

Oracle Database 12cによるアプリケーション開発

ORACLEホワイト・ペーパー | 2014年9月





免責事項

下記事項は、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。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する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。

目次

免責事項	2
はじめに	3
データベース・アクセス・インフラストラクチャとAPI (Java、C、C++) ..	4
概要	4
Oracle Multitenant	4
新しいSQLデータ型	4
32KのVARCHAR、NVARCHAR、およびRAW	4
自動増分 (IDENTITY) 列	5
非表示列	5
パラメータとしてのPL/SQLパッケージ型およびブール型	5
暗黙的結果	5
パフォーマンスとスケーラビリティ	5
SQL計画管理 (SPM)	5
データベース常駐接続プールによるスケーラビリティの実現	5
グローバル・データ・サービス - 実行時ロードバランシング	6
大容量のネットワーク・バッファと高度なネットワーク圧縮	6
アプリケーションの可用性と信頼性の最大化	6
共通HAイベントと通知	6
新しい概念	6
トランザクション・ガード	7
アプリケーション継続性	7
グローバル・データ・サービス	8
使いやすさ	8
配列DMLの反復当たりの行数	8
クライアント側デプロイメント・ディスクリプタと自動チューニング ..	9
データベース操作の監視と追跡	9
インテリジェントな接続と無効なクライアント接続の迅速な検出	9
セキュリティ	9
データベースのJavaに対するデフォルトのJavaセキュリティのカスタ マイズ	9
JDBCのセキュリティ強化	10
Oracle Advanced Securityオプション	10
データベースでのJavaによるRuntime.execの起動	10
Java標準	10
JDBC 4.1	10

JDK 1.7および複数のJDK.....	10
JNDI.....	10
ロギング.....	11
アプリケーションの移行.....	11
SQL変換フレームワーク.....	11
MySQLアプリケーション向けドライバ.....	12
その他のリソース.....	12
Oracle Database Cloud Service.....	12
SQLとPL/SQL.....	13
SQLとPL/SQLの重要な新機能.....	13
SQL CROSS APPLY、OUTERAPPLYおよびLATERAL.....	13
OracleネイティブのLEFT OUTER JOIN構文の拡張.....	13
有効時間をサポートする表の定義.....	14
有効時間のテンポラル・フラッシュバック問合せ.....	14
SQL WITH句でのPL/SQLファンクションの定義.....	14
PL/SQLからSQLへのインタフェースで利用できるPL/SQL固有のデータ型.....	15
PL/SQLユニットから参照できるデータベース・オブジェクトのホワイト・リストを制限するメカニズム.....	15
起動者権限ファンクションによる結果のキャッシング.....	15
DIRECTORYタイプのオブジェクトによるLIBRARYタイプのオブジェクトの定義.....	15
新しいPL/SQLパッケージUTL_CALL_STACK.....	15
新しいPL/SQLサブプログラムDBMS_UTILITY.EXPAND_SQL_TEXT.....	16
DBMS_SQL.PARSE()の新しいSCHEMAパラメータ.....	16
Oracle SQL Developer.....	16
Oracle SQL Developerのコア領域.....	16
利点.....	17
Oracle Database 12cのサポート.....	17
Oracle Application Express.....	18
Oracle Application Express 4.2のおもな機能.....	19
インタラクティブ・レポート.....	19
スプレッドシートを使用した対話型操作.....	19
グラフ機能.....	19
マスター・ディテール・フォームと表形式フォーム.....	20
動的アクション.....	20
プラグイン.....	20
モバイル・アプリケーション.....	20



パッケージ・アプリケーション	20
Webシート	21
Oracle REST Data Services	21
Release 2.0の新機能	21
.NET	22
.NETのおもな新機能	23
ODP.NETのマネージド・ドライバ	23
Oracle Multitenant	23
スキーマ比較ツール	24
トランザクション・ガード – 高可用性	24
グローバル・データ・サービス – 高い可用性とパフォーマンス	24
Entity Frameworkおよび統合言語クエリー – SQL APPLY	24
容易な開発	24
グローバリゼーション・サポート	25
グローバリゼーション・サポートの新機能Unicode 6.1のサポート	25
Unicode照合アルゴリズム（UCA）への適合	25
新しいロケールの対象範囲	25
Database Migration Assistant for Unicode（DMU）	26
JSON	26
結論	29



はじめに

アプリケーションの効率性と機能は、ベースとなるデータベースの機能に大きく左右されます。Oracle Database 12cには、アプリケーションの開発と導入を支援する重要な新機能が組み込まれています。Oracle Database 12cは、クラウド向けに設計された初めてのリレーショナル・データベースです。Oracle MultitenantはOracle Database 12cの新しいオプションで、極めてセキュアなマルチテナント・データベース・アーキテクチャを提供します。Oracle Multitenantを使用すると、アプリケーション・レベルでマルチテナンシーをコーディングする必要がなくなるため、クラウド向けの開発が容易になります。

Oracle Database 12cに搭載された業界最先端のデータベース機能とパフォーマンスは、アプリケーション開発に特に多用されるOracle APEX、PL/SQL、SQL、C、C++、Java、.NET、PHP、Perl、Python、Rubyといった幅広いテクノロジーで活用できます。Oracle Database 12cは、市場トップクラスのセキュリティと高いパフォーマンス、可用性、スケーラビリティをオンライン・トランザクション処理やデータウェアハウス、ビッグ・データにもたらす、単一の統合型データベース・プラットフォームです。

Oracle Database 12cには、使いやすさ、セキュリティ、スケーラビリティ、パフォーマンスに重点を置いた新機能が追加されています。他社製データベースからOracle Database 12cへの移行をこれまで以上に簡単にする新機能も数多く追加されています。このホワイト・ペーパーでは、アプリケーション開発者が関心を持つおもな新機能の概要を説明します。

データベース・アクセス・インフラストラクチャとAPI (Java、C、C++)

概要

この項では、Oracle Databaseのアクセス・インフラストラクチャとAPIの新機能について説明します。どの機能も、JavaアプリケーションやC/C++アプリケーションのパフォーマンス、スケーラビリティ、可用性、セキュリティ、使いやすさに重点が置かれています。Oracle JDBCまたはOracleデータベース・アダプタ (PHP-OCI8、Ruby-OCI8、ActiveRecord対応のOracle拡張アダプタ、Python cx_Oracle、Perl DBD::Oracle) を使用するPHP、Ruby/JRuby、Python/Jython、Perlなどのスクリプト言語も、それぞれのコードの維持管理を担当するコミュニティが統合作業に着手すれば、これらの機能の利点を享受できるでしょう。

Oracle Multitenant

Oracle Multitenantは、“トランスポートابل表領域”の概念をデータベースにまで広げます。このアーキテクチャは、コンテナ・データベース (CDB) と呼ばれるルート・インフラストラクチャと一連のプラガブル・データベース (PDB) で構成されます。CDBには、Oracleから提供されるメタデータのみが含まれます。PDBは本格的なデータベースで、ここにはユーザー・データとアプリケーション・データ、メタデータが含まれます。PDBはCDBに対してプラグイン/アンプラグができます。

プラガブル・データベースの利点は次のとおりです。

- 新しいデータベースや既存のデータベースのコピーを迅速にプロビジョニングできる。
- アンプラグ/プラグイン操作で、既存のデータベースを新しいプラットフォームへ迅速に再デプロイできる。
- PDBをアンプラグし、新しいバージョンの別のコンテナ・データベース (CDB) にプラグインするという方法で、パッチの適用やアップグレードをすばやくできる。
- モノリシックな従来型データベースよりも多くのPDBを、1台のマシンで実行できる。
- アプリケーション管理者の職責 (PDB) とオラクル提供システムの管理者の職責 (CDB) が分離される。

JDBC、SQLJ、OCI、OC CIの各ドライバはプラガブル・データベースに対応しています。接続文字列にサービス名 (Net Servicesの用語) が使用されていれば、JavaアプリケーションやC/C++アプリケーションからのPDBの見え方は従来のOracleデータベースとまったく同じで、操作もまったく同じです。

新しいSQLデータ型

JavaアプリケーションおよびC/C++アプリケーションで利用できるようになった新しいデータ型には、32KのVARCHAR、NVARCHAR、およびRAWデータ型、非表示列、暗黙的結果、自動増分 (IDENTITY) 列、パラメータとしてのPL/SQLパッケージ型およびブール型などがあります。

32KのVARCHAR、NVARCHAR、およびRAW

VARCHAR2、NVARCHAR2およびRAWの各データ型の最大サイズが、4,000バイトから32,767バイトに増加しました。JDBCとOCI/OC CIをそれぞれ使用するJavaアプリケーションとC/C++アプリケーションは、32Kよりサイズの小さいデータをラージ・オブジェクト (LOB) に切り替える必要がなくなります。また、これらのデータ型を使用して宣言した列に索引を組み込むこともできます。これらのデータ型の割当てサイズが増加したことで、ラージ・オブジェクト (LOB) に切り替えなくても格納できる情報量が増加しました。これが特に役立つのは、短いテキストのデータ型と、それらのデータ型の列に索引を構築する機能です。

自動増分（IDENTITY）列

米国規格協会（ANSI）が規定するIDENTITYキーワードが、表の列でサポートされるようになりました。

これで、自動的に増分する列を標準的な方法で宣言できるようになるため、アプリケーション開発が簡単になり、DDLをより簡単にOracleへ移行できるようになります。

非表示列

表に作成または追加した新しい列を、後からINVISIBLEキーワードを使って非表示にすることができるようになりました。非表示列は、“SELECT * FROM table”や“DESCRIBE table”などによる表全体へのアクセスでは表示されませんが、SELECTリストに明示的に指定すれば表示できます。

パラメータとしてのPL/SQLパッケージ型およびブール型

この機能を使用すると、PL/SQLのパッケージ型とブール型をデータベース・クライアントAPI（OCIやJDBCなど）でネイティブに記述してバインドできます。そのため、JavaやCベースのアプリケーションでは、PL/SQLパッケージ型とブール型（Javaを除く）をパラメータとして使用することで、PL/SQLファンクションやPL/SQLプロシージャを簡単にバインドして実行できるようになりました。

クライアント側アプリケーションからPL/SQLファンクションやPL/SQLプロシージャを実行するときの複雑さは、この機能を使用することで軽減されます。

暗黙的結果

暗黙的結果がサポートされるようになったため、JavaアプリケーションやC/C++アプリケーションでは、PL/SQLプロシージャDBMS_SQL.RETURN_RESULTを使用してストアド・プロシージャ（PL/SQL、Java）の結果を直接取得できるようになり、Ref Cursorを使用する必要がなくなりました。この機能を使用すると、JavaアプリケーションやC/C++アプリケーションをより簡単にサード・パーティ製データベースからOracle Databaseへ移行できます。

パフォーマンスとスケーラビリティ

JavaアプリケーションやC/C++アプリケーションで活用できるパフォーマンスとスケーラビリティに関する機能には、SQL計画管理、JavaでのDRCPサポートとその他の拡張機能、非常に大容量のネットワーク・バッファ、高度なネットワーク圧縮、グローバル・データ・サービスを使用した遠隔地間の実行時ロードバランシングなどがあります。

