

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 | 2017年1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

はじめに.....	5
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++)	6
はじめに	6
Oracle Sharding.....	6
Oracle Multitenant	6
SQL の新機能.....	7
自動増分 (IDENTITY) 列	8
行制限句.....	8
非表示列.....	8
パラメータとしての PL/SQL パッケージ型およびブール型	8
暗黙的結果.....	8
パフォーマンスとスケーラビリティ	8
SQL 計画管理 (SPM)	9
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.....	9
Global Data Services – 実行時ロードバランシング	9
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.....	9
アプリケーションの可用性と信頼性の最大化.....	10
共通 HA イベントと通知.....	10
新しい概念.....	10
トランザクション・ガード	10
Application Continuity	11
Oracle Global Data Services	11
使いやすさ	12
配列 DML の反復当たりの行数	12
クライアント側デプロイメント・ディスクリプタと自動調整.....	12
データベース操作の監視と追跡.....	12

インテリジェントな接続と無効なクライアントの迅速な検出	13
ID の長さを拡張	13
セッションの状態を修正するための PL/SQL コールバック	13
セキュリティ	13
データベースにおけるデフォルトの Java セキュリティのカスタマイズ	13
JDBC のセキュリティ強化	14
Oracle Advanced Security オプション	14
Java 標準	14
JDBC 4.1	14
JDK 1.7 および複数の JDK	15
JNDI	15
ロギング	15
アプリケーションの移行	15
SQL 変換フレームワーク	15
MySQL アプリケーション向けドライバ	16
その他のリソース	16
Oracle Database Cloud Service	17
SQL と PL/SQL	18
SQL と PL/SQL の重要な新機能	18
Oracle ネイティブの LEFT OUTER JOIN 構文の拡張	18
有効時間をサポートする表の定義	19
有効時間のテンポラル・フラッシュバック問合せ	19
SQL WITH 句での PL/SQL ファンクションの定義	20
PL/SQL から SQL へのインタフェースで使用できる PL/SQL 固有のデータ型	20
PL/SQL ユニットから参照できるデータベース・オブジェクトのリストの制限	20
起動者権限ファンクションによる結果のキャッシング	21
DIRECTORY オブジェクト・タイプによる LIBRARY オブジェクト・タイプの定義	21

新しい PL/SQL パッケージ UTL_CALL_STACK	21
新しい PL/SQL サブプログラム DBMS_UTILITY.EXPAND_SQL_TEXT	21
新しい事前定義済みの PL/SQL 問合せディレクティブ	21
ACCESSIBLE BY 句の機能強化	22
データ・バインドされた照合	22
PL/SQL 式の機能強化	22
PL/SQL における SQL JSON 演算子のサポート	22
より長い ID のサポート	23
コード・カバレッジ分析	23
DEPRECATE プラグマ	23
SQL 文に拡張された PL/Scope	23
DBMS_SQL への PL/SQL 専用データ型のバインド	23
Oracle SQL Developer	24
Oracle SQL Developer のコア領域	24
利点	24
Oracle Database 12c のサポート	25
Oracle Application Express	26
Oracle Application Express 5.0 のおもな機能	27
ページ・デザイナー	27
ユニバーサル・テーマ	27
インタラクティブ・レポート	28
グラフ機能	28
動的アクション	29
プラグイン	29
モーダル・ダイアログ	29
カレンダー	29
モバイル・レポート	29

パッケージ・アプリケーション	30
Web シート	30
Oracle REST Data Services	31
ORDS で REST API を構築する方法	32
Oracle Database 12c の新たな SQL/JSON 機能との連携	33
ORDS のアーキテクチャ	33
ORDS のセキュリティ	34
結論	35
JSON	36
.NET	37
.NET のおもな新機能	38
Application Continuity	38
Oracle Sharding	39
Oracle Cloud	39
ODP.NET 管理対象ドライバの強化機能	39
より長いスキーマ ID	40
グローバル化・サポート	40
グローバル化・サポートの新機能	40
Unicode 6.1 のサポート	40
Unicode 照合アルゴリズム (UCA) への適合	40
新しいロケールの対象範囲	41
Database Migration Assistant for Unicode (DMU)	42
結論	43

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 Sharding は Oracle Database 12c で導入され、この機能を使用すると、データベースを複数のサーバー間でスケールアウトできます。

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、SQL Developer、ORDS、PL/SQL、SQL、C、C++、Java、.NET、JSON、Node.js、PHP、Perl、Python、Ruby、Go、R 言語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・データにもたらし、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++)

はじめに

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。スクリプト言語ドライバは、これらの機能の多くから即座に恩恵を受けられます（クライアント API が必要ないいくつかの機能は、ドライバ・メンテナによるサポートの追加状況に依存します）。これらのドライバには、Node.js 用の node-oracledb、PHP の OCI エクステンションと PDO_OCI エクステンション、Ruby の ruby-oci8 エクステンション、Python の cx_Oracle ドライバ、Perl の DBD::Oracle、R 言語の ROracle ドライバのような OCI ベースのドライバがあります。Java の JDBC を使用する Jython や JRuby のような言語もメリットを得られます。

Oracle Sharding

Oracle Sharding は、1 つ以上のローカルまたはグローバルなデータセンターに配置された低コストの汎用データベース・サーバーに対し、データベースをほぼ線形にスケーリングできるシェアード・ナッシング・アーキテクチャです。シャーディングのメリットには、グローバルなデータ分散（特定のデータを消費者の近くに保管）、障害の封じ込め（1 つのシェアードに障害が発生しても、他のシェアードの可用性には影響しない）があります。Oracle Global Data Services はシェアード間のデータの位置を管理し、アプリケーション・リクエストをこの分散データベース・システムの適切なシェアードに送ることを可能にします。

