等やクラウドで発生しうるCPUコア数の動的変動をサポートします。



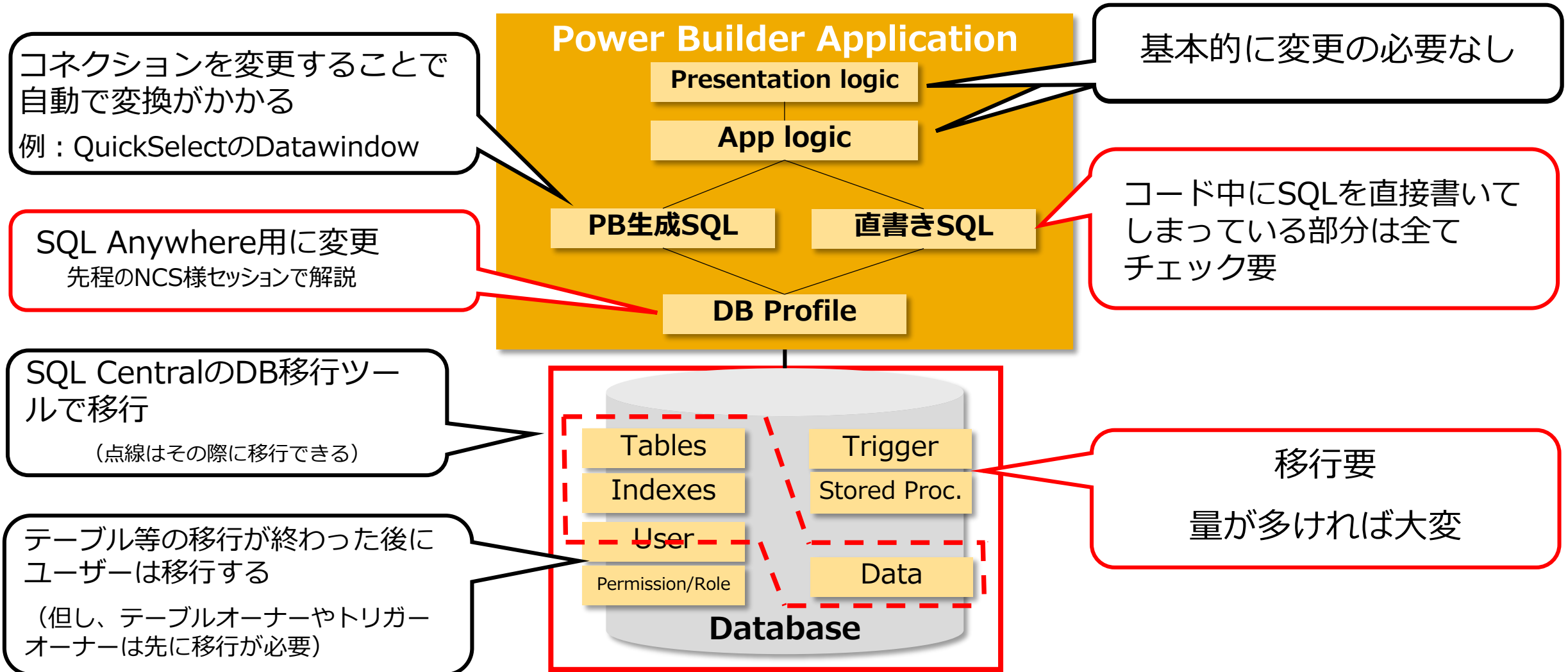
テナントのように使用出来ます。



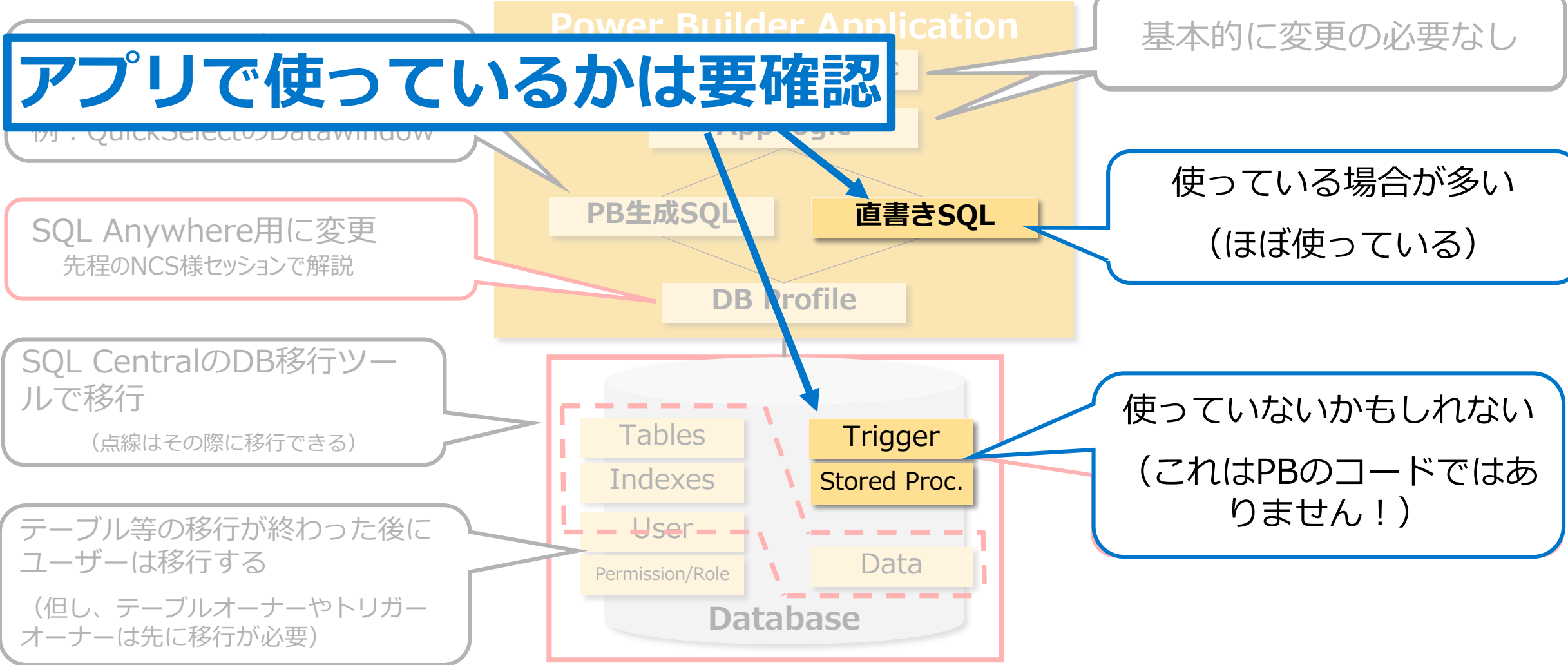
コア数の変動に応じた最適な動作を提供します。

PowerBuilderアプリケーション 移行手順概要

Power BuilderアプリケーションでのDB移行で移行するもの (From Oracle)