SQL計画管理（SPM）

このリリースで行われたSQL計画管理の機能強化は次のとおりです。

- ・計画ベースラインの実行計画の保存。
- ・改良タスクを実行する自動ジョブの作成。別のドキュメントやWebページからこのテンプレートにテキストのコピー/貼付けを行う場合は、コンテンツを張り付けるときに「Home」タブをクリックして「Paste」→「Paste Special」→「Unformatted Text」の順に選択します。この手順でコンテンツを貼り付けると、このテンプレートで使用されるスタイルに変換され、元のファイルから取得された誤ったスタイルは保持されません。

計画ベースラインの作成がSQLJでサポートされるようになりました。計画ベースラインは、SQLJファイルを変換するときに生成できます。計画ベースラインの作成に必要なSQL文は.sqlファイルに生成されます。

データベース常駐接続プールによるスケーラビリティの実現

データベース常駐接続プール（DRCP）はRDBMS側にあるサーバー・プロセスのプールで、クライアント・アプリケーション、プログラミング言語および中間層の間で共有されるサーバー・プロセスがここに保持さ



れます。DRCPは、C、C++、PHP、PythonおよびPerl向けのOracle Database 11g OCIに導入されたもので、クライアント/中間層の接続をプールできないシングルスレッド・システム（PHP/Apacheなど）に必須の機能です。DRCPを使用するとデータベース・サーバー・プロセスの数とメモリ使用量が激減するため、マルチスレッド・システムだけでなく、同じデータベースにアクセスする中間層の数が数千にもなる大規模なシステム展開でも効果があります。

今回、JavaでもOracle JDBCドライバを介してDRCPが使用できるようになりました。クライアント/中間層での接続のチェックインとチェックアウトを追跡する場合は、Oracle Universal Connection Pool (UCP) など、Javaに対応したクライアント側接続プール（C/C++向けOCIセッション・プールに類似したもの）が必要です。

Javaアプリケーションでは、新しいJDBC接続プロパティである `oracle.jdbc.DRCP.name` と `oracle.jdbc.DRCP.purity` を使用して、DRCPプールを指定したり、1つのプールを複数のアプリケーション用にサブパーティション化したりできます。

グローバル・データ・サービス – 実行時ロードバランシング

遠隔地間の実行時ロードバランシングがUCPでサポートされるようになりました。

大容量のネットワーク・バッファと高度なネットワーク圧縮

Oracle Database 11g Release 2では、Oracle Net Servicesでサポートされるネットワーク・バッファ（セッション・データ・ユニット（SDU））のサイズが64Kまででしたが、このリリースでは最大2MBへと非常に大きくなりました。SDUは、RDBMS側のSQLNET.ORAと、クライアント/中間層のSQLNET.ORA、TNSNAMES.ORAまたは接続文字列で構成できます。

ネットワーク接続を介して転送されるOracle Netセッション・データ・ユニット（SDU）のサイズは、高度なネットワーク圧縮機能によって縮小されます。そのため、応答時間とネットワーク・スループットが向上し、帯域幅の消費量が減少します。高度なネットワーク圧縮機能は、一度構成すればクライアント（アプリケーション、中間層）とサーバー（RDBMS）には透過的に実行されます。

詳しくは、Oracle Net Servicesのポータルにある『Advanced Network Compression』ホワイト・ペーパーと『Oracle® Database Net Services管理者ガイド』を参照してください。

アプリケーションの可用性と信頼性の最大化

JavaアプリケーションとC/C++アプリケーションの可用性と信頼性を最大化するための新機能には、共通HAイベントと通知の統合、トランザクション・ガード、アプリケーション継続性、グローバル・データ・サービスを使用した遠隔地間の高速接続フェイルオーバーなどがあります。

共通HAイベントと通知

Oracle Real Application Clusters (Oracle RAC) およびOracle Data Guardは、NODE DOWN、INSTANCE UP/DOWN、SERVICE UP/DOWNなどのHAイベントを発行します。これらのイベントは、ONS (Java) またはAQ (C/C++) のいずれかを使用してサブスクライバ（ドライバ、アプリケーション）に転送および通知されます。Oracle Database 12cを使用する場合は、C/C++アプリケーションでもJavaと同じ通知メカニズム（つまり、ONS）が使用されるようになりました。

新しい概念

リカバリ可能なエラー：Oracle Database 12cでは、新しいエラー属性 `is_recoverable` が使用できるようになりました。この属性をチェックすることで、エラーがリカバリ可能なものかどうかをアプリケーション側で判断できます。これで、OCI/OCIを使用するC/C++アプリケーションもJava並みになり、エラー・コード（ORA-1033、ORA-1034、ORA-xxxなど）のリストを独自に管理する必要がなくなります。



データベース・リクエスト：アプリケーションから送信されるSQL、PL/SQL、ローカル・コール、リモート・プロシージャ・コールなどの作業ユニットには、通常COMMITが1つ含まれますが、1つもない場合や複数ある場合もあります。

論理トランザクションID (LTXID)：最後のCOMMITの結果を取得する新しい概念。

可変関数：コールされるたびに結果が変わる可能性のある、確定的でない関数。SYSDATE、SYSTIMESTAMP、SEQUENCES、SYS_GUIDなどがあります。

セッション状態一貫性モデル：非トランザクション・セッションの状態。動的または静的な状態を取り得ます。

トランザクション・ガード

トランザクション・ガードには、クライアント・アプリケーションとOracle Databaseとの間の通信が中断した後で、最後に実行されたCOMMITの結果を確実に判断するという課題を解決するための汎用ツールが用意されています。

トランザクション・ガードを使用しないと、トランザクションがすでに開始されていてコミットもすでに発行されていた場合は、クライアントに返送されるコミット・メッセージが永続的ではありません。クライアントは、トランザクションがコミットされたかどうか分からない状態になります。非トランザクション状態が正しくない場合、またはトランザクションがすでにコミットされている場合は、トランザクションを再送信できません。コミットされたかどうか、または完了したかどうか不明のままトランザクションを再送信すると、トランザクションが複数回適用され、誤った状態になりかねません。

トランザクション・ガードの利点は次のとおりです。

- コミット結果を保持する最初のRDBMS。
- すべてのトランザクションの結果がわかる。
- トランザクションを最大1回実行するツール。
- トランザクション・ガードAPIはJDBC-Thin、OCI、OCCL、およびODP.NETで使用可能。

詳しい説明とサンプル・コードは、Oracle Technology Network (OTN) にあるホワイト・ペーパー、『Maximum Application Availability』と『Transaction Guard』を参照してください。

アプリケーション継続性

Oracle RACとActive Data Guardを使用すると、何も設定しなくてもアプリケーション継続性が機能し、“ほとんど負荷のない”アプリケーション可用性が得られます。アプリケーション継続性により、次の処理が透過的に実行されます。

1. 正常な実行時に、“データベース・リクエスト”の境界内にある処理中の作業を取得します。
2. RDBMSインスタンスまたはサイトに障害が発生した場合は、リカバリ可能なエラーであれば、内部的にトランザクション・ガードを使用して、正常なRDBMSインスタンス (Oracle RAC) かディザスタ・リカバリ・サイト (ADG) に再接続します。
3. 取得済みの処理中の作業を再実行します。

アプリケーション継続性の利点は次のとおりです。

- ハードウェア、ソフトウェア、ネットワーク、ストレージのエラーや停止が秘匿される。
- 障害発生時の応答時間の遅延がわずかになるなど、エンドユーザー・エクスペリエンスが向上する。



Oracle Database 12cでは、何も設定しなくても、JDBC-ThinのみおよびUCPを介してJava対応のアプリケーション継続性を使用できます。サード・パーティ製ドライバまたは接続プールでアプリケーション継続性を有効化するには、beginRequest/endRequestを使用して作業ユニット（すなわち“データベース・リクエスト”）の境界を明示的に設定します。

Java対応のアプリケーション継続性を有効にするには、非推奨のOracle.SQL具象クラス（BLOB、CLOB、BFILE、OPAQUE、ARRAY、STRUCT、ORADATA）ではなく、標準のJDBCインタフェースを使用する必要があります。詳しくは、My Oracle Support Note 1364193.1を参照してください。

アプリケーション継続性を有効にしても負荷はほとんどかかりませんが、Oracleホワイト・ペーパー『アプリケーション継続性』と『Maximum Application Availability』では設計時の考慮事項と制限事項について詳しく説明しています。

グローバル・データ・サービス

Oracle Universal Connection Pool (UCP) は、遠隔地間的高速接続フェイルオーバーとアプリケーション継続性をサポートします。グローバル・データ・サービスについて詳しくは、Oracle Database 12c Active Data Guardに関するホワイト・ペーパーを参照してください。

使いやすさ

Oracle Database 12cには、アプリケーションを使いやすくする機能として、配列DMLの反復当たりの行数、クライアント側デプロイメント・ディスクリプタ、自動チューニング、データベース操作の監視と追跡、インテリジェントなクライアント接続、無効な接続の迅速な検出が搭載されています。

配列DMLの反復当たりの行数

このリリースでは、配列DML文（array INSERT、UPDATE、DELETE）の反復ごとに影響を受ける行数を、JDBCとOCI/OCIDをそれぞれ使用するJavaアプリケーションとC/C++アプリケーションで取得できるようになりました。

C/C++アプリケーションの場合は、OCIで次の属性がサポートされます。

- 新しいmode/パラメータ属性：OCIStmtExecute() コールのOCI_RETURN_ROW_COUNT_ARRAY
- 新しい文ハンドル属性：OCI_ATTR_ROW_COUNT_ARRAY

Javaアプリケーションの場合は、反復ごとに影響を受ける行数がOracle JDBCから返されるようになりました。次の文を実行すると、UPDATEを実行するたびに、更新される行数が表示されます。

...

```
int rcount[] = stmt.executeBatch();
```

クライアント側デプロイメント・ディスクリプタと自動チューニング

このリリースでは、Java EEの場合と同様に、OCI/OCIIを使用するC/C++アプリケーションの動作をデプロイメント・ディスクリプタoraaccess.xmlで制御できるようになりました。このデプロイメント・ディスクリプタを使用すると、ソース・コードを変更したりリコンパイルしたりしなくても、選択したOCI/OCIIパラメータを構成することで、C/C++アプリケーションの動作をデプロイメント時に変更することができます。デプロイメント時の設定は、プログラムによる設定より優先されます。

oraaccess.xmlを更新しても、すでに実行中のクライアントには影響しません。変更を有効にするには、クライアントを再起動する必要があります。oraaccess.xmlファイルはデフォルトの場所（UNIXの場合は\$ORACLE_HOME/network/admin、Microsoft Windowsの場合は%ORACLE_HOME%\NETWORK\ADMIN）か、通常のクライアント・インストールおよびインスタント・クライアント・インストールのTNS_ADMIN環境変数で指定したディレクトリから読み込まれます。

oraaccess.xmlデプロイメント・ディスクリプタを利用することで、クライアント結果キャッシュ、接続キャッシュ、ステートメント・キャッシュ、プリフェッチ・メモリサイズ、ONSの各領域について、負荷をかけず自動的にC/C++アプリケーションのパフォーマンスをOCIで管理できるようになりました。