Oracle Multitenant

12c の Multitenant は、優れた効率性と優れた敏捷性をもたらします。これらは両方とも、このユースケースでは重要な機能です。

高度な自動化により、初代クラウドに関連するテクノロジーで、このコスト・モデルをほぼ線形の機能に改良することができます。開発環境を例にして考えてみましょう。10 人の開発者をサポートするには 10 台の仮想マシン (VM) が、20 人の開発者をサポートするには 20 台の VM が必要になります。それほどのワークロードがかかっても、本質的に VM ごとにコストが発生します。そのため、コストと規模の間には線形の関係があります。とはいえ、これは正しい方向に進むうえで重要なステップでした。Oracle Multitenant は次世代の Database Cloud のためのアーキテクチャです。Oracle Multitenant は現状を一変させ、真のスケールメリットを実現します。プラガブル・データベース (PDB) がデータベースを搭載した VM というコストのかさむモデルに取って代わります。PDB 固有のコストはほとんどないため、各開発者の PDB のコストは実際の作業量の分だけで済みます。すべての開発者の PDB を 1 つのマルチテナント・コンテナ・データベース (CDB) に統合することができ、その CDB を実行するコストはすべての開発者で共有できます。コンピューティング・リソースの観点から見ると、バックグラウンド・プロセスが 1 セット、単一の共有メモリ領域 (SGA) が 1 つだからです。管理の意味では、バックアップ、高可用性の構成、パッチの適用などが必要な CDB は 1 つです。

現代の革新的で生産性が高い開発およびテスト組織は、優れた敏捷性を求めています。アプリケーションのデータベース・コンポーネントの場合、この敏捷性を達成するには、データベースを極めて効率的にプロビジョニングする機能が必要です。12.2 の Oracle Multitenant は、いくつかの強力

なオンライン・プロビジョニング機能をサポートします。PDB のプロビジョニングは、クローニングというプロセスによって達成されます。クローニングには、完全クローン、サブセット・クローン、そしてスナップショット・クローンのシン・プロビジョニング（Copy-on-Write テクノロジーを利用）など、多くの種類があります。プラグブル・データベース（PDB）は、ソース側で停止時間を発生させることなくクローニングできます。非常に大規模な本番データベースの“開発マスター”クローンに対してもソースにほとんど影響を与えずにシンプルかつ効率的に最新のデータを追加できる増分リフレッシュ機能がサポートされています。PDB のオンライン再配置は、アプリケーションの停止を必要としない、クラウドへのシンプルな移行を実現します。

Software as a Service（SaaS）への対応として、12.2 の Oracle Multitenant にはアプリケーション・コンテナという強力な新機能が導入されています。これにより、DBA だけでなくアプリケーション管理者も、多数のテナントを一括管理できるというメリットを享受できます。オンプレミスでのスタンドアロン・デプロイメント用として過去に設計されたアプリケーションの場合は、アプリケーション・コンテナを即席のクラウド・アーキテクチャとして使用できます。クラウド生まれのアプリケーションの場合は、SaaS の特徴である優れたスケールメリットを得るために敏捷性を犠牲にしてきました。アプリケーション・コンテナは、こうしたスケールメリットを損なうことなく、PDB の敏捷性を完全に達成します。マスター・アプリケーションの定義を、アプリケーション・ルートという単一コンテナにインストールできるようになりました。このアプリケーション・コンテナの個々のテナントはこのマスター定義によって対処されます。その結果、アプリケーション・メンテナンス操作（アプリケーション・パッチの適用など）を中央ロケーションで実行でき、個々のテナントをそれぞれのスケジュールで、この新しいバージョンとシンプルに同期できます。強力なコンテナ間集計機能を使用することが可能になり、ISV はダッシュボードを簡単に作成して、現在の四半期の注文数や総収益などの主要メトリックスを使って、アプリケーションの全体的なパフォーマンスを評価できます。これらの機能は ISV のためだけにあるものではありません。世界中の複数の子会社で同じビジネス手法を実践しているグローバル企業やフランチャイズなど、その他多くのビジネス・モデルでも同様に利用できます。

JDBC、SQLJ、OCI、OCCI の各ドライバはプラグブル・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。

SQLの新機能

次の新しい SQL の機能を Java、C、C++アプリケーションで利用できます。32 K の VARCHAR、NVARCHAR、および RAW データ型、非表示列、暗黙の結果、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。スクリプト言語はこれらの主要言語で作成されるので、新しい機能を利用できます。

32 K 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、4000 バイトから 32767 バイトに増加しました。JDBC と OCI/OCCI 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、32 K よりサイズの小さいラージ・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。

行制限句

Oracle Database 12c では、SELECT の OFFSET 句と FETCH NEXT 句が導入されているため、ネストした問合せ構文を作成しなくても、問合せレコードのサブセットを容易に取得できます。たとえば、Web アプリケーションによるデータのページ付けが必要な場合、正しいシンプルな SQL 問合せをより簡単に作成できます。

非表示列

表に作成または追加した新しい列を、後から INVISIBLE SQL 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 構文の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 に移行できます。

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、Global Data Services を使用した遠隔地間の実行時ロードバランシングがあります。

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

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 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 を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 の 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することができます。

Global Data Services – 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 64 K まででしたが、このリリースでは最大 2 MB 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 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、Net Services のポータルにあるホワイト・ペーパー『Advanced Network Compression』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、Application Continuity、Global Data Services 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、Oracle Notification Service を使用してサブスクリバ（ドライバ、アプリケーション）に転送および通知されます。

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/OCCI 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、SYSTEMTIMESTAMP、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がわ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データベース。
- 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、OCIC、および ODP.NET で使用可能。

詳しい説明とコード部分については、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。

Application Continuity

Oracle Database 12c の Application Continuity により、C または Java アプリケーションで Oracle RAC および Oracle Active Data Guard を使用する際、何も設定しなくても、“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。Application Continuity 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. データベース・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、正常なデータベース・インスタンス (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。

Application Continuity 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をマスク
- 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

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 Application Continuity 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 Application Continuity 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。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。詳細については、My Oracle Support Note 1364193.1 を参照してください。

Application Continuity 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『Application Continuity』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。

