ORACLE DATABASE 12c

Oracleホワイト・ペーパー
2013年6月

Oracle Database 12cによる アプリケーションおよびデータベースの移行

ORACLE

免責事項

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント（確約）するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクルの製品に関して記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

はじめに

アプリケーションやデータのデータベース間の移行は、多くの場合、リスクが高く、費用と時間のかかるプロセスです。ただし、オラクルの提供する製品を使用することにより、サード・パーティ・データベースからOracleプラットフォームへの移行に伴う時間、リスク、および費用に関する問題を削減できます。

Oracle Database 12cでは、サード・パーティ・データベースからOracleプラットフォームへの移行に必要なコストと時間を削減するために設計された重要な新機能が導入されています。これらの機能には、強化されたSQL Developer、強化されたSQL Developer Migration Workbench、自動増分ID列、暗黙的結果セット、32K VARCHAR、SQL変換フレームワーク、MySQLアプリケーション向けドライバ、FETCH FIRST行が含まれます。このホワイト・ペーパーでは、移行に役立つ新しいデータベース機能について説明します。

Oracle SQL Developer

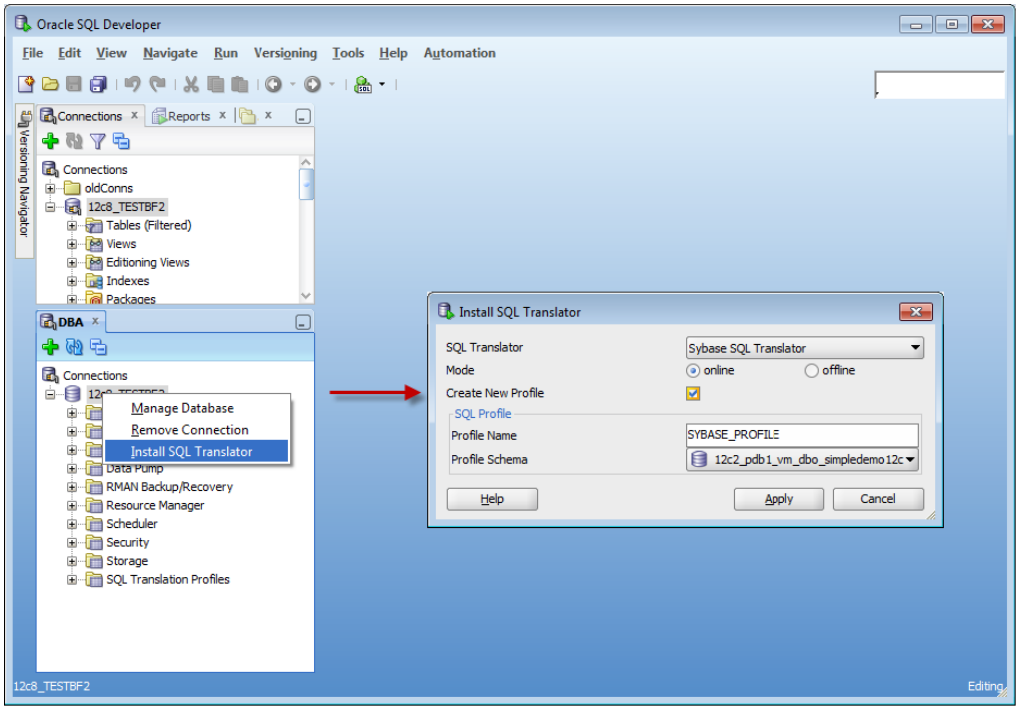
はじめに

Oracle Database 12cにはOracle SQL Developerが同梱されています。SQL Developerは、生産性の向上とデータベース開発作業の簡素化を実現する、グラフィカルなインタフェースを備えたツールです。SQL Developerを使用すると、データベース・オブジェクトの参照、作成、および変更、SQL文の実行、PL/SQLの編集とデバッグ、広範な事前定義済みレポート・リストへのアクセス、ユーザー固有のレポートの作成などが可能になります。

Oracle SQL Developerは、Oracleデータベース向けのおもなサード・パーティ・データベースの移行プラットフォームでもあります。Oracle SQL Developerには、Microsoft SQL Server、Sybase、MySQL、Microsoft Access、IBM DB2 LUW、およびTeradataをOracleデータベースに移行するための統合移行ツールが用意されています。