また、新しいセッション・ハンドル属性OCI_ATTR_MAX_OPEN_CURSORSが追加されました。

データベース操作の監視と追跡

エンド・ツー・エンドの追跡ができるように、Oracle Databaseにはタグを格納するためのネームスペース（OCSID）が予約されています。格納されるタグは、MODULE、ACTION、クライアントID、ExecutionContextID（ECID）、MODULE、SEQUENCE_NUMBERおよび新たに追加されたDBOPです。これらのタグは、アクティブなデータベース接続やクライアント/サーバー接続がなくてもスレッドと関連付けることができます。アプリケーションからデータベース・コールを発行すると、アプリケーションの接続に便乗（ピギーバック）して、タグがデータベースに送信されます。

Javaアプリケーションの場合はJDBCのsetClientInfo()メソッドまたはDMS APIを介して、C/C++アプリケーションの場合はDMS APIおよびOCIの新しい属性OCI_ATTR_DBOPを介して、DBOPおよび他のOCSIDを使用できます。

インテリジェントな接続と無効なクライアント接続の迅速な検出

接続試行時のNet Servicesの動作がさらにインテリジェントになり、応答しないノードは接続記述子のアドレス文字列内での優先順位が低くなるため、後続の試行時の接続時間が短縮され可用性が向上します。

同様に、終了しているセッション/接続の検出の高速化が図られています。

終了しているクライアントの検出時間は、sqlnet.ora構成ファイル内のSQLNET.EXPIRE_TIMEパラメータの設定で短縮できます。TCP keepaliveをサポートしているシステムの場合は、この拡張された検出モデルがOracle Net Servicesで自動的に使用され、TCP keepalive/パラメータが調整されます。

セキュリティ

データベースのJavaに対するデフォルトのJavaセキュリティのカスタマイズ

Oracle Databaseでは、Java実行時の権限とポリシーの管理機能が強化されました。Javaポリシーは、サード・パーティの暗号化スイートを追加した後でシステム管理者がリロードすることができます。また、アルゴリズムの検索順序を変更することもできます。

このような強化により、サード・パーティの暗号化ライブラリに対する権限とポリシーの管理が厳格化されるほか、柔軟かつ高度なセキュリティのサポートが実現できるようになりました。

JDBCのセキュリティ強化

Java Database Connectivity (JDBC)はOracle Databaseのセキュリティ強化に対応し、Kerberos認証、SHA-2およびWindows認証(NTS)などをサポートするようになりました。

この機能により、Javaアプリケーションのセキュリティが強化されます。

Oracle Advanced Securityオプション

Oracle Advanced Securityは、ハッシュ・アルゴリズムとしてMD5、SHA1、SHA256、SHA384またはSHA512を使用してセキュアなメッセージ・ダイジェストを生成し、ネットワーク経由で送信された個々のメッセージをそこに含めます。

今回のリリースで、SHA-224、SHA-256、SHA-384、SHA-512を含むSHA-2ハッシュ・アルゴリズムがJDBCでサポートされるようになりました。OCIのDRCPでは、暗号化と厳密認証（TCP以外）がサポートされるようになりました。

データベースでのJavaによるRuntime.execの起動

セキュリティ上の理由から、Runtime.exec機能によりフォークされるプロセスは、弱い権限を付与されたOS IDを使用して実行することが推奨されています。次のプロシージャは、データベース・ユーザー/スキーマDBUSERをOSのosuserアカウントに関連付けています。

```
dbms_java.set_runtime_exec_credentials('DBUSER', 'osuser', 'ospass');
```

Java スタンドード

Oracle Database 12cは、Java標準のJDBC 4.1、JDK 1.7、Java LoggingおよびJNDIもサポートするようになりました。

JDBC 4.1

このリリースでは、2つのシグネチャを持つJDBC 4.1のgetObject()メソッドがサポートされます。
getObject(int parameterIndex,java.lang.Class<T> type)

throws SQLException

getObject(java.lang.String parameterName,java.lang.Class<T> type) throws SQLException

例ResultSet rs = ...;

Character c = rs.getObject(1, java.lang.Character.class);

JDBC 4.1の他の機能にはtry-with-resourcesがあり、これはOracle JDBCで使用できますが、RowSetFactoryインタフェースはまだサポートされていません。

JDK 1.7および複数のJDK

このリリースでは、Oracle JDBCドライバとSQLJドライバが、いずれもojdbc7.jarを介してJDK 1.7をサポートするようになりました。

JDK 1.6（デフォルト）とJDK 1.7は、プレーンJava SEアプリケーションをRDBMSで直接実行できるようにするデータベース常駐型のJava VMでサポートされており、JDK 1.6と1.7の間でアップグレード/ダウングレードが可能です。

JNDI

このリリースでは、オブジェクトやオブジェクト参照の格納に使用される次の汎用ディレクトリ・サービスが、データベース常駐型のJava VMでサポートされるようになりました。

- Oracle Java Directory Serviceのネームスペース・プロバイダ (OJDS)
- Unixのファイル・システム構造/pub、/etcに類似したネームスペース
- OJDSに格納されたオブジェクトの権限やプロパティを参照できる、ネームスペース・ブラウザ (ojvmjavaユーティリティ)

ロギング

データベース常駐型のJava VMでロギング・プロパティがサポートされるようになりました。このプロパティは、データベース・リソース参照で拡張されたLogManager APIを使用して、各セッションにつき1回初期化されます。

- 現行のユーザー・スキーマに\$ORACLE_HOME/javavm/lib/logging.propertiesリソースが設定されている場合は、このリソースがLogManagerの構成に使用され、java.util.logging.config.fileプロパティが設定されます。
- このリソースが設定されていない場合は、SYSスキーマの\$ORACLE_HOME/javavm/lib/logging.propertiesリソースが使用されます。

異なるプロパティ・ファイルを構成して、loadjavaユーティリティによってそのファイルをスキーマにロードすることもできます。

アプリケーションの移行

Oracle Database以外のRDBMSに組み込まれているJavaアプリケーションやC/C++アプリケーションを簡単かつ低コストでOracle Databaseへ移行できるようにするために、前述した新しいデータ型の他に新機能として、Oracle以外のSQL構文をOracle Databaseに対して実行できるように変換するためのフレームワークとメカニズム、MySQLアプリケーションをOracle Databaseに対して実行できるようにするドライバが追加されました。

SQL変換フレームワーク

Oracle以外のデータベースをOracle Databaseへ移行する重要なプロセスには、Oracle以外のSQL文をOracle Databaseで処理できるSQL文に変換する作業が含まれます。アプリケーションに含まれるOracle以外のSQL文を変換する作業は、退屈な手動の工程です。この労力を最小化する、あるいはこうしたSQL文の変換を不要にすることを目的に、SQL変換フレームワークという新機能がOracle Database Release 12cに導入されました。クライアント・アプリケーションに含まれるOracle以外のSQL文はこのフレームワークに取り込まれ、ランタイムで変換されてから実行されます。

SQL文は、Oracle Database内でSQLトランスレータによって変換されます。トランスレータは、Oracle以外のクライアント・アプリケーションの変換を担当するものとして、SQL変換プロファイルで登録しておきます。SQL文の実行時にエラーが発生した場合は、Oracleエラー・コードやANSI SQLSTATEを、アプリケーションで想定されているベンダー固有の値に、SQLトランスレータで変換することもできます。変換された文はSQL変換プロファイルに保存されるため、任意で調査や編集が可能です。

SQL変換フレームワークの利点は次のとおりです。

- SQL文、Oracleエラー・コード、ANSI SQLSTATEの変換が自動的に実行される。
- 変換結果は一元管理され、調査が可能。
- 変換結果を抽出し、後からアプリケーションに戻すことができる。

今回のリリースでは、ODBC、OCIおよびJDBCを使用するJavaアプリケーションおよびC/C++アプリケーションでこの新しいSQL変換メカニズムを使用し、SQL文のテキストがOracle DatabaseのSQLコンパイラに



送信される前にOracle SQL構文に変換できるようになりました。

MySQLアプリケーション向けドライバ

MySQLアプリケーション向けOracle Databaseドライバは、MySQL 5.5のクライアント・ライブラリを一時的に置き換えるものです。これを使用すると、MySQL C APIに基づく言語（PHP、Ruby、Perl、Pythonなど）で構築されたアプリケーションやツールを、MySQL C APIを実装する新しいライブラリを使用してOracle Databaseに対して実行できます。

このドライバは、SQL変換フレームワーク（MySQLプロファイルを使用）、または自社製のカスタムSQL変換メカニズムのいずれかによってMySQLのSQL言語が変換されることを前提としています。

おもな利点は、MySQLとOracleの両方でMySQLアプリケーションを再利用できることと、MySQLアプリケーションをOracleに移行するときのコストと複雑さが低減されることです。

その他のリソース

この項で説明した製品と機能について詳しくは、次のURLを参照してください。

OCI : <http://www.oracle.com/technetwork/jp/database/features/oci/index-094133-ja.html>

Oracle Java, JDBC, UCP, & Database Web Services :

<http://www.oracle.com/technetwork/jp/database/enterprise-edition/overview/index.html>

Oracle Database Cloud Service

Oracle Databaseは世界有数のエンタープライズ・データベースです。Oracle Database Cloud Serviceはさまざまな機能を兼ね備えた堅牢な製品で、マウスを数回クリックするだけで、高速アプリケーション開発ツール、アプリケーションの即時デプロイメント機能、RESTful Webサービス・アクセス機能、一連のパッケージ・アプリケーションがインストールされます。Oracle Database CloudはOracle Database上に構築されたものであるため、Oracle Databaseのパワーと堅牢さを備えています。完全管理のサービスにさまざまな機能が追加されています。

クラウド・コンピューティングという観点から言うと、Oracle Database Cloud ServiceはOracle Databaseを基盤としたOracle Database常駐型の製品ですが、Database-as-a-Service製品というよりもむしろPlatform-as-a-Service製品であると言えます。Database Cloud Serviceには主要なコンポーネントが3つ付属しています。1つは、高速なアプリケーション開発およびデプロイメント・ツールであるOracle Application Expressです。このツールは2004年にリリースされて以来、エンタープライズ・レベルの強度を持つアプリケーションを開発できるツールとして実績があります。もう1つは、RESTful Webサービス・インタフェースです。SQLやPL/SQLへは、この代表的なWebサービス標準を介してセキュアにアクセスできます。最後の1つはアプリケーションのコレクションです。これは、数回のクリックでインストールと実行ができるため、即座にビジネス価値がもたらされます。現在Database Cloud Serviceで実行されているのはOracle Database 11gですが、将来的にはOracle Database 12cへアップグレードされる予定です。

Oracle Database Cloud Serviceについて詳しくは、次のURLを参照してください。

<https://cloud.oracle.com/mycloud/f?p=service:database:0>

SQLとPL/SQL

PL/SQLは、SQLコマンドをシームレスに処理することだけを目的に設計された必須の3GLです。PL/SQLにはこの目的に合わせた固有の構文があり、SQLとまったく同じデータ型がサポートされます。PL/SQLは他の環境でも使用できますが、本書では、Oracle Databaseに格納してコンパイルされ、Oracle Databaseの移植性を自動的に継承するOracle実行可能ファイル内で実行されるPL/SQLに限定して説明します。

多くのOracleユーザーが使用するベスト・プラクティスは、クライアント・コードによるOracle Databaseへのアクセスに必ずPL/SQLサブプログラムをコールするというものです。このアプローチには一般的なモジュール型プログラミングのベスト・プラクティス（クリーンで機能的なAPIを定義し、実装を隠ぺいするやり方）が反映されているだけでなく、このアプローチを取った場合はたいいてい、クライアントとOracle Database間のネットワーク・ラウンドトリップが激減します。