Oracle Global Data Services

Oracle Universal Connection Pool は、遠隔地間の高速接続フェイルオーバーと Application Continuity をサポートします。Global Data Services の詳細については、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が多数導入されています。配列 DML の反復当たりの行数の検出、クライアント側デプロイメント・ディスクリプタ、自動調整、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などの機能があります。

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/OCCI 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調整

このリリースでは、Java EE の場合と同様に、OCI/OCCI 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/OCCI 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 Instant Client 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。

セッション・プールでは OCI のユーザーのために、無効な接続の検出、クリーンアップ、アイドル・セッションの再確立も向上しています。これにより、ネットワークが停止した後のアプリケーションの信頼性が向上します。

IDの長さを拡張

Oracle Database 12.2 の ID では ID の長さが拡張されており、OCI アプリケーションと Java アプリケーションもこの長さの ID を利用できます。ID は 128 バイトになっています。

セッションの状態を修正するためのPL/SQLコールバック

OCI セッション・プールのユーザーは、セッションに複数のプロパティで'タグ付け'することができるようになりました。アプリケーションはこれらのタグ・プロパティを使って、どの状態がセッションに設定されているのかを示すことができます。

アプリケーションがセッション・プールのセッションをチェックアウトする場合に、取得したセッションに、必要なすべてのタグが設定されていないときは、必要な実際のタグで PL/SQL コールバックを起動できます。PL/SQL コールバックは、必要なセッションの状態の修正ロジックをどこでも実行できます。修正は自動的に実行されるため、アプリケーション・ロジックが簡素化されます。修正はサーバー上で実行されるため、パフォーマンスが向上します。

セキュリティ

データベースにおけるデフォルトの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る個々のメッセージにメッセージ・ダイジェストを含めます。

JDBC では、SHA-2 ハッシュ・アルゴリズム (SHA-224、SHA-256、SHA-384、SHA-512 など) が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標準

Oracle Database 12c は、JDBC 4.1、JDK 1.7、ロギング、および JNDI といった Java 標準も提供します。

JDBC 4.1

Oracle Database 12c 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るようになりました。getObject(int parameterIndex, java.lang.Class<T> type)