移行するアプリで確認してください！



大まかな移行順番

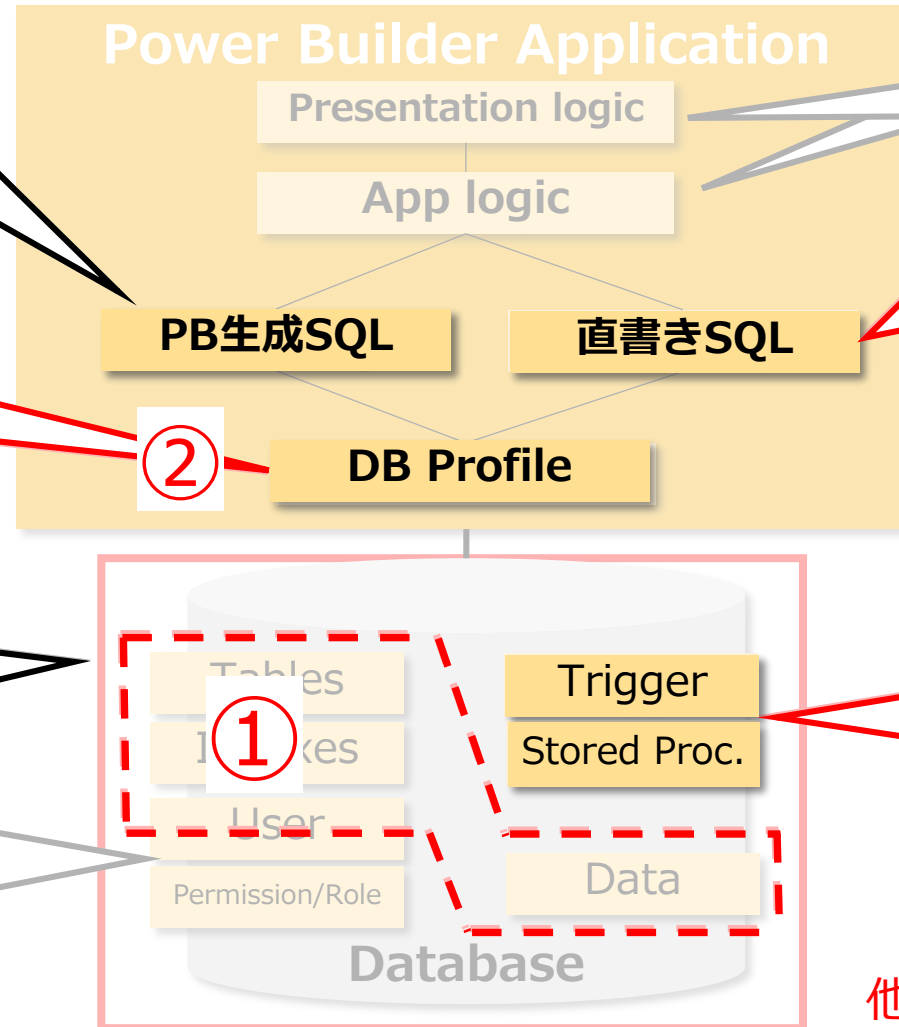
① & ③は別添資料で概要を解説

コネクションを変更することで
自動で変換がかかる
例：QuickSelectのDatawindow

SQL Anywhere用に変更
先程のNCS様セッションで解説

SQL CentralのDB移行ツールで移行
(点線はその際に移行できる)

テーブル等の移行が終わった後に
ユーザーは移行する
(但し、テーブルオーナーやトリガー
オーナーは先に移行が必要)



基本的に変更の必要なし

コード中にSQLを直接書いて
しまっている部分は全て
チェック要

③ 同時進行可能、あるいは
関係が必要

移行要
量が多ければ大変

他の項目は④となる (順序に決まりはない)

SQLの移行とポイント 概略版

SQLの移行 (From Oracle)

- SQL AnywhereのSQLは構文としてANSI書式、Watcom SQL書式、Transact SQL書式を使用することが出来ます。
- OracleではANSI書式とOracle書式（PL/SQL書式）を使用することが出来ます。



双方で利用できるANSISQLに変換することが移行の第1ステップです。

重要：“移行”においては“SQLが同じ結果を返す”ことが重要です。

- SQL結果確認テスト用にOracleのデータベースを用意しておいて下さい。
 - 9i以上を用意するのが良い
 - 9i以上はANSISQLが使えるので移行途中の確認が可能、それ以前の場合途中の確認が難しい。

Step1:ANSIJOINへの変換 (+)(-)による結合

Oracleの独特な書式として(+), (-)によるテーブルの結合があります。これはOracle独自のものであり、他のRDBMSでは動作しません。この書式をSQL Anywhereでも使用できるようにANSI書式に直す必要があります。