SQLとPL/SQLの重要な新機能

以降の項では、Oracle Database 12cに含まれるSQLとPL/SQLの新機能について説明します。

問合せの行制限と行オフセットのネイティブSQLでのサポート

FETCH FIRST句およびOFFSET句を使用すると、返される行数の制限と、戻される結果セットの開始行の指定ができるネイティブSQL問合せが有効化されます。

問合せでは、多くの場合、返される行数を制限したり、結果の開始行をオフセットしたりすることが必要になります。たとえば、上位N個の問合せでは、結果セットをソートし、最初のn行のみを返します。FETCH FIRSTとOFFSETを使用すると、構文が単純化され、ANSI SQL標準に準拠した形になります。

SQL CROSS APPLY、OUTERAPPLYおよびLATERAL

APPLY SQL構文を使用すると、問合せの外部表式から返される行ごとに表値ファンクションを呼び出すことができます。表値ファンクションは右側の入力として、外部表式は左側の入力として使用されます。右側の入力が左側の入力の各行に対して評価され、生成された行が結合されて最終出力となります。

したがって、表値ファンクションには左相関を渡すことができます。

APPLYには、CROSS APPLYとOUTER APPLYという2つの形式があります。CROSS APPLYは、表値ファンクションから結果セットを生成する外部表の行のみを返します。OUTER APPLYは、結果セットを生成する行と、生成しない行の両方を返します。結果セットを生成しない行の場合は、表値ファンクションによって生成される列にNULLが含まれます。

ANSI標準には、インライン・ビュー構文を拡張したLATERAL句があります。LATERAL句を使用すると、左相関の範囲がインライン・ビュー内になります。これらの新しいキーワードを使用することで、SQL問合せの結果をより簡単かつ柔軟性の高い方法で評価して返すことができます。

OracleネイティブのLEFT OUTER JOIN構文の拡張

このリリースでは、OracleネイティブのLEFT OUTER JOIN構文が拡張され、外部結合の左側に複数の表を指定できるようになりました。

Oracle Database 12cより前のバージョンでは、外部結合の左側に複数の表を指定すると構文違反になるため、ORA-01417エラーが発生しました。このような問合せを実行するには、問合せをANSI構文に変換するしか方法がありませんでした。Oracle Database 12cではLEFT OUTER JOINのネイティブの構文が拡張され、左側に複数の表を指定できるようになりました。この拡張には次の利点があります。

- 外部結合の左側に複数の表ビューをマージできる。このようなビューは、ユーザー問合せにより生成され



たものである場合や、LEFT OUTER JOIN構文を変換したときに生成されることもあります。

- このようなビューをマージすることで、結合の並べ替えをさらに実行できるようになり、より適切な実行計画が生成されます。これらのビューは経験則的にマージされるため、コストに基づく問合せ変換を行う必要がなくなります。
- ビューやLEFT OUTER JOIN構文の面で、問合せを作成する負担が軽減されます。

有効時間をサポートする表の定義

既存の列を使用するか、データベースによって自動的に作成された列を使用して、表に1つ以上の有効時間ディメンションを追加できます。この操作には、CREATE TABLE文とALTER TABLE文の両方を使用できます。

アプリケーションでは、データベースに記録された事実の妥当性を、アプリケーションで管理している基盤ビジネスに関連する日付やタイム・スタンプで示すことがよくあります。そのような日付の例としては、人事管理アプリケーションにおける従業員の雇用開始日や、保険契約の対象期間の発効日、住所変更の完了日などがあります。これは、その事実がデータベースに記録された日付や時刻とは対照的です。通常、前者のテンポラル属性を有効時間 (VT) と呼び、後者をトランザクション時間 (TT) と呼びます。有効時間は一般的にユーザーまたはアプリケーションによって制御されますが、トランザクション時間はシステムによって管理されます。フラッシュバック・データ・アーカイブはTotal Recallの一部としてOracle Database 11g Release 1 (11.1) で初めて導入された機能ですが、この機能にはトランザクション時間機能が実装されています。Oracle Databaseで有効時間がサポートされるようになり、簡単な宣言的インタフェースによってアプリケーション・コードの複雑さが緩和されたため、行の妥当性をデータベースで管理できるようになりました。

有効時間のテンポラル・フラッシュバック問合せ

このリリースでは、AS OF句およびVERSIONS BETWEEN句を使った問合せを、有効時間に基づいて実行できるようになりました。有効時間とトランザクション時間が混在する問合せはバイテンポラル問合せと呼ばれます。

今回のリリースでは、2つの時間ディメンションに基づくデータが含まれると考えられるすべてのビューに対して宣言的にアクセス権を付与することで、現在値(すなわち、有効時間とトランザクション時間におけるCURRENT)か、現在わかっている値(すなわち、有効時間におけるAS OFとトランザクション時間におけるCURRENT)か、以前わかっていた値(すなわち、有効時間とトランザクション時間におけるAS OF)に基づいてデータを問い合わせることができるようになりました。Oracle Database 12c Release 1 (12.1) のバイテンポラル問合せを使用すると、これまでは大量の複雑なアプリケーション・コードを記述しなければ使用できなかった機能が使用できます。

SQL WITH句でのPL/SQLファンクションの定義

このリリースから、副問合せのWITH句にPL/SQLファンクションを定義して通常ファンクションとして使用できるようになりました。

SQL文をサポートするために必要なプロシージャ・ロジックは、SQL文でカプセル化されます。これは、読取り専用データベースで特に便利です。

このコンストラクトを使用すると、スキーマ・レベル・ファンクションよりもパフォーマンスが向上します。

PL/SQLからSQLへのインタフェースで利用できるPL/SQL固有のデータ型

Oracle Database 11g Release 2 (11.2) までは、PL/SQLでSQLを起動すると、SQLでサポートされているデータ型の値しかバインドできませんでした。コールされるSQLがPL/SQL無名ブロックの場合でも、このような制限がありました。Oracle Database 12c Release 1 (12.1) にはこの制限がありません。たとえば、データ型がBOOLEANの仮パラメータを伴うPL/SQLサブプログラムを、無名ブロックを使用して動的に起動できるようになりました。

撤回された制限は他にもあります。データ型がPL/SQLで宣言されたコレクションに対するPL/SQLプログラムで、表演算子を使用できるようになりました。このとき、データ型としてPL/SQL連想配列を使用することもできます(以前のリリースでは、スキーマ・レベルでコレクションのデータ型を宣言する必要がありました)。

これらの制限がなくなったことで、式の効力とPL/SQLの有用性が向上しています。特に、表演算子の柔軟性が向上したことにより、他のベンダーのストアド・プロシージャ言語を実行できるように記述されたコードを、簡単にPL/SQLに移行できるようになりました。

PL/SQLユニットから参照できるデータベース・オブジェクトのホワイト・リストを制限するメカニズム

今回のリリースでは、スキーマ・レベルのファンクションやプロシージャ、パッケージ、型指定を、許可されたコール元のホワイト・リストでマークできるようになりました。許可されたコール元としては、PL/SQLサブプログラムを起動できる任意のオブジェクト・タイプのもの(トリガー、ビュー、表、索引など)を指定できますが、ホワイト・リストがあるユニットと同じスキーマに存在しているものである必要があります。ホワイト・リストの使用は任意ですが、使用すると、リストされているオブジェクトからでなければ対象のユニットを参照できなくなります。したがって、SYSが所有するスキーマから参照を試行した場合でも、ホワイト・リストを設定したユニットへの相互スキーマ参照は許可されません。

この機能を使用すると、ヘルパー・ユニットがヘルプ対象のユニット以外からアクセスすることを禁止するため、メイン・ユニットとヘルパー・ユニットで構成されるモジュールの堅牢な実装が実現します。

起動者権限ファンクションによる結果のキャッシング

Oracle Database 11 Release 2 (11.2) までは、定義者権限のPL/SQLファンクションでなければ結果をキャッシュできませんでした。このリリースでは、起動者権限のPL/SQLファンクションでも結果をキャッシュできるようになりました(起動ユーザーのIDは結果のキーに暗黙に追加されます)。

起動者権限のPL/SQLファンクションを使用して、1つ以上のSELECT文を発行することが適切な場合もあります。この機能を使用することで、パフォーマンスが向上します。

DIRECTORYタイプのオブジェクトによるLIBRARYタイプのオブジェクトの定義

以前のリリースでは、LIBRARYタイプのオブジェクトを定義するには、明示的なパスを使用しなければなりませんでした。しかし、このリリースでは、DIRECTORYタイプで一元的にファイル・システム・パスを管理できるようになりました。さらに、DIRECTORYタイプを使用することには、セキュリティ上のメリットもあります。ディレクトリ・オブジェクトは、DIRECTORYタイプを使用して定義できます。

また、LIBRARYタイプのオブジェクトの定義に資格証明を含めることができるようになったため、指定した外部プログラムを、Oracleインストールの所有者ではなく別のオペレーティング・システム・ユーザーとして実行できるようになりました。

このような拡張により、外部プロシージャを使用するアプリケーションのセキュリティと移植性が向上します。

新しいPL/SQLパッケージUTL_CALL_STACK

UTL_CALL_STACKパッケージは、PL/SQLプログラムの現在のコール・スタックを返すサブプログラムを提供



します。

これは、判読可能な文として情報を返す既存のDBMS_UTILITY.FORMAT_CALL_STACKプロシージャに機能的に似ています。この新しいパッケージを使用すると、プログラムによる分析に適した構造化された表現として、この情報を利用できるようになります。

新しいPL/SQLサブプログラムDBMS_UTILITY.EXPAND_SQL_TEXT

DBMS_UTILITY.EXPAND_SQL_TEXTサブプログラムは、ビューを参照する副問合せを受け取り、表のみを参照する同一の意味を持つ副問合せを返します。

この機能が役立つのは、アプリケーション・ロジックの修正やパフォーマンスの問題の解決を目的に、ビューに依存するSQLを分析するような場合です。

新しい事前定義済みのPL/SQL問合せディレクティブ

事前定義済みのPL/SQL問合せディレクティブ、\$\$PLSQL_OWNERと\$\$PLSQL_TYPEが、このリリースでサポートされるようになりました。

Oracle Database 11g Release 2 (11.2) までは、事前定義済みの問合せディレクティブ\$\$PLSQL_LINEと

\$\$PLSQL_UNITを使用して、診断コードで現在のPL/SQL文を特定することができましたが、不正確な部分がありました。この不正確な部分が、今リリースでは排除されています。

DBMS_SQL.PARSE()の新しいSCHEMAパラメータ

DBMS_SQL.PARSE()に、新たにSCHEMAパラメータが追加されました。これは、未修飾のオブジェクト名を解決するスキーマを指定するパラメータです。

このパラメータを使用すると、定義者権限ユニットから発行した動的SQLの名前解決を定義者権限ユニットで制御できるようになります。PL/SQLについて詳しくは、次のURLを参照してください。