throws SQLException

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

throws SQLException

```
Example 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 で使用できませんが、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 Oracle Database 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 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可能。

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 の 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 Oracle 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の詳細については、次の URL を参照してください。OCI：
<http://www.oracle.com/technetwork/jp/topics/index-094133-ja.html>

Oracle Java、JDBC、UCP、データベース Web サービス：
<http://www.oracle.com/technetwork/jp/database/application-development/java/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の観点から言うと、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・インターフェースです。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をもたらされます。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 の詳細については、

<https://cloud.oracle.com/mycloud/f?p=service:database:0> にアクセスしてください。

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、実装を隠すやり方）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 規格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 値が含まれます。

ANSI 規格には、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可能。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るため、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 (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 (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 (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のファンクションとして使用することが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使用できる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、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、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・オブジェクトのリストの制限

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できるため、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 11g 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オブジェクト・タイプによるLIBRARYオブジェクト・タイプの定義

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が、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。

ACCESSIBLE BY句の機能強化

ACCESSIBLE BY 句は、サブプログラムを起動しても安全と見なされる PL/SQL ユニットのリストを指定し、他をブロックします。

Oracle Database 12c Release 2 (12.2) 以降、アクセッサ・リストはパッケージの個々のサブプログラム上で定義できます。パッケージ自体（存在する場合）で定義されたアクセッサ・リストに加えて、このリストがチェックされます。このリストは、サブプログラムへのアクセスの制限しかできず、アクセスを拡張することはできません。このコード管理機能は、内部サブプログラムの不適切な使用を防ぐ上で役立ちます。たとえば、1つのパッケージを2つのパッケージ、つまり、1つはアクセス制限が必要な少数のプロシージャ用のパッケージ、もう1つはパブリック・アクセスが必要な残りのユニット用のパッケージに再編成することは有益ではない可能性があります。

データ・バインドされた照合

照合（ソート順ともいいます）は、2つの文字列を比較しソートするときに、文字がもう一方の文字列に対して等しいか、優先するか、または従うかを決定する一連のルールです。

Oracle Database 12c Release 2 では、デフォルト照合という新しいプロパティが表、ビュー、マテリアライズド・ビュー、パッケージ、ストアド・プロシージャ、ストアド・ファンクション、トリガー、型に追加されています。ユニットのデフォルト照合は、列、変数、パラメータ、リテラル、戻り値などのデータ・コンテナの照合を決定し、そのユニットに明示的な自身の照合宣言がない値を返します。

PL/SQL式の機能強化

Oracle Database 12c Release 2 (12.2) 以降、これまでリテラル定数しか許可されていなかった宣言で式を使用できるようになっています。静的な式もサブタイプの宣言で使用できるようになりました。

PL/SQL 開発者にとって、拡張した式と一般化した式はおもに2つの主要な利点があります。(1) 環境の変化に応じてプログラムをはるかに容易に適応できます。(2) プログラムがよりコンパクトで明確になり、非常にわかりやすく、保守しやすくなりました。

PL/SQLにおけるSQL JSON演算子のサポート

この機能により、Oracle Database に保管された JSON ドキュメントで作業し、リレーショナル・データから JSON ドキュメントを生成することが容易になります。

データベースへの JSON ドキュメントの保管と問合せに対する Oracle Database のサポートは、SQL を使ってリレーショナル・データから JSON ドキュメントを宣言的に生成する機能、JSON ドキュメントを PL/SQL オブジェクトとして操作する機能など、新しい機能の追加によって拡大されています。SQL JSON 演算子は、いくつかの例外を除き PL/SQL でサポートされています。

より長いIDのサポート

PL/SQL で使用、定義されるすべての ID の最大長が、以前のリリースでは 30 バイトだったのが 128 バイトに拡張されました。

COMPATIBLE パラメータを 12.2.0 以上の値に設定する場合、データベース・キャラクタ・セットの ID の表示を 128 バイト以下にする必要があります。COMPATIBLE パラメータを 12.1.0 以下の値に設定する場合、制限は 30 バイトです。

コード・カバレッジ分析

プログラム・ユニットのどの行が特定のコード実行で実行されるのか、コード・カバレッジに関するデータが PL/SQL によって収集されるようになりました。DBMS_PLSQL_CODE_COVERAGE パッケージにより、基本ブロック・レベルでデータを収集することが可能になります。PL/SQL の開発者とテスト・エンジニアは標準的な品質保証メトリックの一部としてコード・カバレッジのテスト結果を使用できます。

DEPRECATE プラグマ

PL/SQL の開発者は DEPRECATE プラグマを使ってプログラム・ユニットや個々のサブプログラムを非推奨とマーク付けできるようになりました。非推奨に対応するために他のコードを変更する必要がある非推奨要素がコンパイラによってユーザーに警告されます。

SQL文に拡張されたPL/Scope

ID の使用に関する情報を収集して、プログラム・ユニットの詳細なインパクト分析を可能にする PL/Scope が、PL/SQL プログラム・ユニットに埋め込まれた SQL 文の情報も収集するようになりました。ネイティブの動的な SQL と SQL ヒントの使用も識別します。

DBMS_SQLへのPL/SQL専用データ型のバインド

PL/SQL 専用データ型の値を SQL でバインドできる機能（12.1 で EXECUTE IMMEDIATE 用に初めて実装）が DBMS_SQL に拡張されました。

PL/SQL の詳細については、<http://www.oracle.com/technetwork/database/features/plsql> にアクセスしてください。