もし、そのSQLが (+) (-)を使用していなければ、この項はスキップできます。

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B
WHERE
A.COL1 = B.COL1(+)
AND A.COL3 = 'xxxxxx'
```

まずはSQLにこのような(+), (-)が有るかを確認して下さい。

- これはSQL文の構造から変わる変更であり、文字列置換のレベルで機械的に置換することが不可能です。
- 但し、現在は必ずしもこの書式が使用されているわけではありません。Oracleでも9以降ではANSI書式が使用可能で、最近のバージョンでは(+)(-)の結合ではなくANSI書式の使用を推奨しています。

(-)はあまり使われないのでこの資料では(+)に絞って解説します。

(-)は基本的に(-)を削除するだけで対応可能です。

Step1 : ANSIJOINへの変換

(+) の外部結合をANSI形式のジョインに直します。

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B
WHERE A.COL1 = B.COL1(+)
AND A.COL3 = 'xxxxxx'
```



ANSI書式への変換

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON
A.COL1 = B.COL1
WHERE A.COL3 = 'xxxxxx'
```

変換の基本

- 右辺に(+)なら左辺のテーブルにLEFT OUTER JOIN
- 左辺に(+)なら右辺のテーブルにRIGHT OUTER JOIN
 - FROMでのテーブルの順番は合わせる
- ON句にそのテーブル同士を結合する条件を指定
- テーブル同士の結合に関わらない条件はWHERE句の後に指定

Step1の完了

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B
WHERE A.COL1 = B.COL1(+)
AND A.COL3 = 'xxxxxx'
```



ANSI書式への変換

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON
A.COL1 = B.COL1
WHERE A.COL3 = 'xxxxxx'
```

変換が完了したら、**OracleDB上**でsql*plusなどから変換後のSQLを実行し、同じ結果が出力されるか確認してください（上記の場合、DECODE関数の問題でこの時点ではSQL Anywhere上で実行できません）
同じ結果になればStep2へ。そうでなければどこかが間違っているの見直しましょう。

Step1:ANSIジョインへの変換・様々なパターン

- 左辺が(+)の場合はRIGHT OUTER JOIN

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B
WHERE A.COL1(+) = B.COL1
AND A.COL3 = 'xxxxxx'
```

ANSI変換



```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A RIGHT OUTER JOIN TABLEB B on A.COL1
= B.COL1
WHERE A.COL3 = 'xxxxxx'
```

※勿論等価になるよう左右を入れ替えても良い

- 複数条件の場合はon句の後を（ ）で囲んでAND/OR

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B
WHERE A.COL1 = B.COL1(+) AND A.COL2=B.COL2(+)
AND A.COL3 = 'xxxxxx'
```

ANSI変換



```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A LEFT OUTER JOIN TABLEB B on ( A.COL1
= B.COL1 AND A.COL2=B.COL2)
WHERE A.COL3 = 'xxxxxx'
```

- 複数条件で片方が(+)を忘れている場合は内部結合になってしまう。(Oracleでエラーにならない)

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B
WHERE A.COL1 = B.COL1(+) AND A.COL2=B.COL2
AND A.COL3 = 'xxxxxx'
```

ANSI変換



```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA ,TABLEB B WHERE A.COL1 = B.COL1 AND
A.COL2=B.COL2
WHERE A.COL3 = 'xxxxxx'
```

※この場合は(+)を消すだけで対処可能だが、これが意図的なものかは確認が必要。(潜在的なバグの可能性はある)

- 3表の場合

```
SELECT A.COL1,B.COL2,C.COL4,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B,TABLEC C
WHERE A.COL1 = B.COL1(+) AND B.COL1 = C.COL1(+)
AND A.COL3 = 'xxxxxx'
```

ANSI変換



```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON A.COL1 =
B.COL1 LEFT OUTER JOIN TABLEC C ON B.COL1 =C.COL1
WHERE A.COL3 = 'xxxxxx'
```

※OUTER JOINした後にさらにOUTER JOINする

- 3表の場合の(+)忘れて内部結合になるパターン（Oracleでエラーにならない）

```
SELECT A.COL1,B.COL2,C.COL4,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B,TABLEC C
WHERE A.COL1 = B.COL1(+) AND B.COL1 = C.COL1
AND A.COL3 = 'xxxxxx'
```