<http://www.oracle.com/technetwork/jp/database/features/plsql/index.html>

Oracle SQL Developer

Oracle SQL DeveloperはOracle Databaseの開発と管理を簡素化する無償の統合開発環境です。PL/SQLアプリケーションのエンド・ツー・エンドの開発ができるほか、問合せやスクリプトを実行するためのワークシート、データベース管理用のDBAコンソール、レポート・インタフェース、完全なデータ・モデリング・ソリューション、サード・パーティ製データベースをOracleに移行するための移行プラットフォームを備えています。

Oracle SQL Developerのコア領域

開発 - データベース・オブジェクトの参照、編集、作成ができるほか、SQL文の実行、PL/SQL文の編集とデバッグ、PL/SQLユニット・テストの構築、レポートの実行、バージョン管理下へのファイルの配置ができます。

データベース管理 - データベースの管理性を向上させるDBAナビゲータや、生産性とパフォーマンスを最大化するDatabase DiffやDatabase CopyなどのSQL開発者用のDBAユーティリティが用意されているほか、RMAN、Data Pump、リアルタイムSQL監視、EXPLAIN PLANおよび自動トレースに対応したグラフィカル・ユーザー・インタフェースも揃っています。

モデリング - Oracle SQL Developer Data Modelerは、生産性の向上とデータ・モデリング・タスクの簡素化



を実現するグラフィカル・ツールです。Oracle SQL Developer Data Modelerを使用して、論理モデル、リレーショナル・モデル、物理モデル、多次元モデル、およびデータ型モデルを作成、参照および編集できます。フォワード/リバース・エンジニアリング機能があるほか、統合されたソース・コード管理機能を介した共同開発もサポートしています。Oracle SQL Data Modelerは、スタンドアロン・ツールとして使用することも、Oracle SQL Developer内で使用することもできます。

移行 - Oracle SQL Developerは、サード・パーティ製データベースからOracleへの移行をサポートしています。機能が緊密に統合されているため、サード・パーティ製データベースのデータベース・オブジェクトとデータを参照したりOracleへ移行したりといった作業を一箇所で実行できます。Oracle SQL Developerは、IBM DB2 UDB LUW、Microsoft SQL Server、Microsoft Access、MySQL、Sybase Adaptive Server、Teradataをサポートしています。

他のデータベースからOracle Databaseへの移行について詳しくは、次のURLを参照してください。

<http://www.oracle.com/technetwork/jp/products/migration/index-084442-ja.html>

利点

Javaで開発されたOracle SQL Developerは、Windows、Linux、およびMac OS X上で稼働します。これは、さまざまなプラットフォームを使用する開発者が増え続けている現状においては大きな利点です。複数のプラットフォームに対応しているということは、データベースと同じシステムにOracle SQL Developerをインストールしてデスクトップからリモート接続できるということであり、クライアント/サーバーによるネットワーク・トラフィックが発生しなくなるということも意味します。データベースへのデフォルト接続はJDBC Thinドライバ経由であるため、Oracleホームは不要です。Oracle SQL Developerは、ダウンロードしたzipファイルを解凍するだけでインストールできます。Oracle SQL Developerを使用すると、ユーザーは、

Oracle Database Express Editionも含め、サポートされているどのOracle Databaseにも接続できます。また、Oracle以外のデータベース用のデータベース接続を作成し、オブジェクトやデータを参照したり移行したりすることもできます。Oracle SQL Developerは、Oracle Databaseのサポート契約を結んでいるすべての顧客を対象に、サポートが提供される製品です。

Oracle Database 12cのサポート

Oracle SQL Developer Release 4.0でも引き続き、最新のOracle Database 12cの機能がサポートされます。

Oracle Multitenant – Oracle Database 12cに新しいアーキテクチャが導入され、1つのOracle Databaseインスタンスの内部に多数のプラガブル・データベースを保持できるようになりました。DBAは、Oracle SQL Developerを使用して12cのデータベースに接続し、コンテナ・データベースを1つ作成して複数のプラガブル・データベースを統合することができます。プラガブル・データベースの作成、削除、クローニング、アンプラグ、同期などの管理機能はすべて、使いやすいウィザードから使用できます。

SQL変換フレームワーク – SQL Developerは、Oracle向けの公式データベース移行ツールです。Oracle Database 12cにSQL変換フレームワークが導入されたことで、Oracleへのアプリケーション移行をサポートできるようになりました。SQL変換フレームワークを使用すると、アプリケーション・コードの変更を最小限に抑えられます。Oracle以外のSQLは変更されないままOracle Databaseに送信され、アプリケーションのSQLはここで記録および変換されます。ユーザーは、変換されたSQLをレビュー、チューニング、変更することができます。変換フレームワークを構成する要素は、変換された文を収集する変換プロファイルと、SQLを変換するJavaコンパイラとしてのSQLトランスレータです。Oracle SQL Developerを使用すると、各アプリケーションのSQL変換プロファイルの作成とSQLトランスレータのインストールをユーザーが実行できます。Oracle SQL Developerでは、Sybase向けとSQL Server向けのトランスレータがサポートされています。

Oracle Data Redaction – Oracle Database 12cのOracle Advanced Securityオプションに含まれる機能です。



権限の低いユーザーまたはアプリケーションが発行した問合せの結果データをマスクすることができます。完全、部分、ランダム、正規表現など、さまざまなマスキング・スタイルを適用してデータをマスクすることができます。これらのマスキング・スタイルを適用する場合は、改訂ポリシーを使用します。Oracle SQL Developerを使用すると、改訂ポリシーを一元管理できます。ポリシーを作成して適用することで、どのような機密データも非表示にすることができます。

必ず最新バージョンのOracle SQL DeveloperをOracle Technology Network (OTN) からダウンロードする必要があります。Oracle SQL Developerについて詳しくは、次のURLを参照してください。

<http://www.oracle.com/technetwork/jp/developer-tools/sql-developer/overview/index.html>

Oracle Application Express

Oracle Application Express (Oracle APEX) は、Oracleデータベース用の宣言型高速Webアプリケーション開発ツールです。完全にサポートされた無償のオプションとして、Oracleデータベースのすべてのエディションで使用できます。Webブラウザを使用するだけで、高パフォーマンスでセキュアなプロフェッショナル・レベルのアプリケーションを開発してデプロイできます。

SQLやPL/SQLの経験豊富な開発者やレポート作成に慣れているパワー・ユーザーも、ウィザードを使用すれば、Oracleデータベース・オブジェクト上にWebアプリケーションを迅速に作成できます。アプリケーションの機能拡張や保守は、宣言型フレームワークを使用して実行できるため、生産性が向上します。

Oracle Application Expressはデータベース中心で、多種多様なアプリケーションの構築に適しています。共同作業を容易にするスプレッドシートのWeb化から開始することも、外部インタフェースを多数持つOracle Storeのような非常に複雑なアプリケーションに挑むこともできます。Oracle APEXはOracle Database内に存在し、認証スキーム (Oracle Access Manager、SSO、LDAPなど) と容易に統合できるため、最大規模のユーザー・コミュニティに合わせて拡張できるセキュアなアプリケーションを構築できます。

Oracle Application Expressには、おもなツールとして次の4つが用意されています。

- ・ **アプリケーション・ビルダー** - データベースに基づく動的なWebアプリケーションを作成できます。
- ・ **SQLワークショップ** - データベース・オブジェクトの表示と管理、非定型SQL問合せの実行ができ、グラフィカル・クエリー・ビルダーもあります。また、データのロード/アンロード、ユーザー・インタフェースのデフォルトの設定、DDLスクリプトの生成、データベースのさまざまな側面の確認ができるユーティリティも用意されています。RESTful Webサービスを使用した場合は、Oracle APEXリスナー2.0以降を使用するSQLおよびPL/SQLを介して、データとロジックヘステートレスにアクセスできます。
- ・ **チーム開発** - To Do、バグ、マイルストーン、フィードバックの収集などの統合された機能を使用して、ソフトウェア開発プロジェクトを管理できます。
- ・ **管理** - サービスの管理、ユーザーの保守、アクティビティの監視といった、Application Expressワークスペースの管理ができます。

Oracle Application Expressを利用すると、データベース・データに関するレポートを作成するアプリケーションを簡単に構築できます。レポートには、他のレポートへのハイパーテキストをリンクとして追加することができるため、Webサイトをナビゲートするようにデータベース内のデータをナビゲートできます。レポート内の列を、他のレポート、グラフ、データ入力フォームに簡単にリンクさせることができます。豊富な機能を持つグラフ・エンジンを使用し、SQL問合せをグラフィカルに表現することができます。Oracle Application Expressはデータベース・データの編集にも非常に適しており、多数の宣言的なフォーム・コントロール (ラジオ・ボタン、チェック・ボックス、選択リスト、シャトル、テキスト・エディタ、日付ピッ



カーなど)をサポートしています。

多くの組織は、ワークグループ・データベースが急増するという事態に遭遇しています。こうしたデータベースは、そもそもユーザーがコントロールできる当座のソリューションが必要とされたことで作成されたものです。それぞれの部門が戦術上の課題を解決するために、一元化されていないデータベースを絶えず構築しています。非効率なその場限りのデータベースが何百もデプロイされているため、コストが跳ね上がります。また、これらのデータベースにはビジネス・クリティカルなデータが含まれることがよくありますが、多くの場合、セキュリティ、共有、データ精度、高可用性に問題があります。残念なことに、これらのデータベースのユーザーがIT組織の足を引っ張る結果になります。

Oracle Application Expressの統合モデルは分散された開発環境に一元化管理機能を提供します。Oracle Application Expressを使用すれば、ワークグループ・データとアプリケーション開発サービスをOracle Database 12cに統合することができます。アプリケーション開発に対するコントロールは開発者が完全に維持できる一方で、IT組織はデータ管理に徹することができます。エンドユーザーは、最新のWebベース・アプリケーションを介したデータ・アクセスが、劇的に改善されたことを実感します。Oracle Database 12cは、Oracle Application Expressアプリケーションのセキュリティ、信頼性、スケーラビリティを確かなものにします。Oracle Application Expressは、ブラウザベースの設計時インタフェースや宣言的なプログラミング・フレームワーク、簡単なウィザードを備えているため、Microsoft Accessなどのマルチユーザー向けワークグループ・データベース・アプリケーションの置き換えに最適なツールです。

Web開発経験をほとんど、またはまったく持たない開発者でも、SQLを知っていれば、Oracle Application Expressを利用して簡単にデータベース・アプリケーションを作成できます。スクリプト言語や複雑なデプロイメント・フレームワークを習得する必要はありません。いくつかの問合せを書き、用意されているユーザー・インタフェース・テーマやフォーム・コントロールの中から適切なものを選択するだけで、専門性の高いセキュアでスケーラブルなアプリケーションを開発できます。

Oracle Application Express 4.2のおもな機能

Oracle Database 12cに付属するOracle Application Express 4.2には、次のおもな機能があります。

インタラクティブ・レポート

インタラクティブ・レポートを使用すると“SELECT * FROM My_Favorite_Table”などの文を非常に簡単に記述できますが、極めて強力なレポート機能も使用できます。列を選択して、ソートやフィルタを適用し、計算、ハイライト、集計を追加するなどのデータ操作を、エンドユーザーが実行することができます。このようにして作成したレポートは、保存してダウンロードできるようにしたり、サブスクライブの対象にしたりすることもできます。