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 ユティリティが用意されているほか、Recovery Manager、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。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 - 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、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 への移行については、

<http://www.oracle.com/technetwork/jp/database/migration/index.html> にアクセスしてください。

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。ユーザーは Oracle SQL Developer を使用して、Oracle Express Edition など、サポートされている任意の Oracle Database に接続できます。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。

SQL Developer もコマンドライン・インタフェースを介して利用できるようになりました。Oracle SQLcl は、SQL*Plus の機能をすべてサポートしますが、以下も提供します。

- » SQL 履歴
- » コード・アシスタンス。<TAB>を押すと、キーボードと表、ビュー、列名の完成を支援
- » CD、CTAS、DDL、REPEAT などの新コマンド
- » JavaScripting

JRE を実行するだけで SQLcl を使用でき、ダウンロードのサイズは 20 MB 未満です。

Oracle Database 12cのサポート

Oracle SQL Developer Release 4.1 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、調整、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 の詳細については、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、外観がきれいで応答性に優れた、高速でセキュアなデータベース駆動型デスクトップ・アプリケーションとモバイル・アプリケーションを開発、設計、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。
- **パッケージ・アプリケーション** - 数回クリックするだけで、幅広いポイント・ソリューションのコレクションをインストールできます。

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 の詳細については、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 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部門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 5.0のおもな機能

Oracle Database 12c Release 2 に付属する Oracle Application Express 5.0 には、次のおもな機能が 있습니다。

ページ・デザイナー

ページ・デザイナーは、極めてパワフルで直感的な最新のブラウザベースの IDE です。開発者の生産性を大きく向上させるように設計された完全に新しい IDE で、この IDE を使用すると、Application Express アプリケーションを非常に迅速に開発、保守できます。ページ・デザイナーは視覚的表示やアプリケーション・ページを見やすくし、直感的なドラッグ・アンド・ドロップを介してまったく新しい方法でページをその場で開発することを可能にします。強化されたこのコード・エディタは、SQL および PL/SQL のインライン・エラーの検証、オート・コンプリート、構文の強調表示、regex サポートによる検索と置換の機能を搭載し、UNDO と REDO をサポートします。

ユニバーサル・テーマ

ユニバーサル・テーマは、Application Express 5 用に最初から開発されたアプリケーション向けのまったく新しいユーザー・インタフェースです。過剰なテンプレートを排し、組込みの Theme

Roller や Template Options を使った容易なカスタマイズを実現する、よりシンプルでありながら機能性に優れたテーマです。ユニバーサル・テーマは、HTML、CSS、または JavaScript の専門知識がなくても、応答性の高い高度な最新アプリケーションを構築できるように開発者を支援するためのツールです。さまざまな画面サイズやフォーム・ファクタでスムーズに動作する、応答性に優れたアプリケーションを構築できます。ユニバーサル・テーマには、複雑なフォーム・レイアウトの作成や柔軟なグリッドの使用など、主要な強化機能が含まれています。

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方が、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。

インタラクティブ・レポートは、開発者とエンドユーザー双方の機能を強化するために、Application Express 5 で最初から再構築されています。Application Express 内の他の領域と同様に、インタラクティブ・レポートの領域も CSS を使ってスタイルを変更できるようになりました。1 つのページに 1 つのインタラクティブ・レポートしか作成できなかった以前の制限がなくなって、これからは 1 つのページにいくつでもレポートを定義できます。各レポートはカスタマイズ可能な個別の領域です。エンドユーザーは、インタラクティブ・レポート内で直接ピボット・レポートを定義できるようになりました。エンドユーザーは、Application Express からデータをダウンロードして Excel にインポートする代わりに、データ上でピボットを実行するように、レポート内の列にピボットを宣言的に定義できるようになりました。

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。

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、プロセス、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。

モーダル・ダイアログ

標準ページ・プロセスを利用してモーダル（および非モーダル）ページを容易に定義できるようになりました。JavaScript を使い、大量の手動作業を行ってページを細かく設計する必要はありません。表示タイプと適切なテンプレートを設定し、あとは Application Express に任せるだけです。

カレンダー

新しいカレンダー・コンポーネントには、月、週、日、アジェンダ・ビューのサポートが組み込まれており、カスタマイズが格段に簡単になりました。人気の高い FullCalendar ライブラリをベースにし、ドラッグ・アンド・ドロップ、時間ベースのイベントをサポートし、応答性も備えています。

モバイル・レポート

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、どのモバイル機器でもモバイル・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。

Application Express 5 では、リフロー表や列トグルを使ってレポートを作成して、モバイル機器にすべてのデータを表示できるようになりました。リフロー表は、各列を折り返すか、非常に小さい画面に複数の行が表示されるように変更します。列トグルを使用すると、もっとも重要な列、および小さい画面で必要に応じて非表示にする列を指定できます。

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 を記述したりすることができます。

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。

Oracle REST Data Services

Oracle REST Data Services (ORDS) は、Oracle Database 用の RESTful サービスに対応する中間層テクノロジーです。ORDS により、先端をいく今日のアプリケーション開発者にとって使いやすい RESTful API サービスに Oracle Database を変換できます。ORDS は REST リクエストを受信して、SQL または PL/SQL コードを実行する Oracle Database にこれらのリクエストをマーシャリングし、出力を JSON コレクションとして呼び出し側アプリケーションに返します。ORDS を使用すると、どのようなデータベース・リソースも REST 経由で利用可能になります。

データ・アクセス API は、Oracle データベース・テクノロジーの全パワーをフルに利用して、最高レベルの信頼性、セキュリティ、パフォーマンスを達成します。アプリケーション開発者はこれらのデータ・アクセス API を使用し、10 年以上にわたって新興企業の現場（新人開発者が研修を受ける大学、Google や Facebook などの先進的な Web 企業、もっとも先進的な主流エンタープライズ IT 部門）で主流になっているものと同じ手法と技法を採用してエンタープライズ・クラスのアプリケーションを構築できます。