SQL Developerを使用すると、ユーザーはサード・パーティ・データベースへの接続を作成して、オブジェクトおよびデータを表示できます。接続を作成したら、SQL Developerのユーティリティを使用して、サード・パーティ・データベースをOracleデータベースに移行します。移行するデータベースに応じて、表、トリガー、ストアド・プロシージャ、および他のすべての関連オブジェクトが自動的にOracleデータベースに変換されます。ターゲットのOracleデータベースが生成された時点で、SQL Developerのサポートにより、サード・パーティ・データベースからターゲットのOracleデータベースにデータが移行されます。

Oracleデータベース移行のターゲット・データベースがバージョン12cの場合、Oracle SQL Developerは次に説明するデータベース12cの新機能を使用してオブジェクトおよびストアド・プロシージャを自動的に移行します。コードまたはオブジェクトをバージョン12g以降のOracleデータベースへ移行する以外に、バージョン11g以前のOracleデータベースへ移行する方法の例についても説明します。



Oracle Database 12cのアプリケーション移行のための拡張機能

はじめに

以下の新機能により、顧客がOracleデータベースへの移行を計画する際に直面する課題に直接対処できます。このホワイト・ペーパーでは、以下のそれぞれの機能について例を挙げて簡単に説明します。

- ID列
- 32k Varchar2s
- FETCH FIRST行（SQL）
- 暗黙カーソル
- Oracle Multitenant
- SQL変換フレームワーク
- MySQLアプリケーション向けドライバ

これらの各機能により、アプリケーションをOracle Databaseへ移行するために必要な作業量が大幅に減少し、時間と費用を節約できます。

識別列

主キー制約は、表のレコードを一意に識別するのに役立つ表の列を定義します。一般的なプログラミング手法では、表の行が生成されて挿入される際に、値を自動的に生成して割り当てます。以前のOracle Databaseのバージョンでは、これはシーケンスおよびトリガーを作成することによって頻繁に実行されていました。シーケンスが生成される値を定義し、トリガーが挿入を起動して、シーケンスの値を表にフィードします。

別のサード・パーティのリレーショナル・データベース管理システムでは、ID列を使用してこれを実行することもできます。これにより、シーケンス・ロジックを表の定義に直接埋め込むことができるため、表のレコードの主キー値の生成および移入を処理するシーケンスおよびトリガーを作成する必要がなくなります。

このことは、Oracle Databaseへ移行を行う顧客にとっては、大幅なコストの節約になります。識別列を使用する各表に2つのデータベース・オブジェクトを追加で作成する代わりに、表自体で定義を行えるようになります。また、管理およびサポートするデータベース・オブジェクトが削減されるため、将来的な保守コストが減少します。

オブジェクトの削減、コードの削減、そして作業量の減少のすべてが、Oracle SQL Developerを使用してOracle Database 12cに移行する際に自動的に実施されます。

32KのVARCHAR2

導入以来、VARCHAR2データ型（VARCHAR2、NVARCHAR2、RAWなど）のサイズは最大4,000バイトで、これはシングルバイト・キャラクタ・セットでは4,000文字になります。

このサイズを超える表の列定義は、CLOBまたはBLOBとして移行されることになります。これは、他のサード・パーティ・データベースがはるかに多くのネイティブ・データ型の文字列をサポートしていたため、非Oracleデータベース環境から移行を行う多数の顧客にとって課題でした。Oracle Database 12cの導入により、VARCHAR2、NVARCHAR2、およびRAWでは、最大32,768バイトがサポートされるようになりました。

CLOBでは、最適化および柔軟性に関する問題が発生することがあります。VARCHAR2のサイズが拡張されたということは、ほとんどの場合、大きな文字列を含む表の列定義をCLOBへ切り替える必要なしに移行を継続できることを意味します。さらに、CLOBやBLOBとは異なり、列の上に索引を構築することもできます。