スプレッドシートを使用した対話型操作

開発者は、スプレッドシートのデータを既存の表へアップロードできるエンドユーザー向けウィザードをOracle Application Expressで簡単に構築できます。IT部門に依頼してデータを既存の表にアップロードしてもらうより、この機能を使用して、エンドユーザーが簡単なウィザードの手順に従ってデータを既存の表にアップロードできるようにしたほうが、エンドユーザー自身でできることが増加します。

また、開発者やパワー・ユーザーは、Create Applicationウィザードを使用して、スプレッドシートに基づく完全なCRUDアプリケーションを簡単に作成できます。そのため、スプレッドシートを配布する代わりに、スプレッドシートのデータに基づくアプリケーションをユーザー自身に数分で作成してもらうことができます。作成アプリケーションのURLは配信することができるため、信頼できる情報源が1つだけになり、データはOracle Database 12cに安全に格納されます。

グラフ機能

Oracle Application Expressには“Flash推奨”またはHTML5のさまざまなグラフが搭載されており、データを視



覚的に表示することができます。“Flash推奨”として定義すると、iPad上などのようにFlashを使用できない環境では、HTML5を使用してグラフが表示されます。国や州全体をマップで表示することや、プロジェクト・ガント・チャートやリソース・ガント・チャートを定義することもできます。

マスター・ディテール・フォームと表形式フォーム

Oracle Application Expressに含まれるウィザードを使用すると、複数のレコードを一度に更新するためのページを数分で作成できます。マスター・ディテール・ページの定義はごく簡単で、親表を選択して子表を選択し、表示するフィールドをそれぞれの表から選択するだけです。この複数レコード・コンストラクトには、行と列の検証をすばやく定義する機能が含まれているため、通常のフォーム・ページに含まれる1件のレコードを検証するのと同じくらい容易に行を検証できます。

動的アクション

動的アクションを使用すると、クライアント側のさまざまな対話型機能を宣言的に組み込むことができます。JavaScriptやAJAXの知識がない開発者でも、この機能を使用すれば、クライアント側の動作を宣言的に定義できます。操作を実行するタイミング、操作の内容、操作の対象の指定は、簡単なウィザードと宣言的なコンストラクトを使用して行います。手動作成したJavaScriptやAJAXを宣言的な定義で置き換えると、クライアント側のさまざまな対話型機能の品質、一貫性、管理性が大幅に向上します。

プラグイン

プラグインを使用すると、独自のアイテム・タイプ、リージョン・タイプ、動的アクション、認証スキームおよび認可スキームを開発したり共有したりできます。プラグインはOracle Application Expressアプリケーションでカバーされる範囲を広げる機能であり、Oracle Application Express向けのカスタム機能のライブラリが用意されています。開発者が必要とする機能がネイティブ・コンポーネントに含まれていない場合でも、このアーキテクチャを使用すれば、サポートとメンテナンスの対象になっている方法でアプリケーションを拡張できます。使用できるプラグインは、Oracle Application Express開発チームが作成したものが多数あり、APEXコミュニティ・サイトにも120以上があります。

モバイル・アプリケーション

モバイル・アプリケーションや、HTML5グラフ、HTML5アイテム・タイプ、モバイル・カレンダーをはじめとするモバイル・アプリケーション・コンポーネントを宣言的に定義することができます。また、デスクトップ向けとモバイル向けの2つのユーザー・インタフェースを備え、自動検出機能も備えたアプリケーションを開発できます。モバイル・アプリケーションの構築にはjQuery Mobileを使用します。iOS、Android、Blackberry、Windowsなど、モバイル・オペレーティング・システムの種類に合わせて個別にアプリケーションを構築するかわりに、どのモバイル・デバイスでもモバイル・ブラウザを使用して同じアプリケーションを稼働させることができます。

モバイル・アプリケーションの開発には、デスクトップ・アプリケーションの開発に使用すると同じビルダーおよび宣言的なフレームワークを使用します。そのため、SQLとPL/SQLという同じスキル・セットを使用して、極めて短期間でモバイル・アプリケーションを構築できます。

Oracle Application Expressには、HTML5のグラフやカレンダーも含まれているだけでなく、メールや電話番号、URL向けにレイアウトが最適化されたキーボードなど、他のHTML5属性も含まれています。

パッケージ・アプリケーション

Oracle Application Expressには多数のパッケージ化されたビジネス・アプリケーションが含まれていますが、これらは無償で使用でき、サポートの対象にもなっています。生産性向上のためのアプリケーションでビジネス・プロセスを改善できるため、Oracle Database 12cへの投資はすぐに活用され始めます。また、Oracle Application Expressの固有機能を開発者に短期間で理解してもらえるように設計されたサンプル・アプリケーションも一式揃っています。



この生産性向上のためのアプリケーションはデフォルトでロックされており、開発者は変更することができません。パッケージ・アプリケーションを変更したい、または単に実装を確認してベスト・プラクティスを学びたいという場合は、簡単にロックを解除できます。一度ロックを解除したアプリケーションは、オラクルのサポート対象ではなくなり、更新されなくなりますが、ロックを解除したコピーを削除し、ロックされているバージョンを再インストールすれば、いつでもサポート対象のバージョンに戻すことができます。

Webシート

Webシートは、Oracle Application Expressに含まれる別種のアプリケーション開発機能ですが、Oracle Database内のデータをWebブラウザから管理する際のハードルは一段と低くなっています。Webシートは開発者よりむしろビジネス・ユーザーを対象に設計されています。Webシートは設計環境と実行時環境が1つになったもので、操作はWikiに似ています。エンドユーザーはWebブラウザのみを使用してページ、データ・グリッドおよびレポートを定義し、定義したコンテンツを表示および編集できるコミュニティを選択することができます。

Wikiの場合はいったん追加したデータは古くなっていく一方であるのに対し、Webシートの場合はデータ要素をページに直接組み込むという点が、WikiとWebシートのおもな相違点です。

Webシート内では、データ・グリッド（データベースに格納されるスプレッドシート）を定義したり、各自のOracle Database 12cスキーマ内の表にレポートを定義したり、表に対して直接SQLを記述したりできます。

ユーザーがWebシートのページにアクセスすると、必ずその時点でデータベースに直接問い合わせたデータが表示されます。

必ず最新バージョンのOracle Application ExpressをOracle Technology Network（OTN）からダウンロードする必要があります。Oracle Application Expressについて詳しくは、次のURLを参照してください。

<http://www.oracle.com/technetwork/jp/developer-tools/apex/overview/index.html>

Oracle REST Data Services

Oracle REST Data Servicesを使用すると、Oracle Database向けの最新式RESTインタフェースを簡単に開発できます。主要なベンダーが提供するGoogle、Facebook、Twitterなどのサービスを含め、インターネット上のサービスにアクセスするためのインタフェースとしては、RESTが優勢になってきています。また、RESTはSOAPなどの標準規格に代わる強力ながらもシンプルな規格であるため、企業での採用も急速に進んでいます。RESTサービスでは、共通機能とも呼べるシンプルなHTTPコールが使用されるため、さまざまなクライアント言語環境からRESTサービスを起動することも容易です。

RESTインタフェースの設定は簡単です。各RESTful Webサービス・コールURIで起動したいSQLブロックまたはPL/SQLブロックと、データを返す方法を指定するだけです。返されたデータは、JSON形式または.csv形式のいずれかにマーシャリングできます。

Oracle REST Data Servicesは、Oracle Database Cloud Serviceとしても、オンプレミスでも使用できます。オンプレミス・デプロイメントの場合、Oracle REST Data Servicesは、Oracle Web Logic Server（WLS）、Oracle Glassfish ServerおよびApache Tomcat内のJ2EEコンポーネントとして実行されます。Oracle REST Data ServicesはOracle Application Express（APEX）に完全に統合されたファシリティとして使用することもできます（Oracle REST Data Servicesは、これまでAPEXリスナーと呼ばれていました）。

Release 2.0の新機能

Oracle SQL Developerによる管理 – Oracle SQL Developer Release 3.2.2以上を使用してOracle REST Data



Servicesインスタンスの管理ができます。また、SQLエディタなどの機能を使用して、新しいRestfulサービスの作成、テスト、および検証ができます。

- » コマンドラインのサポート - コマンドラインを使用してOracle REST Data Servicesを構成できます。
- » 複数データベースのサポート - Release 2.0では、複数のデータベース接続を構成できます。
- » セキュリティの強化 - RESTfulサービスを使用して生成されたリソースへのアクセスは、さきごろ最終決定されたOpen Authorization 2.0 (OAuth2) フレームワークを使用して保護されるようになり、クロス・オリジン・リソース共有の仕様のサポートも追加されました。
- » ページ区切りとリソース・ハイパーリンク - RESTfulサービスで結果のページ区切りがサポートされることになったため、大きな結果セットを1つのリソースではなく多数のページ・リソースに分割できるようになりました。また、他のリソースへのハイパーリンクを結果に含めるメカニズムも用意されました。

.NET

オラクルは、Oracle Databaseを用いた.NET開発を容易にする4つのコンポーネントを提供しています。4つのコンポーネントとは、Oracle Developer Tools for Visual Studio、Oracle Data Provider for .NET (ODP.NET)、Oracle Database Extensions for .NET、およびOracle Providers for ASP.NETです。これらのコンポーネントは無償で、初心者でも上級のデータベース・プログラマーでも簡単に使用できます。これらのコンポーネントでは、Entity Frameworkなどの既存のMicrosoft標準テクノロジーと、Visual Studioなどのツールが完全にサポートされると同時に、Oracle RACやOracle Data Guardをはじめとする強力なOracle Databaseエンタープライズ・テクノロジーも使用できます。

Oracle Developer Tools for Visual Studio (ODT) は、Microsoft Visual Studio向けに緊密に統合された"アドイン"です。ODTは無償で、Visual Studio 2012、Visual Studio 2010、Visual Studio 2008で利用できます。

ODTを使用すると、Oracle向けの.NETコードの開発が容易かつ迅速になり、開発者は開発ライフ・サイクル全体を通してVisual Studioから作業を実施できます。Oracleスキーマ・オブジェクトの参照や編集は、統合されたビジュアル・デザイナを使用して容易に行うことができ、.NETコードはドラッグ・アンド・ドロップ操作だけで自動生成できます。また、表データの変更、Oracle SQL文の実行、PL/SQLコードの編集およびデバッグも簡単で、スキーマ比較ツールを使用してSQLデプロイメント・スクリプトを自動生成し、それをソース管理システムにチェックインするのも簡単にできます。Oracle Multitenantと統合されたことで、クローン・データベースを作成してテストを実行するのも容易になりました。Oracle SQLやPL/SQLのユーザーズ・ガイドなどは状況依存のオンライン・ヘルプとして統合されているため、Oracle Databaseのドキュメントを簡単に参照できます。

Oracle Developer Tools for Visual Studioについて詳しくは、次のURLを参照してください。

<http://www.oracle.com/technetwork/jp/developer-tools/visual-studio/overview/index-086829-ja.html>

ODP.NETは.NET環境からOracleデータベースへのデータ・アクセスを最適化します。ODP.NETを使用すると、Oracle RACやOracle Data Guard、セルフチューニングやキャッシング機能、XML DBやREF Cursor、セキュリティ機能など、Oracle Databaseの先進的な機能の利用が可能になります。また、生産性が向上し、.NET FrameworkおよびOracle Database開発の自由度と機能選択の幅が広がります。ODP.NETを使用すれば、オラクルの強力なデータ管理機能を.NETでも活用できます。ODP.NET 12cは.NET Framework 3.5以上をネイティブにサポートします。