ORDS と REST により、開発者は次のことを実現できます。

1. 使用したいプログラム言語を選択する

何年も前、先進的な IT 部門はすべてを Java か .Net で構築していました。しかし、もうそんなことはなくなりました。今日の世界ははるかに多様化しています。JavaScript は現在、Java と同様に幅広く使用されています。Python、PHP、Ruby などの他の言語もかなりの人気を博しつつあります。Go from Google や Swift from Apple などの新しい言語は市場シェアを大きく獲得しました。アプリケーション開発言語のこの多様な世界をサポートする API を 1 つ持つことは非常に重要なことであり、新しい多言語およびマイクロサービスのアーキテクチャに対応する上で必要なことでもあります。

2. クライアント・ドライバや複雑な専用ネットワークを使用せずに、母国語の機能を使用する REST は、全プログラミング言語に対してネイティブです。

SQL アクセスとは違って、インストールや保守を必要とする特別なドライバ、あるいは格闘しなければならない複雑な占有ネットワークはありません。REST はインターネット・テクノロジー、特に、今日すべてのプログラミング言語でサポートされている HTTP に完全に基盤を置いています。

3. 他の外部サービスへのアクセスと同じ方法で、データベースにアクセスする

REST は今日、プログラム外部のサービスへのアクセス手段としてアプリケーション開発者が利用している圧倒的な主流テクノロジーです。ORDS と REST を使用すると、まったく同じ方法でデータベースにアクセスできます。IT 企業は特別なクライアント・ドライバ、占有ネットワーク、または SQL の厳しい研修を追加で実施しなくても、大学、新興企業の現場、REST のみを使っている多くの Web 企業、あるいは先進的なエンタープライズ IT 企業のアプリケーション開発者を即座に利用できます。また、こうしたスキルを学びたくない開発者から受ける抵抗に対処する必要もありません。

4. JSON を使用してプログラム言語と俊敏な開発を容易に統合する

プログラミング言語でデータベースを操作することはこれまで容易ではありませんでした。一般に"インピーダンス不整合"と呼ばれ、SQL を使ってデータベースのリレーションをプログラミング言語のオブジェクトにマッピングすることは困難でした。Oracle Database 12c の ORDS と新しい SQL/JSON ファンクションは、プログラミング言語オブジェクト間のマッピングを容易に行う JSON でアプリケーション・プログラムとデータベース間の全データを交換できるようにして、この問題を排除しました(JSON は、JavaScript によって使用されるオブジェクト表記法で、他のプログラミング言語で使用されるオブジェクト表記法に非常によく似ています)。

Oracle Database 12c では、データを JSON としてデータベース内に保管することもできます。そのため、開発者は新しい敏捷な"スキーマレス"または"スキーマラスト"開発手法を採用して、小さい開発者チームはより独立して安全に操作を行い、新しい機能をより迅速に小単位でリリースできます。

5. すべてを最初にクラウドで構築する

ORDS と REST を使用して、すべてを最初からクラウドに構築できます。REST は HTTP プロトコルを使用します。HTTP プロトコルはインターネットの中核をなし、インターネットに簡索性、信頼性、スケーラビリティをもたらす主要な"ステートレス"テクノロジーです。アダプタは不要です。トンネリングもありません。インターネットに対して完全にネイティブです。ORDS と REST は、オンプレミスと混合ハイブリッド・クラウド・デプロイメントにも完全に適しています。

ORDSでREST APIを構築する方法

ORDS は、個々の表とビューへの基本的なデータ・アクセス REST API を自動的に生成したり、ストアド・プロシージャの実行や複数の表にネストされたデータへのアクセスなど、より高度な API を手動で生成したりするためのツールを提供します。

URI を定義して REST リソースを特定し、リクエストを処理する SQL や PL/SQL ハンドラ・ルーチンへの GET (問合せ)、POST (挿入)、PUT (更新) といった REST HTTP コールをルーティングするツールなどがあります。パラメータはさまざまな方法によってハンドラ・ルーチン間で渡されます。そのような方法には、JSON を使った方法、または POST や PUT などコール本体の画像といったデータ・オブジェクト用の他のフォーマットがあります。また、パラメータ・コールは、問合せ文字列やルート・パターンを使って URI で渡されます。結果は JSON で、または選択した場合は CSV 形式で、呼び出し側のアプリケーションに戻されます。



図1 : ORDSの機能

ORDS はハンドラを自動生成して、単一の表やビューで基本的な CRUD 操作を実行します。それらのツールには、すべての行の問合せ、個々の行の問合せ、単一行の挿入、一括挿入、更新、メタデータの返却といった機能が含まれます。

ハンドラは、SQL または自律型 PL/SQL ブロックを使って手動で定義して、より高度なファンクションを実行することもできます。これには、入出力の変数が指定されたストア・プロシージャのコールが含まれます。また、Oracle Database 12c の SQL/JSON ファンクションを使った、階層リレーション間でのネストされた JSON ドキュメントのマッピングも含まれます。

Oracle Database 12cの新たなSQL/JSON機能との連携

ORDS REST API ハンドラは、Oracle Database 12c で新たに利用可能となった SQL/JSON 機能を最大限に活用できます。例えば、ハンドラは `json_table` ファンクションを使用して、発注書項目のネストされた配列を含む JSON の発注書を、`PurchaseOrder` 表と `PurchaseOrderItems` 表に挿入できます。リバース操作の場合、`json_object` ファンクションと `json_arrayagg` ファンクションを使い、`PurchaseOrder` 表と `PurchaseOrderItem` 表を問い合わせることで、発注書項目のネストされた配列を含む JSON の発注書オブジェクトを返すことができます。

また、ORDS と REST、Simple Oracle Document Architecture (SODA) プロトコルを使って、Oracle Database 12c の個別の JSON ドキュメント・ストア・ファシリティにある Oracle Database 12c に保管された JSON にアクセスすること、または通常のリレーションや表内の列の JSON にアクセスすることが可能です。IS JSON ファンクションなど、他の多くの便利な SQL/JSON ファンクションも使用できます。データベースに保管された JSON が適切な JSON フォーマットになるように、これらをデータベース制約として使うことができます。

ORDSのアーキテクチャ

ORDS は、Java アプリケーション・サーバーで実行される Java アプリケーションです。Jetty を使用してスタンドアロン・モードで実行されるため、開発やテストに最適です。強力なアプリケーション・サーバー管理機能が不要な本番デプロイでも使用できます。要求がより厳しい本番デプロイでは、ORDS は WebLogic、Tomcat、または GlassFish で実行されます。

ORDS は Oracle Database 10g Release 2 以上をサポートします。また、Oracle Database 12c の JSON ドキュメント・ストアを使用した SODA プロトコルもサポートします。ORDS はまた、Oracle NoSQL Database に REST API サービスを提供するためにも使用されます。

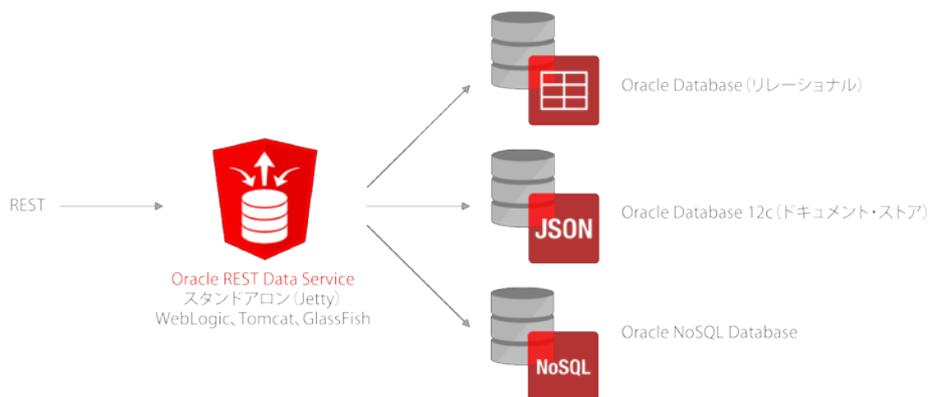


図2 : ORDSのアーキテクチャ

ORDSのセキュリティ

セキュリティについては、ORDS では REST API へのアクセスを保護するための IETF 標準である OAuth2 がサポートされます。これには、サード・パーティのアプリケーション向けにサーバーからサーバー（2 脚）、エンドユーザーからアプリケーション（3 脚）という双方の形式が含まれます。また、リクエスト元が REST サービスと同じ第一者認証もサポートされます。さらに、Identify Manager、シングル・サイン・オン・メカニズム、その他のファシリティを使用した外部認証もサポートされます。

ORDS は、WebLogic や Tomcat などのアプリケーション・サーバーで実行されるので安全でもあります。これらのエンタープライズ・クラスのアプリケーション・サーバーが提供するセキュリティ機能によって保護されます。

また、ORDS は、クライアント・アプリケーションからデータベースへの直接アクセスを許可しません。クライアント・アプリケーションは、高いレベルのデータベース・スキルを持つ、特別に認定されたデータ・アクセス開発者によって、API を確実に保護するために定義された REST API にしかコールを送ることができません。

結論

ORDS により、先端をいく今日のアプリケーション開発者が使用を望む RESTful API サービスに Oracle Database を変換できます。SQL などのデータベース・スキルを持つ開発者は、以下を達成する、データベースへのデータ・アクセス REST API を構築できます。