ANSI変換



```
SELECT A.COL1,B.COL2,C.COL4,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B,TABLEC C
WHERE A.COL1 = B.COL1 AND B.COL1 = C.COL1
AND A.COL3 = 'xxxxxx'
```

※これが意図的なものかは確認が必要。（潜在的なバグの可能性はある）

- 一部のみ外部結合の場合

```
SELECT A.COL1,B.COL2,C.COL4, D.COL1,E.COL2
FROM TABLEA A,TABLEB B,TABLEC C,TABLED D,TABLEE E
WHERE A.COL1 = B.COL1(+) AND B.COL1 = C.COL1(+)
AND D.COL1=E.COL1 AND A.COL1=D.COL1
AND A.COL3 = 'xxxxxx'
```

ANSI変換



```
SELECT A.COL1,B.COL2,C.COL4, D.COL1,E.COL2
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON A.COL1 =
B.COL1 LEFT OUTER JOIN TABLEC C ON B.COL1
=C.COL1,TABLED,TABLEE
WHERE D.COL1=E.COL1 AND A.COL1=D.COL1
AND A.COL3 = 'xxxxxx'
```

※(+)のところのみ直す

- 長いSQL（サブクエリー使用）はクエリー毎に区切って考えると楽（？）

```

SELECT .....
FROM ( SELECT .....
      FROM ( SELECT ..... FROM .....), (SELECT ..... FROM .....) ), (
      SELECT ..... FROM ....)
)

```

それぞれを個別に変換して当てはめる

- サブクエリーを使った複雑な、長いSQLでは、サブクエリー毎に分解し、ANSI変換を行う事を推奨
- サブクエリーでOracle特有書式が使われていない場合はそのサブクエリーは変換する必要はない
- テーブル名のシノニム（FROM TABLEA A）を使用している場合は、そのSQL内の全てのテーブル・サブクエリー結果にシノニムを付与してください。シノニム経由のカラム参照を使用しない場合でもシノニムを付与する必要があります。

```

SELECT .....
FROM ( SELECT .....
      FROM ( SELECT ..... FROM ..... ) Tab1, (SELECT ..... FROM .....), (
      SELECT ..... FROM ....)
)

```



```

SELECT .....
FROM ( SELECT .....
      FROM ( SELECT ..... FROM ..... ) Tab1, (SELECT ..... FROM .....),
      TAB2 ), (
      SELECT ..... FROM ....) TAB3
) TAB4

```

Step2 : 関数の変換

SQL内で関数が使用されている場合、その関数がSQL Anywhereで利用できるかが重要です。

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON
A.COL1 = B.COL1
WHERE A.COL3 = 'xxxxxx'
```

例えば、上記で使用されているDECODE関数はSQL Anywhereでは使用できませんので、SQL Anywhereが提供する同義のもので置き換えることになります。

- これは単純文字列置換で可能な場合もありますし、不可能な場合もあります。
- また、同名称・同定義のものも存在しますので変換不要な場合もあります。

これを判別するために、置換前にそのまま実行してしまうのも手です。SQL Anywhereに存在しない、あるいは定義が異なるのであればエラーになりますのでそれで判別が出来ます。

Step2 : 関数の変換 例

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON
A.COL1 = B.COL1
WHERE A.COL3 = 'xxxxxx'
```

Oracle:DECODE関数

DECODE (expr , search_and_result_list [, default])

expr NULL を設定可能な式（数値式、文字列式、日付式、etc）
search_and_result_list 検索式と結果式リスト [search , result]
default 検索式に一致しない場合の結果式 default NULL

return [第3引数（search_and_result_list の最初の result 式） のデータ型]

式 expr が search_and_result_list の第1番目の要素の search1 と同値なら result1 を戻す。search2 と同値なら result2 を戻す。これを 検索式と結果式リスト [search , result] のある分だけ繰り返す。最後まで一致する要素が見つからない場合にはオプション指定の 式 default の値を戻す。