ODP.NETについて詳しくは、次のURLを参照してください。

<http://www.oracle.com/technetwork/jp/topics/index-086363-ja.html>

Oracle Database Extensions for .NETはWindows上で動作するOracle Database 12cの機能で、この機能を使用すると、ストアド・プロシージャやストアド・ファンクションをC#やVB.NETなどの.NETマネージド言語での開発、デプロイ、実行が容易になります。.NETのストアド・プロシージャやストアド・ファンクションの開発にはMicrosoft Visual Studioを使用し、デプロイには緊密に統合された.NET Deployment Wizardを使用します。このウィザードはOracle Developer Tools for Visual Studioに含まれる機能です。デプロイされた.NETストアド・プロシージャは、.NETアプリケーション・コード内、SQLもしくはPL/SQL、別の.NET、PL/SQL、Javaのストアド・プロシージャ、トリガー、またはそれ以外でも、ストアド・プロシージャやストアド・ファンクションの呼出しが可能な場所ならどこからでも呼び出すことができます。

Oracle Database Extensions for .NETについて詳しくは、次のURLを参照してください。

<http://www.oracle.com/technetwork/jp/topics/dotnet/whatsnew/index-352676-ja.html>

Oracle Providers for ASP.NETは、ASP.NETプロバイダ・モデルに従い、Oracle Databaseをデータ・ソースとして使用するデータ・サービスのコレクションです。開発者は、ショッピング・カートやユーザー情報などのWebアプリケーション状態を永続的なOracleデータベースに格納できます。ASP.NET開発者は.NET Frameworkの構成要素であるASP.NETサービスとコントロールを介してWebアプリケーションをビルドするため、生産性が向上します。Oracle Providers for ASP.NETを構成ファイルでデフォルト・プロバイダとして構成するだけで、さまざまな種類のアプリケーション状態をASP.NETアプリケーションでOracleデータベースに格納できます。オラクルが提供しているプロバイダは、メンバーシップ・プロバイダ、ロール・プロバイダ、サイト・マップ・プロバイダ、セッション・ステート・プロバイダ、プロファイル・プロバイダ、Webイベント・プロバイダ、Webパーツ・パーソナライズ・プロバイダ、キャッシュ依存性プロバイダです。

Oracle Providers for ASP.NETについて詳しくは、次のURLを参照してください。

<http://www.oracle.com/technetwork/jp/topics/dotnet/whatsnew/index-352675-ja.html>

.NETのおもな新機能

ODP.NETのマネージド・ドライバ

Oracle Data Provider for .NET 12では、100%ネイティブな.NETコードを含む完全管理の新しいプロバイダ・バージョンを導入しています。ODP.NETのマネージド・ドライバは、ODP.NETのアンマネージド・ドライバのほとんどすべての機能を含み、同じアプリケーション・プログラミング・インタフェースを使用しています。このため、既存のODP.NETアプリケーションを簡単にODP.NETのマネージド・ドライバへ移行できます。

ODP.NETのマネージド・ドライバを使用すると、ODP.NETのデプロイがより簡単で迅速になります。デプロイすべきアセンブリがわずか1つと少なくなっているため、パッチの適用も簡単になります。インストール・サイズは、10MB未満です。使用している.NET Frameworkが32ビットか64ビットかにかかわらず、ODP.NETのマネージド・ドライバに必要なアセンブリは1つだけです。考慮すべきアンマネージド・アセンブリが存在しないため、他のバージョンのODP.NETとの同時デプロイが容易です。完全管理のプロバイダであるODP.NETでは、.NETコード・アクセス・セキュリティおよびClickOnceのデプロイとの統合が強化されます。

Oracle Multitenant

ODTとODP.NETはOracle Multitenantとシームレスに統合されているため、開発やテスト時に使用するクローン・データベースを簡単かつ迅速に作成できます。作成したプラガブル・データベースは、"Clone"、"Plug"、"Unplug"などの操作を使用してVisual StudioのServer Explorerから直接表示および管理できます。ODP.NETはPDBと一緒に標準で機能し、.NETでのPDBの使用にコード変更を必要としません。

スキーマ比較ツール

ODTでは、Visual Studio内に統合されたスキーマ比較ツールが導入されています。これらのツールにより、開発者は、個々のOracleスキーマ・オブジェクト間またはスキーマ全体における変更を検出できます。デプロイ時になると、これらのツールを使用して、必要な新しいスキーマ変更を含めるために、ターゲット・データベースをアップグレードするデプロイメント ("diff") スクリプトが生成できます。

トランザクション・ガード – 高可用性

トランザクション・ガードは、計画停止および計画外停止中のトランザクション結果を維持し、すべてのトランザクション結果がODP.NETアプリケーションに認識されるようにします。そのため、同じトランザクションが繰り返しコミットされることがなくなります。停止が起これば、ODP.NETは、トランザクションがリカバリ可能かどうかを示すOracleException.IsRecoverableプロパティを返します。リカバリ可能な場合は、ODP.NETが論理トランザクション識別子を取得し、コミットされていない最後のトランザクションの結果ステータスを判定します。このトランザクション・ステータスの場合は、計画がクリアされた状態で、.NET アプリケーションによるトランザクションのリカバリを続行できます。

計画停止は、エンドユーザーの操作を中断することなく実行できるようになりました。データベースの計画停止ステータスをODP.NETアプリケーションに送信することにより、データベースを迅速にオフライン状態にします。続いて、ODP.NETによってアイドル状態のプール接続が削除され、データベースへの新規接続が禁止されます。

グローバル・データ・サービス – 高い可用性とパフォーマンス

グローバル・データ・サービスを使用すると、Oracle RACの自動ワークロード管理機能をODP.NETアプリケーションでActive Data GuardインスタンスおよびOracle GoldenGateインスタンスまで拡張できます。.NETアプリケーションでは、使用できるグローバル・データベース・リソースをすべて活用して、パフォーマンスと可用性を向上できます。

Entity Frameworkおよび統合言語クエリー – SQL APPLY

ODAC 11.2 Release 4では、Entity Frameworkおよび統合言語クエリー (LINQ) がODTおよびODP.NETと統合されています。LINQは、データベースに問合せを実行する前にネイティブ・データベースSQLに変換されます。環境によっては、LINQはラテラル・ビューの取得に非標準のAPPLYキーワードをSQLで使用します。Oracle DatabaseおよびODP.NETは、Oracle Database 12cでAPPLYキーワードをサポートすることで、LINQをより完全にサポートしています。

容易な開発

Oracle Database 12cおよびODP.NET 12cは、データベース開発を容易にする次の新機能をサポートしています。

- 自動増分ID列 – 自然主キーのないデータに適した.NETの開発を容易にします。
- VARCHAR2データ型、NVARCHAR2データ型、およびRAWデータ型の拡張 – 各データ型はそれぞれ最大32KBのデータ・サイズを格納できるようになりました。
- Booleanデータ型 – ODP.NETデータ型のOracleBooleanは、新しいデータベースのPL/SQL Booleanデータ型へマッピングします。
- 暗黙的なREF Cursorのバインディングの拡張 – Entity Frameworkおよびユーザー定義型の場合を除いて、宣言された戻り値の型なしにストアド・プロシージャの結果セットを暗黙的に取得します。

配列バインディングからの行数のリターン – パラメータ配列がバインドされる複数のDML文を実行する場合、ODP.NETはおのおのの配列入力値に影響する行の番号を返します。

グローバルゼーション・サポート

Oracle Database 12cではグローバルゼーション・サポートが強化され、Unicode®キャラクタ・セットでのデータベース構築や標準に準拠した多言語エンタープライズ・アプリケーションの開発を容易にする新機能が導入されました。オラクルは、どの言語の文字データも格納、処理、取得が可能なUnicodeベースのシステム・アーキテクチャを推奨しています。新しいデータベース機能が組み込まれたことで、ビジネス要件を満たす完全なUnicodeソリューションを構築する際の使いやすさと業界内での互換性が強化されました。

グローバルゼーション・サポートの新機能Unicode 6.1のサポート

Unicode規格は、世界で使用されているほとんどの表記法の文字をエンコードするための汎用キャラクタ・セットを定義しています。Unicodeを使用すると、プラットフォームやプログラミング言語に関係なく、テキスト情報を一様に表現できるようになります。Oracleは、Oracle 7からUnicodeキャラクタ・セットをサポートしています。Oracle Database 12cでは、Unicode規格バージョン 6.1までサポート範囲に含まれました。

Unicode照合アルゴリズム (UCA) への適合

Unicode照合アルゴリズム (UCA) は、Unicode文字列の語順を決定するためのUnicode規格です。UCAには、すべてのUnicode文字を合理的に照合するのに必要なデフォルトを指定するデフォルトUnicode照合基本テーブル (DUCET: Default Unicode collation Element Table)が定義されています。DUCETは、特定の言語が持つ特殊な語順に対応できるようにカスタマイズすることもできます。UCAは、照合に関する国際標準であるISO 14651と完全に互換性がありますが、拡張機能が用意されており、照合動作に柔軟性があります。

Oracle Database 12cでは、既存のデータベースの単一言語照合と多言語照合に加え、UCAのサポートも始まりました。オラクルのUCA実装はUnicode規格6.1に準拠しています。おもな機能は次のとおりです。

- Unicode 6.1 DUCET 23に基づく完全な照合順序
- 最大4つの照合レベルに対応した多段階比較アルゴリズム
- 可変重み付け文字（空白、句読点、記号）をソートするための構成可能なオプション
 - Blanked
 - Non-ignorable
 - Shifted
- スペイン語、伝統的スペイン語、フランス語(カナダ)、デンマーク語、タイ語、簡体字中国語など、言語に合わせた12種類の言語固有のUCA照合

言語的な操作を行う場合、相対的な順序を評価する前に、文字データは照合キーと呼ばれるバイナリ値に変換されます。Oracleでは照合キーがRAWデータ型で表現されます。Oracle Database 12cでは、VARCHAR2、NVARCHAR2、およびRAWの各データ型の最大長が32,767バイトまで拡張されたため、より長くより精度の高いテキストをソートできるようになっています。

新しいロケールの対象範囲

Oracle Database 12cでは、グローバルゼーション・サポートの対象範囲を拡大し、顧客要件の急速な変化に対応するための継続的な取組みの一環として、12の新しい言語と32の新しい地域のセットがデータベース・ロケールとしてサポート対象になり、対象範囲となるアジア、アフリカ、アメリカ、ヨーロッパの地域が増えました。

- 新しい言語 – アムハラ語、アルメニア語、ダリー語、ディベヒ語、クメール語、ラオ語、ボスニア語（ラテン文字）、マルタ語、ネパール語、ペルシャ語、シンハラ語、スワヒリ語
- 新しい地域 – アフガニスタン、アルメニア、バハマ、ベリーズ、バミューダ、ボリビア、ボスニア・ヘ



ルツェゴビナ、カンボジア、カメルーン、コンゴ・ブラザビル、コンゴ民主共和国、エチオピア、ガボン、ホンジュラス、イラン、コートジボワール、ケニア、ラオス、モルディブ、マルタ、モンテネグロ、ネパール、ナイジェリア、パキスタン、パラグアイ、セネガル、セルビア、スリランカ、タンザニア、ウガンダ、ウルグアイ、ザンビア