- » 高度なデータベース機能をフル活用して、最大限の信頼性、セキュリティ、パフォーマンスを達成
- » 単一の表やビュー用に自動的に生成されるか、より高度なファンクション用に手動で容易に生成することが可能
- » JSON を使用して、アプリケーションとデータベース間のデータ交換に対応
- » データベースに保管された JSON にアクセスして、アジャイル開発手法の使用を促進
- » OAuth2 や他の高度なセキュリティ・メカニズムで保護
- » 完全にクラウド対応

アプリケーション開発者は、任意のプログラミング言語を使ってネイティブにこれらのデータ・アクセス REST API を使用できます。クライアント・ドライバや複雑な占有ネットワークは不要です。アプリケーション開発者は、SQL や他の高度なデータベース・スキルを身につける必要はありません。REST や JSON を使って他の外部サービスにアクセスする場合と同様に、これらのデータ・アクセス API をコールできます。

ORDS の詳細については、Oracle Technology Network の ORDS の Web ページ ([こちら](#)) を参照するか、Google で“ORDS”を検索してください。これらの Web ページの“Downloads”タブ・セクションでは製品をダウンロードして試用でき、“Documentation”タブ・セクションには ORDS の全ドキュメントが掲載されています。“Community”タブ・セクションには ORDS フォーラムがあるので、技術的な質問をして答えを得ることができます。また ORDS 製品マネージャーや上級 ORDS 開発者による ORDS 関連トピックの各種ブログへのリンクもあります。“Learn More”タブ・セクションには、強くお勧めする ORDS Hands-On Lab (HOL) と YouTube での ORDS Video Channel へのリンクがあります。ORDS HOL では、主要な ORDS 機能を段階的に学習できる演習など、ラップトップで ORDS を試すために必要な VM とすべてのソフトウェアをダウンロードできます。ORDS YouTube Video Channel には現在、ORDS の概要ビデオ、インストール、セキュリティ、自動インポート、複数のデータ・ストア・サポートなどおもな ORDS のトピックに関するいくつかの短いビデオ (10 分未満のビデオ) があります。

JSON

現代のアプリケーション開発者は、リレーショナル・ストレージでサポートされるエンティティ・リレーションシップ・モデルを使う方法ではなく、アプリケーション・データをドキュメントとして保管する方法を選んでいきます。この切り替えをおもに推進しているのは、JSON や XML ベースのストレージの使用がもたらす柔軟性です。この柔軟性により、アプリケーション開発者はビジネスのニーズにより素早く対応できるようになりました。アプリケーション・データ・モデルが変更されても、データベース・スキーマを変更する必要がなくなったためです。その結果、アプリケーションをはるかに速いサイクルでデプロイし、更新することができます。ドキュメントベースの永続性への切り替えが、データ永続性に対応する NoSQL ドキュメント・ストアの採用につながりました。

Oracle Database 12c は、この種のアプリケーション開発を完全にサポートするように設計されています。Oracle Database 12c の一部として導入された Simple Oracle Document Architecture (SODA) 仕様は、Oracle データベースを JSON ドキュメント・ストアとして使うことを可能にする非常にシンプルな API を記述します。SODA API は、ドキュメント・コレクションの作成とドロップ、ドキュメントへの Create、Retrieval、Update、Delete (CRUD) 操作、ドキュメント・コレクションへのリストおよび例示による問合せ (QBE) 操作、一括挿入や索引付けなどの各種補助操作をサポートします。SODA により、アプリケーション開発者は、SQL、JDBC の知識、Oracle DBA の支援がなくても、JSON ドキュメントを使ってデータを管理するアプリケーションを開発し、デプロイできます。SODA の導入に加え、データベース自体が JSON の有効性の適用、JSON コンテンツの索引付け、これらの索引を使った JSON コンテンツへの操作の最適化を行うことができるようになりました。

SODA を選択すると、オラクルのデータ管理プラットフォームの利点を失うことなく、JSON ベースの永続性の全利点を得られます。組織は、複数のデータベース管理の複雑さを招くことなく、NoSQL 式開発を採用できます。引き続き Oracle Database を利用して、高い信頼性、スケーラビリティ、セキュリティ、リカバリを得られます。

Oracle Database 12c を NoSQL 式 JSON ドキュメント・ストアとして使うことで得られるもう 1 つの大きな利点は、必要なときに SQL のパワーをすべて利用できることです。アプリケーション開発者は SQL の知識がなくても、例示による問合せ手法を使ってアプリケーション・データを問い合わせることで、アプリケーションを開発し、デプロイすることができます。ただし、アプリケーション開発者が想定した以外の方法 (非定型の問合せ) でキャプチャしたデータを使ったり、JSON ドキュメントに保管された情報の報告や分析を実行したりするときには、Oracle Database 12c では、そのような目的のために SQL を使用できます。

Oracle Database 12c では SQL 言語が拡張されているので、JSON ドキュメントを SQL 操作の一部として問い合わせることができます。これらの拡張により、シンプルかつ単純な方法で SQL の全パワーを JSON ドキュメントのコンテンツに適用できます。また、JSON ドキュメント間の結合操作、Oracle Database によって管理される他のあらゆる種類のコンテンツ (リレーショナル・データ、XML コンテンツ、空間コンテンツ、セマンティック・コンテンツ、テキスト・コンテンツなど) と JSON ドキュメント間の結合操作も実行できます。

Oracle Cloud 向け Oracle Database 12c Release 2 では、素晴らしい新機能、Oracle Data Guide for JSON も導入しており、データベースが管理している JSON ドキュメントの構造を理解しやすくします。Oracle Data Guide for JSON は JSON ドキュメントの構造を動的にたどるので、ユーザーは JSON ドキュメントのリレーショナル・ビューを容易に生成でき、その結果、JSON を知らないプログラマーやツールでも、JSON ドキュメントで直接作業を行うことができます。

Oracle Database 12c Release 2 は以下のサポートを追加することで、データベースの JSON へのサポートを拡張します。

- » JSON ドキュメントをリレーショナル・データから直接生成
- » JSON ドキュメントへの部分更新操作により、プログラマーは JSON ドキュメントの特定部分の内容を変更することが可能
- » Oracle TimesTen In-Memory Database と Oracle Exadata により、JSON ドキュメントへの問合せ操作を最適化
- » Oracle Spatial を使って、JSON ドキュメントに埋め込まれた Geo-JSON オブジェクトへの問合せを実行

.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 2015、Visual Studio 2013、Visual Studio 2012 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・デザイナー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 の詳細については、

<http://www.oracle.com/technetwork/jp/developer-tools/visual-studio/overview/index.html> にアクセスしてください。

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 は、.NET Framework 3.5 以上をネイティブにサポートします。

ODP.NET の詳細については、<http://www.oracle.com/technetwork/jp/topics/dotnet/index-085163-ja.html> にアクセスしてください。

Oracle Database Extensions for .NET は Windows 向け Oracle Database 12c の機能で、これによって C# や VB.NET などの .NET マネージド言語で記述されたストアード・プロシージャやファンクションの開発、デプロイ、実行が容易になります。.NET ストアド・プロシージャやファンクションは、Microsoft Visual Studio を使用して開発され、Oracle Developer Tools for Visual Studio の機能である、緊密に統合された .NET Deployment Wizard を使用してデプロイされます。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 のストアード・プロシージャ、トリガー、またはそれ以外でも、ストアード・プロシージャやストアード・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 の詳細については、<http://www.oracle.com/technetwork/jp/topics/dotnet/whatsnew/index-352676-ja.html> にアクセスしてください。