上記例では “B.COL3がYの場合は1、そうでなければ0を返す” という動きとなる。

SQL AnywhereではDECODEという関数は存在しません。類似の動きをするものとして“CASE式”があります。これを利用して変換します。

```
CASE expression-1 WHEN expression-2 THEN expression-3, ...  
[ ELSE expression-4 ]  
{ END | END CASE }
```

SQL Anywhere :CASE式

CASE 句に続く式が WHEN 句に続く式と等しい場合、THEN 文の後の式に復帰します。それ以外の場合、ELSE 文があればそれに続く式に復帰します。
ELSE 句が存在しなく、*expression-1* が *expression-2*...*expression-n* のいずれの値とも一致しない場合、CASE 式は NULL を返します。



“B.COL3がYの場合は1、そうでなければ0を返す。” は実現可能

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON
A.COL1 = B.COL1
WHERE A.COL3 = 'xxxxxx'
```



CASE式を用いて同義変換

```
SELECT A.COL1,B.COL2,CASE B.COL3 WHEN 'Y'
THEN 1 ELSE 0 END
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON
A.COL1 = B.COL1
WHERE A.COL3 = 'xxxxxx'
```

※CASE式には変形でCASE WHEN <式> THEN ... や NULLとの比較に特化した ISNULL もあります。

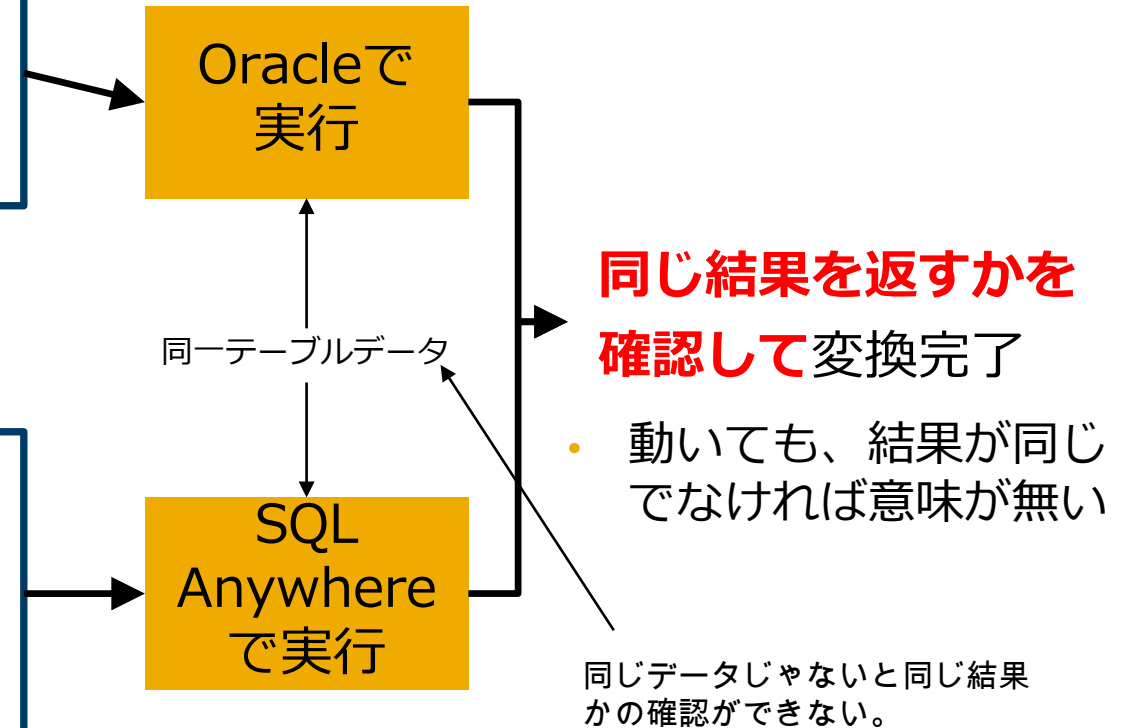
最後に：実行して確認してください！

元のOracle用SQL