さらに、エチオピア暦（長さが5日または6日の13月があるコプト暦に基づく暦法）もサポート対象に含まれています。

Database Migration Assistant for Unicode (DMU)

Unicodeキャラクタ・セットへの移行は、時間がかかるうえにリソースの大量消費につながりかねない、さまざまな処理側面が多数含まれた複雑なプロセスです。途中に1つでも失敗があれば、データ損失が発生したり、ビジネスに深刻な影響を及ぼしたりする恐れがあります。Oracle Database Migration Assistant for Unicode (DMU) は、直感的なGUIで移行プロセス全体を効率化し、DBAの手動作業や意志決定を最小化する次世代の移行ツールです。移行に伴うあらゆる問題に事前にしっかりと対処し、データ変換を正しく効率的に実行するうえで、このツールが役立ちます。DMUを使った移行ワークフローには、次のプロセスが含まれます。

- リストアップ - 変換が必要なテキストデータを含むデータベース・オブジェクトを自動的に識別します。
 - スキャン - 移行の実現可能性を包括的に評価し、データに関わる潜在的な問題を洗い出します。
- クレンジング - 高度なツール・セットを使用して、データ分析とデータの安全性の確保に必要なクレンジングを繰り返し実行します。
- 変換 - データのインプレース変換が自動的に実行され、時間と領域に関わる要件が最小化されます。

DMUには移行後のデータベースの状態を継続的にチェックする機能があるため、Unicodeキャラクタ・セットへの移行に役立つだけでなく、Unicode規格に準拠したデータの維持にも有効です。Unicodeキャラクタ・セットを使用するデータベースを使用している、アプリケーションの構成が誤っていると、無効な文字コードがデータベースに取り込まれ、データが破損する恐れがあります。DMUの検証モード機能を使用すれば、問題の発生源を明らかにし、エンドユーザーに指摘される前にデータの問題を検出することができます。

DMUは無料でダウンロードできる製品として、2011年4月にOTNで初めてリリースされました。最新のDMUバージョン 1.2はOracle Database 12cにバンドルされており、データベースをUnicodeキャラクタ・セットに移行する方法として公式にサポートされています。DMUは、以前のOracle Database Release 10.2、11.1および11.2の一部の移行もサポートしています。従来のコマンドライン・ユーティリティであるCSSCANとCSALTERはサポート対象外となりました。

Oracle Database 12cで移行を実行することの利点は他にもあります。Unicodeに変換するとデータが拡大し、値が長くなる場合がありますが、VARCHAR2型の上限が新たに32,767バイトまで拡張されたことで、上限が4,000バイトであった12.1より前のVARCHAR2列は、いずれも列を長くするだけでこのようなデータを保持できます。こうした列については、切捨てもCLOBデータ型への移行も不要になりました。

JSON

アプリケーション開発者の中には、データベース・アプリケーションの構築にドキュメント中心（または‘スキーマ・オン・リード’）アプローチを取ることを好む人もいます。これは、アプリケーション・データをリレーショナル・スキーマに格納する代わりに、‘ドキュメント’（JSONオブジェクトやXMLオブジェクト）に

格納する方法です。

ドキュメント中心の永続化モデルでは、データの構造をドキュメントの内容で定義します。各ドキュメントは自己記述的であるため、これはアプリケーションのデータを格納するメカニズムとして極めて柔軟です。ドキュメントベースのアプローチでは、ベースとなるストレージ・モデルの変更が必要になっても、データベース・スキーマに新しい列を追加して欲しいとDBAに依頼せずに、新しいデータ・フィールドをアプリケーション開発者が追加することもできます。

つまり、アプリケーション・データ・モデルが変更された場合でも、データベース・スキーマの変更はないため、即座にアップグレードができるということです。そのため、適切な承認テストが完了したら、すぐに新しいバージョンのアプリケーションを本番環境に安全にデプロイでき、データベースのスキーマ変更を実施できるシステムのメンテナンス期間を待つ必要がなくなります。

ドキュメントベースの開発モデルをサポートするために、Oracle Database 12cにJSONのサポートが追加されたため、Oracle DatabaseをJSONドキュメント・ストアとして使用できるようになりました。JSONはVARCHAR列、CLOB列またはBLOB列を使用して格納されるだけであるため、JSONドキュメントの構造が変更された場合でもリレーショナル・スキーマを変更する必要はありません。

つまり、必要に応じて開発者が自由にJSONドキュメントの構造を変更できるということであり、JSONコンテンツの管理という点では、“noSQL”のドキュメント・ストア・データベースとまったく同じレベルの柔軟性をOracleデータベースで実現できます。

また、JSONは標準のデータ型を使用して格納されるため、Oracle DatabaseをJSONドキュメント・ストアとして使用する場合は、Oracle Databaseが備えるエンタープライズ・グレードの機能をすべて活用できます。つまり、JSONドキュメント・ストアとしてOracleを使用することにした組織は、スケーラビリティ、可用性およびパフォーマンスのレベルが保証されるということです。また、Oracle Databaseが持つ圧縮、レプリケーション、バックアップ、リカバリに関連する機能も使用できます。

オラクルが実施したJSON対応の強化は、ストレージ、問合せ、索引付けという3つのカテゴリに大きく分類されます。

- » ストレージへの対応としては、新しい制約、“IS JSON”が追加されています。この制約はVARCHAR2型、CLOB型またはBLOB型の列に適用できます。この制約を適用することで、列のコンテンツが有効なJSONドキュメントであることが保証されます。
- » 問合せへの対応としては、JSON_VALUE、JSON_QUERY、JSON_TABLE、JSON_EXISTS、JSON_TEXTCONTAINSという5つの新しいSQL演算子が追加されています。これらの演算子はSQL標準を拡張するように提案された演算子の基本となるものであり、宣言的なSQLのあらゆる機能を、格納済みのJSONデータに取り込むことを可能にするものです。これらの演算子を使用すると、Oracle Databaseに格納されているJSONデータをリレーショナル・データのように問合せたり分析したりできます。また、JSONコンテンツとリレーショナル・コンテンツを結合する問合せや、Oracle Databaseに格納できる他の種類のデータ（XMLデータや空間データなど）とJSONコンテンツを結合する問合せも生成できます。
- » 索引付けは、次の2つの方法で対処されています。1つ目として、オラクルの既存のファンクション索引付け機能を使用して、JSONドキュメント内の特定のキーにBツリー索引を作成できるようになりました。2つ目として、新しいJSON検索索引を使用することで、JSONの構造が事前にわかっていなくてもJSONドキュメントの完全な検索付けができるようになりました。前述した新しいSQL演算子のいずれかを含む問合せを最適化する場合は、これらの索引がOracleオプティマイザで自動的に使用されます。

Oracle Database 12cに組み込まれたJSONの機能はどれも、ドキュメントベースのストレージの基本原理に完全に対応しています。つまり、データベースで管理されているJSONドキュメントの構造定義をデータベース側で事前に把握していなくても、JSONコンテンツの格納、索引付け、問合せが実行されるのです。



JSONは標準のSQLデータ型を使用して格納されるため、JSONデータを操作する場合は、SQL文を使用してデータベース・コンテンツを操作できるAPIを、アプリケーション開発者はどれでも自由に使用できます。つまり、JAVA、C、.NETといった一般的なプログラミング環境からはもちろん、PHP、RUBY、PYTHON、PERLなどの一般的なスクリプト・フレームワークからも、JSONコンテンツの挿入、アクセス、更新ができるということです。

Oracle Database 12cをJSONドキュメント・ストアとして使用すると、エンタープライズ・グレードの機能でJSONドキュメントのデータを管理できるほか、noSQLベースのソリューションが抱えるもう1つの根本的な弱点、すなわちセキュリティにも対処できます。Oracleデータベースに格納されたJSONドキュメントは、リレーショナル・コンテンツに使用できるのと同じアクセス制御メカニズムとセキュリティ機能を使用して管理されます。たとえば、暗号化が使用されるため、IT組織は他のエンタープライズ・データと同様のセキュリティをJSONコンテンツにも確実に施すことができます。

結論

Oracle Database 12cには、Oracle APEX、PL/SQL、SQL、C、C++、Java、.NET、PHP、Perl、Python、Rubyといった特に多用される開発テクノロジーをサポートする生産性に優れた強力なアプリケーション開発ツール群が付属しています。Oracle MultitenantおよびOracle Database Cloud Serviceは、クラウド向けアプリケーションの開発を容易にします。トランザクション・ガードおよびアプリケーション継続性は、アプリケーションの可用性と信頼性を劇的に向上させます。

VARCHAR、NVARCHAR、およびRAWの各データ型が32Kに拡大され、自動増分（IDENTITY）列が追加されたことで、Oracle Databaseへの移行はかつてなく容易になりました。アプリケーションをOracle Databaseに移行するときに必要な作業量は、SQL変換フレームワークとMySQLアプリケーション向けドライバを使用することで大幅に減少します。移行を支援するその他の機能には、SQLからPL/SQLへのバインド処理の拡張、問合せの行数制限と行オフセットのネイティブSQLによるサポートがあります。

Oracle Application Expressはオラクル独自のWebアプリケーション開発ツールで、Oracle Databaseを使用するデスクトップ・アプリケーションやモバイル・アプリケーションの高速構築に最適です。Oracle SQL Developer Data Modelerは、論理モデルや概念モデル（Oracle Business Intelligenceの多次元モデルなど）、リレーショナル・データベース・モデルの作成をサポートし、最後の詳細な物理実装も支援します。Oracle SQL DeveloperはOracle Database向けのグラフィカルな作業環境で、データベース開発者の生産性を向上させるとともに、他のデータベースからOracle Databaseへの移行を容易にします。

Oracle Databaseの.NETコンポーネントは、Entity Frameworkや統合言語クエリーを含む最新の.NET機能はもちろん、Oracle Multitenantやトランザクション・ガードなどのOracle Database 12cの機能もサポートしています。

Oracle Database 12cの新しいグローバル化・サポート機能は、Unicode規格6.1に準拠したデータベース・アプリケーションの開発を可能にします。Unicode照合アルゴリズム（UCA）実装は、業界標準の多言語照合に対応しているほか、柔軟な機能も備えています。データベース・ロケールの対象範囲が拡張したことで、アプリケーション・ローカライゼーションのサポート対象もさらに拡大し、ローカル・ユーザーの文化的習慣に適合した動作が取られるようになりました。Oracle Database Migration Assistant for Unicode（DMU）は停止時間を大幅に短縮し、コストを低減し、データベースをUnicodeキャラクタ・セットへ移行する作業を簡素化します。

JSONがもっとも重要なデータ型としてOracleでサポートされるため、Oracle DatabaseをJSONドキュメント・ストアとして使用できます。そのため、アプリケーション・データの保存に使用する主要なツールとしてJSONを選択する次世代のアプリケーション開発者のニーズにOracle Databaseで対応できます。

ひとことで言うと、これらのツールを使用することで開発タスクが簡素化され、コストが減少し、アプリケーション開発期間の短縮を図ることができます。Oracle Database 12cには、セキュアで高可用性かつパフォーマンスの高いアプリケーションの作成を可能にする革新的な新機能が組み込まれています。