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 の詳細については、<http://www.oracle.com/technetwork/jp/topics/dotnet/whatsnew/index-352675-ja.html> にアクセスしてください。

.NETのおもな新機能

Application Continuity

データベース・セッションで障害が起こってリカバリ可能なエラーが発生した場合、Oracle Application Continuity によりデータベース・リクエストは中断されず迅速に、トランザクション操作または非トランザクション操作を自動的に再実行できます。Oracle Application Continuity は、計画内または計画外の関連エラーをマスキングすることで、エンドユーザー・エクスペリエンスを向上させます。例外を処理する複雑なロジックなしでアプリケーションを開発できる一方で、リカバ

り可能なエラーが発生したときには自動的にデータベース操作を実行できます。

ODP.NET の Unmanaged Driver 12.2 では、エンドユーザーからは完全に透過的に、多くのシステム障害、通信障害、ハードウェア障害、ストレージの停止を自動的にマスキングし、回復させる Application Continuity が新たにサポートされています。アプリケーションで .NET コードを変更せずに、ODP.NET Application Continuity を既存のアプリケーションに追加でき、機能の採用を簡素化します。

Oracle Sharding

Oracle Sharding は、1 つ以上のローカルまたはグローバルなデータセンターに配置された低コストの汎用データベース・サーバーに対し、データベースをほぼ線形にスケーリングできるシェアード・ナッシング・アーキテクチャです。シャーディングのメリットには、グローバルなデータ分散（特定のデータを消費者の近くに保管）、障害の封じ込め（1 つのシェアードに障害が発生しても、他のシェアードの可用性には影響しない）があります。Oracle Global Data Services はシェアード間のデータの位置を管理し、ODP.NET クライアント・リクエストをこの分散データベース・システムの適切なシェアードに送ることを可能にします。

Release 12.2 以降、ODP.NET の管理対象外ドライバと Oracle Database はシャーディングをサポートします。Oracle Sharding は、複数の個別の Oracle データベース（シェアード）にまたがってデータを水平方向にパーティション化する機能を提供します。ODP.NET は、接続文字列に指定されたキーに基づいてデータベース・リクエストを特定のシェアードに送ります。

Oracle Cloud

マネージドおよびアンマネージド ODP.NET は、Oracle Cloud、プライベート・クラウド、サード・パーティのクラウド環境に Web Deploy を介して容易にデプロイできます。すべての ODP.NET 固有の設定で、オペレーティング・システム・レベルの構成が不要になりました。これらの設定は、アプリケーションの .NET 構成ファイルで行うことができます。マネージドおよびアンマネージド ODP.NET では、統一された構成ファイル形式が共有されるようになったため、オンプレミスとクラウドでこの形式を使うことで、これらの構成ファイルをより簡単に均一的に管理できます。

オンプレミス ODP.NET は、新しい Oracle Database Exadata Express Cloud Service を含む Oracle Database Cloud (PaaS) に接続できるように容易に構成できます。ODP.NET アプリケーションは、Oracle Developer Tools for Visual Studio および Web Deploy を使って、Windows 上の Oracle Compute Cloud Service (IaaS) にデプロイできます。これらのクラウド・アプリケーションは、Microsoft .NET Framework と Internet Information Services でセットアップできます。詳細については、[Using .NET applications and Visual Studio with Oracle Public Cloud](#) の Web ページにアクセスしてください。

ODP.NET管理対象ドライバの強化機能

- » データの整合性 – ODP.NET の管理対象ドライバでは、データベース・サーバーとクライアント間のデータの整合性を強化するために、暗号化ハッシュ関数をサポートします。サポート対象のアルゴリズムには、MD5、SHA-1、SHA-2 (SHA-256、SHA-384、SHA-512) があります。
- » Transport Layer Security (TLS) – ODP.NET の管理対象ドライバでは、TLS 1.0 と SSL 3.0 の既存のサポートに加え、TLS 1.1 と 1.2 のサポートが追加されました。SSL/TLS を使った ODP.NET の管理対象ドライバ接続では、接続先データベース・サーバーに対し識別名 (DN) が正確であることを保証できるようになりました。

- » PL/SQL ブール – ODP.NET の管理対象ドライバでは、データベースの PL/SQL ブール・データ型を使用する場合に、OracleBoolean データ型をサポートします。ブールは TRUE または FALSE の値を保存します。ODP.NET OracleBoolean データ型により、ブール値とのパラメータ・バインディングとデータ型マッピングの設定が容易になります。

より長いスキーマID

ODP.NET では、表、列、ビュー、ストアド・プロシージャ、ファンクションなどのスキーマ・オブジェクトの ID 名を最長 128 字までサポートするようになりました。この機能は管理対象および管理対象外の両方の ODP.NET で利用できます。

グローバル化・サポート

Oracle Database 12c ではグローバル化・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。

グローバル化・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、Oracle7 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 32767 バイトまで拡張されたため、より長いテキストをより高い精度で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、グローバル化・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。途中で 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 (Oracle 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。24
- » Oracle ホワイト・ペーパー — Oracle Database 12c によるアプリケーション開発
- » 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 32767 バイトまで拡張されたことで、上限が 4000 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。

結論

Oracle Database 12c には、Oracle APEX、PL/SQL、SQL、C、C++、Java、.NET、Node.js、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 Application Continuity 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 32 K 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 や統合言語問合せを含む最新の .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。

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

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

CONNECT WITH US

 blogs.oracle.com/oracle

 facebook.com/oracle

 twitter.com/oracle

 oracle.com

Hardware and Software, Engineered to Work Together

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracle および Java は Oracle およびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。0115

ホワイト・ペーパー・タイトル
2015年1月
著者：[オプション]
共著者：[オプション]



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