デフォルトのOracle Database 12cパラメータは、32Kという新しいサイズのVARCHAR2を使用できるように更新する必要があります。

これらのデータ型のサイズ制限を増やせるようにするには、以下のデータベース・パラメータが必要です。

MAX_SQL_STRING_SIZEは、SQLの拡張されたデータ型の最大サイズを制御します。

LEGACYは、Oracle 12cより前のデータベースで使用されるデータ型の長さの制限を表します。

EXTENDEDは、Oracle Database 12cでの32,767バイトの制限を表します。

12.0.0.0以降では、COMPATIBLE初期化パラメータを設定し、MAX_SQL_STRING_SIZE = EXTENDEDを設定する必要があります。

FETCH FIRST行

データを整理してから行出力を制限する問合せは広範に使用され、多くの場合、上位N問合せと呼ばれています。Oracle Database 12cより前のリリースでは、開発者は、擬似列'ROWNUM'を使用してこのANSI SQL機能を実装し、問合せで返される行の数を制限しようとするでしょう。

Oracle Database 12c Release 1では、結果セットで返される行の数を制限するrow_limiting_clauseを使用できるようにSQL SELECT構文が拡張されています。row_limiting_clauseにより、構文が理解しやすくなり、表現力が増します。返される行の数が制限されることは、レポート作成、分析、データ表示、その他のタスクにとって有益です。

FETCH_FIRSTキーワードを用いて、返される行の数または行のパーセンテージを指定できます。OFFSETキーワードを使用すると、返される行が完全な結果セットの最初の行のあとの行で始まるように指定できます。WITH TIESキーワードには、行が制限された結果セットの最後の行と同じ並び方のキーを含む行が含まれます（問合せでORDER BYを指定する必要があります）。

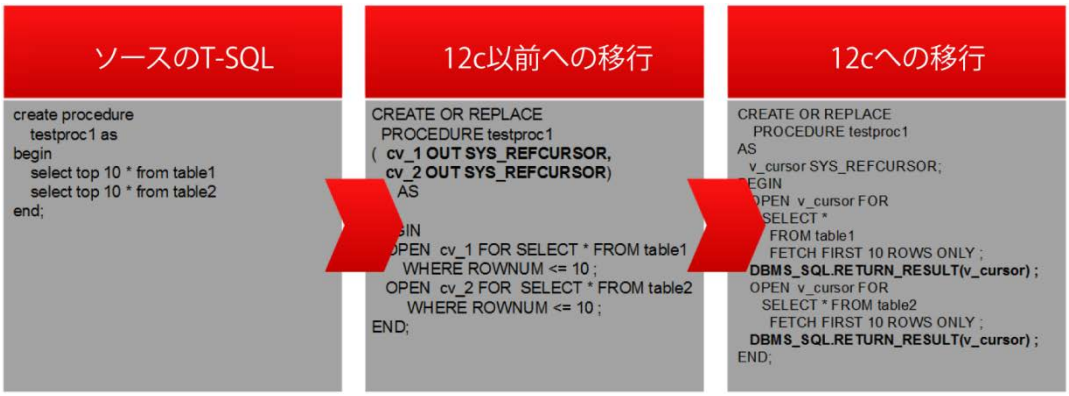
row_limiting_clauseは、互換性を拡張して移行を容易にする目的でANSI SQL国際規格に従っています。

新しいFETCH FIRST SQLは強力で読取りが簡単

暗黙カーソル

Microsoft SQL ServerデータベースおよびSAPのSybase ASEデータベースの拡張SQL言語であるT-SQLでは、一般的なプログラミングはSQL文を直接ストアド・プロシージャに書き込むことで行われます。これらのストアド・プロシージャを呼び出すことにより、1つまたは複数の問合せに対する結果セットを、呼出しを行っているユーザーまたはプログラムが即座に使用できるようになります。

Oracle Database 12cより前のリリースでは、これらのストアド・プロシージャをOracle Database PL/SQLの同等のプロシージャに移行するには、OUTパラメータまたはRETURNパラメータとして1つまたは複数のSYS_REFCURSORを含めるようにプロシージャ・ヘッダーを変更し、続けてRef Cursors経由で結果セットを取得する必要があります。



Oracle Database 12cでは、ストアド・プロシージャがDBMS_SQLパッケージのRETURN_RESULT()関数を用いることにより、呼出しを行っているユーザーまたはプログラムが問合せ結果を使用できるようにします。