```
SELECT A.COL1,B.COL2,DECODE(B.COL3,'Y',1,0)
FROM TABLEA A,TABLEB B
WHERE A.COL1 = B.COL1(+)
AND A.COL3 = 'xxxxxx'
```

SQL Anywhere用に変換したSQL

```
SELECT A.COL1,B.COL2,CASE B.COL3 WHEN NULL
THEN 0 ELSE B.COL3 END
FROM TABLEA A LEFT OUTER JOIN TABLEB B ON
A.COL1 = B.COL1
WHERE A.COL3 = 'xxxxxx'
```



そして変換したSQLをPowerBuilderのコードに貼り付ければ完了です。

1文の変換にどのくらいかかる？

これまでのお客様の御意見では・・・

- **勿論文の長さや複雑さ、あるいは担当者のスキルや慣れに依存するが、1文あたり10分から15分もあれば十分**
 - また、アプリケーション内では“似たようなSQL”が複数登場する場合が多く、一つを直せば他にも適用出来るパターンが存在するので、トータルで見るともっと短くなる可能性がある
 - 同一構造のSQLで対象テーブル違い、WHERE句のみ違う
 - 出力カラムが若干違う 等
- **アプリを開発するのではなく、「移行」なのでメンテ業務に近い**
 - Oracleだってバージョン間でうまく動かなくなるコマンドは存在するので、それを確認し、直すのと同じ
- **重要なのは「答え合わせ」が出来る環境を用意すること**
 - 同一データのOracleとSQL Anywhereデータベース環境を用意する。
 - 実行して同じ結果かを確認できる環境を用意することが重要

まとめ SQL Anywhere とは？

SAP SQL Anywhere は、標準SQLに準拠し、スタンドアロンから、クライアント・サーバー型またはWebのシステムまで対応が可能な汎用RDBMS「SQL Anywhere」と、異種のRDBMS間のデータ同期およびデータ交換のテクノロジー「Mobile Link」などを含むスイート製品です。

「SQL Anywhere」は、使いやすく、**自動管理、自動チューニングの機能**を豊富に揃えているため、短期間のアプリケーション開発、手離れ良い運用を実現します。

自己学習機能（クエリ・オプティマイザ）、データベースエリアの自動拡張、データベース自動起動/停止、タスクスケジューリングとイベント処理、自動リカバリ機能、キャッシュサイズの自動拡張・縮小、移動中DBの断片化解消機能、クエリ間とクエリ内の2種類の並列処理、スナップショットアイソレーション、自動マルチプログラミングレベルチューニング、カラム統計の自己回復、その他複数の自動管理機能を装備