JDBCの例

Oracle JDBCの新しいメソッド

getMoreResults()またはgetMoreResults(int)は、他にも使用できる結果が結果セットにあるかどうかをチェックします。

intパラメータには、以下の値のうちの1つが含まれます。

KEEP_CURRENT_RESULT、CLOSE_ALL_RESULTS、CLOSE_CURRENT_RESULT

getResultSet()は、それぞれの暗黙的結果を繰り返し取得します。

サード・パーティ・データベースから移行された以下のJavaコードは、変更なしでOracle Databaseと一緒に稼働します。

```
CallableStatement cstmt = null;
ResultSet rs = null;
cstmt = conn.prepareCall("{call testproc1()}");
cstmt.execute();
boolean resultsAvailable = cstmt.getMoreResults();
```

Oracle Multitenant

Oracle database 12cでは、Oracle Multitenantによってアプリケーションのマルチテナント環境がデータベース・レベルで実現可能になりました。単一のOracle Database 12c Container Database（CDB）で、1つまたは複数のプラガブル・データベース（PDB）を提供できます。移行される各データベースの'コンテナ'としてスキーマを使用して複数のサード・パーティ・データベースを1つのOracle Databaseに移行するのではなく、Oracle Database 12cでは個々のPDBを使用できるようになりました。

SQL変換フレームワーク

はじめに

データベース・オブジェクトおよびデータの移行は大変な作業ですが、データベース・アプリケーションの移行も同様にクリティカルで時間がかかります。リレーショナル・データベース管理システムにはそれぞれ、SQL標準の独自実装が搭載されています。Sybase ASEで実行可能なSQLはOracleデータベースでは実行できません。アプリケーション内に存在するカスタムSQLの量によって、データベースやそのアプリケーションを完全に移行するために必要となる時間が大体決まります。

データベース・アプリケーションの移行は、Oracle SQL Developerとそのアプリケーション・スキャナ・スクリプトによって支援されます。SQL Developerは、Oracle Databaseに対して実行する前に変換が必要なSQL文を解析してドキュメント化できます。実際の変換を行うタスクはエンドユーザーが担当するか、またはSQL DeveloperのSQL Translation Scratch Editorを使用して一度ずつ試行できます。

非定型の変換エンジンであるSQL Translation Scratch Editorを使用すると、ユーザーはサード・パーティ・データベースに接続し、SQL文を実行してそれらをOracle向けに変換し、Oracle Databaseで文を再度実行して結果を比較できます。そして、開発者はアプリケーションを手動で更新して修正済みのコードを実行できます。この作業は静的SQLには適していますが、動的に生成されたSQL文に対するソリューションにはなりません。

このエラーが発生しやすく時間のかかるプロセスが、Oracle Database 12cのSQL変換フレームワークの導入によって、大幅に改善されています。このフレームワークでは、Oracle SQL Developerのトランスレータを、Javaストアド・クラスおよびストアド・プロシージャのコレクションとして、データベース内に直接ロードできます。Oracle Database 12cでは、Sybase ASEおよびSQL Server向けのトランスレータが使用できます。

SQL Developerからデータベースにインストールされたトランスレータは、セッション・レベルまたはサービス・レベルでアクティブ化できます。データベースに送信されたSQL文は、Oracle以外のSQLとして解析され、変換され、実行されます。これらの一連の変換方法は、SQL変換プロファイルに保管されます。プロファイルの内容については、変換方法が正確であることを確認するために、移行チームによるレビューや変更、承認が可能です。変換方法はトランスポータブルであるため、移行対象のアプリケーションごとにプロファイルを作成してから、複数のデータベース間で送信できます。

SQL変換フレームワークのワークフロー

1. フレームワークがSQL呼出しを受信
2. SQL変換ディクショナリ（プロファイル）で検索を実行
3. 結果が得られない場合は、文にフィンガープリンティングを実行してそれをディクショナリに追加
4. 提供された値を用いてテンプレートを処理