データベース管理者を置けない遠隔地のサーバーでも問題なく運用でき、またパッケージに含まれる「Mobile Link」を使うことで、ネットワークが不安定な遠隔地のSQL AnywhereデータベースサーバーとASE, Oracle, SQL Server, DB2, HANA など本社データベースサーバーとを随時データ同期させ、業務を止めないAlways Available なシステムを実現できます。

「SQL Anywhere」データベースのサブセットであるより小型の「Ultra Light」データベースを利用すれば、オフライン利用が可能なモバイルアプリケーションの開発も可能です。

付録 : SQL Anywhere 関連情報



技術書

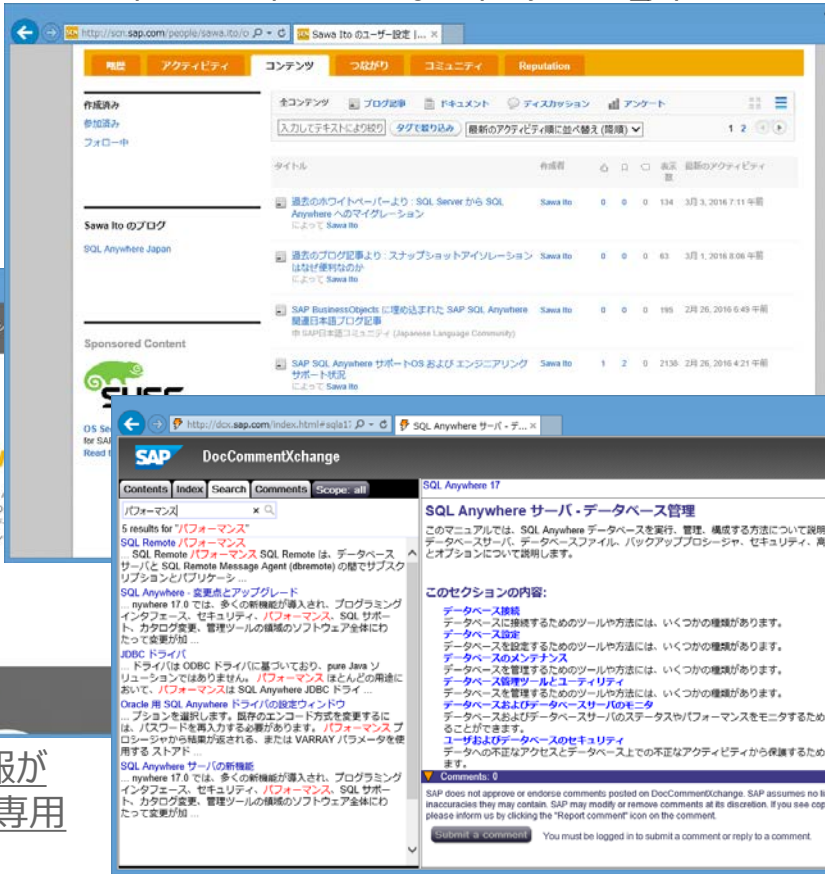
- 【目次】
- 第1章 SAP SQL Anywhereとは
 - 第2章 SQL Anywhere の使い方
 - 第3章 SQL Anywhereの内部動作
 - 第4章 パフォーマンスチューニング
 - 第5章 バックアップとリカバリ
 - 第6章 高可用性構成とスケールアウト構成
 - 第7章 Programming API
 - 第8章 Mobile Link
 - 第9章 Ultra Light
 - 第10章 SAP SQL Anywhereのユースケースと事例



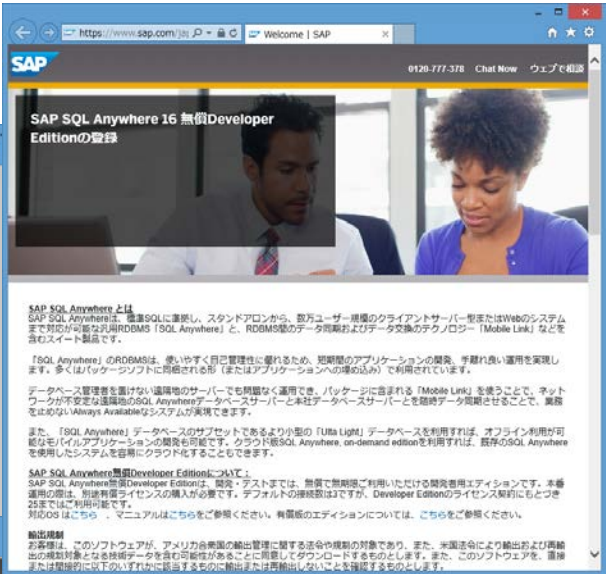
15年以上もの間蓄積された技術情報が
掲載された日本語SQL Anywhere専用
サイト

<http://www.sqlanywhere.jp/>

SAP コミュニティーネットワーク上の
豊富な日本語技術情報と日本語によるQ&A
http://scn.sap.com/blogs/sqlanywhere_japan



使いやすい日本語オンラインマニュアル
<http://dcx.sap.com/index.html>



気軽に試せる無期限無償
Developer Edition
<http://www.sqlanywhere.jp/dl/>



ありがとうございました

SAPジャパン株式会社
ソリューション統括本部
デジタル・ビジネス・プラットフォーム部
シニアソリューションスペシャリスト
境 直人 (Naoto SAKAI)
[E-mail: naoto.sakai@sap.com](mailto:naoto.sakai@sap.com)