例

- フレームワークによる受信

```
SELECT TOP 2 * FROM T1
```

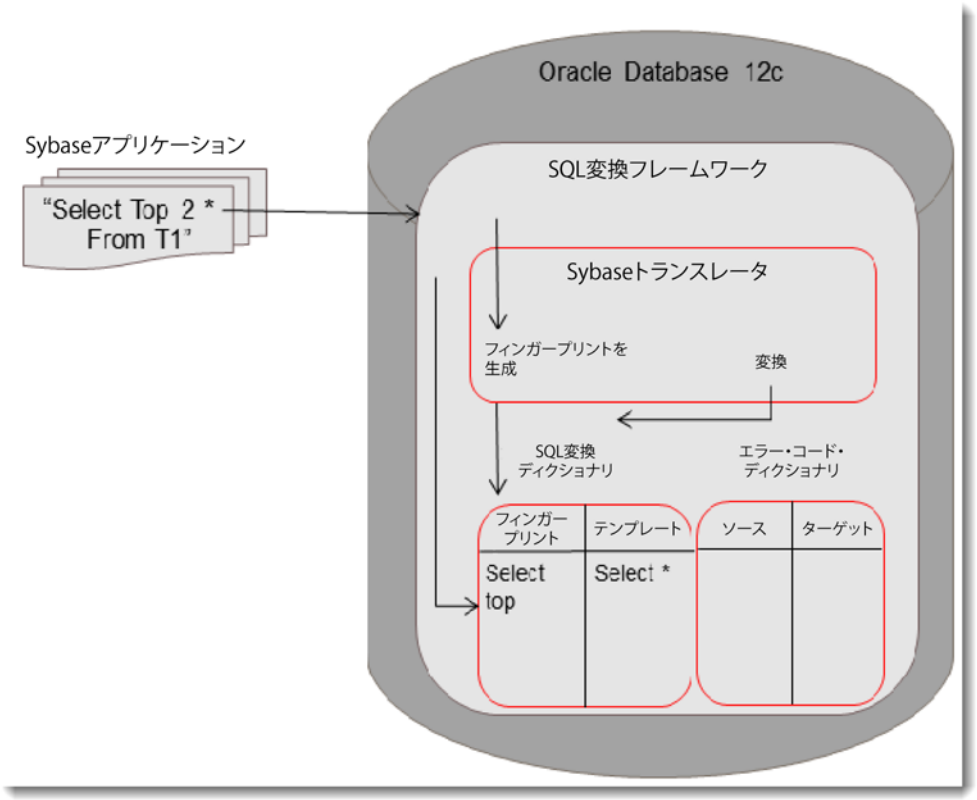
- SQL変換ディクショナリで変換の静的検索を実行
- 検索結果なし：フィンガープリントを生成
- Select Top <ora:literal type=integer order=1> * From T1

  注：リテラルは、'select 1; select 2 select 3;'が1つの文として処理されるように変換でマッピングされ、リテラル（1、2、または3）は、フィンガープリントで<ora:literal type=integer order=1>）として保存されます。
- SQL変換ディクショナリでフィンガープリントを検索

  検索結果あり：フィンガープリントを取得

  Select * From T1 FETCH FIRST <ora:literal type=integer order=1> ROWS ONLY
- 獲得した値を用いてテンプレートを処理

  Select * From T1 FETCH FIRST 2 ROWS ONLY
- 変換されたSQLをフレームワークへ戻す
- 必要な場合はSQL変換フレームワークがバインドに対処



SQL変換フレームワークのダイアグラム：SybaseアプリケーションはOracleに接続して稼働し、Oracle向けにその場で文を変換および実行させます。

結論

Oracle Database 12cは、顧客によるITコストの削減を支援し、Oracle ExadataやOracle Database Applianceなどのエンジニアド・システムやデータベース・クラウド上への統合を可能にすることによって、より高い質のサービスを提供します。Oracle Database 12cは、エンタープライズ・アプリケーション、データウェアハウス、ビッグ・データ分析を含むあらゆるタイプのデータベース・ワークロードにとって、迅速で信頼性が高く、セキュアで管理しやすいことが証明されています。

多くの場合、データベースおよびデータベース駆動型アプリケーションをOracle Databaseへ移行するには、既存のOracleストラクチャ、データ型、独自のSQLおよび手続き型言語（PLSQL）と対応させるために非Oracleのテクノロジーを実装する必要があるため、膨大なアプリケーションおよびデータ・モデルの更新が必要です。Oracle Database 12cに含まれる多数の新機能により、元来はOracle Database向けに開発されていないアプリケーションに対応させるためのデータベースおよびアプリケーションの変更を最小限にできます。

Oracle Database 12cによるアプリケーション開発
2013年6月
著者：Jeff Smith
共著者：Ashley Chen, David Gambino,
Kuassi Mensah

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

海外からのお問い合わせ窓口：
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200

oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together