



Oracle Database 12c Release 2の Oracle XMLデータベース

Oracle ホワイト・ペーパー | 2017年4月



目次

はじめに	1
XML の概要	1
XML スキーマ	2
XQuery と XPath	2
SOAP	2
XML のユースケース	4
データ取得のための XML	4
データ交換のための XML	4
業界標準 XML データ・モデルのための XML の永続性	4
アプリケーションの状態の永続性のための XML	4
オフィスの生産性および専門的オーサリング・ソフトウェアのための XML	5
Oracle XML DB の概要	6
効率的な XML ストレージ	6
バイナリ XML ストレージ	6
オブジェクト・リレーショナル XML ストレージ	7
リレーショナル XML ストレージ	8
標準ベースの XML 問合せおよび更新	8
XML の索引付け	9
XML 最適化コンテンツ・リポジトリ	10

Oracle Database 12c の Oracle XML DB 拡張機能	11
XQuery Update	11
XQuery Full Text と XML 全文索引付け	12
Oracle Multitenant アーキテクチャとの統合	14
HTTPS プロトコル構成と Oracle EM Express	14
Digest 認証	15
バイナリ XML のための XMLTable の最適化	15
パーティションおよびパラレル処理の改善	15
レプリケーションと可用性	15
ローリング・アップグレード	15
XML 管理性パッケージ	16
まとめ	16

はじめに

XML は、ビジネス・クリティカルな情報を提供したり、交換したりするための非常によく知られている方法です。XML は、単一のベンダーによって自由に変更されることのない、W3C が管理するオープン・スタンダードです。多くの業界で、情報を表示するための XML ベースのスタンダードが開発されました。これらのスタンダードは、通常、指定された XML ファイルに予期される内容を定義するために W3C で開発されたスタンダードである XML スキーマに基づいています。XML ベースのスタンダードは、医療、金融サービス、製造、出版、警察、公的機関で使用されています。また、XML は、SOAP ベースのアプリケーションを開発するための基盤も提供します。情報を交換する場合に、そのようなスタンダードを使用することが政府の規制によって実際に義務付けられる場面も増えてきています。これらの動向により、対処しなければならない XML の総量が大幅に増加したために、企業や組織は、他の種類の業務系データに必要とされるものと同程度の厳密さとセキュリティを備えた XML を管理する XML プラットフォームの導入を余儀なくされています。

このニーズを満たすために、オラクルは Oracle XML データベース (DB) を開発しました。Oracle XML DB は、すべてのバージョンの Oracle Database が備える機能の一つであり、高パフォーマンスのネイティブ XML ストレージおよび検索テクノロジーを提供します。Oracle XML DB は、従来のリレーショナル・データを管理するのと同じ方法による XML コンテンツの管理を可能にします。これにより、単一のプラットフォームですべてのミッション・クリティカル・データを管理および保護できるため、コストが削減され、投資収益率が向上します。

Oracle XML DB は、最初に Oracle9i Release 2 とともにリリースされた後、Oracle Database の主要リリースごとに機能が拡張されています。

XMLの概要

XML には本質的に自己記述的であるという長所があり、人間もマシンも XML を読むことができます。XML は、その自己記述的な性質のために、高密度のデータやスパース・データまたは変化の激しいデータの表現に非常に適しています。また、XML データ・モデルには優れた拡張性があるため、特定の情報ストレージおよび検索要件に合わせて XML コンテンツ・モデルを簡単にカスタマイズできます。XML の主要なユースケースは、以下のとおりです。

- データの取得：XML は、センサーやロガーによって生成されるデータを格納するために使用されます。
- データの交換：XML は、疎結合システム間の情報交換に使用されます。
- XML の永続性：業界標準のデータ・モデルに基づく永続性
- XML の永続性：アプリケーション・オブジェクト、メタデータ、および状態の永続性
- XML の永続性：広く使用されている生産性ソフトウェアによって作成されるコンテンツの永続性

これらのユースケースについては、後で詳しく説明します。W3C は、基本的な XML スタンドードに加えて、XML コンテンツに関連する多数のスタンダードを開発しました。次にその例を挙げます。

XMLスキーマ

XML ドキュメントの複雑さが増すにつれて、XML ドキュメントに予期される内容がいかなるものかを記述するための標準化された言語の必要性も高まります。XML スキーマ・スタンダードで指定されるボキャブラリにより、XML ドキュメントのクラスの内容と構造を正確かつ明確に記述する型定義と要素宣言のコレクションを定義することが可能になります。XML スキーマで定義される約 49 の基本データ型を組み合わせ、別のより複雑なオブジェクトを構築できます。XML スキーマ標準には、「スキーマのためのスキーマ」として知られる、XML スキーマ言語を定義する XML スキーマが含まれています。XML スキーマは、単に「スキーマのためのスキーマ」で定義されたボキャブラリに準拠した XML ドキュメントです。

交換する情報を正確に記述したり、交換する情報が同意した仕様に従っていることを確認したりすることが可能になるため、XML スキーマは広く採用されました。多くの業界標準化団体が、顧客、サプライヤ、およびパートナーの間で情報を交換するために必要な永続性モデルと交換モデルを、XML スキーマを使用して定義しました。

XQueryとXPath

XQuery は、SQL がリレーショナル・コンテンツのための自然な問合せ言語であるのと同様に、XML コンテンツのための自然な問合せ言語です。XQuery では、XML ドキュメントの特定の部分に対処するために W3C の XPath 式構文のスーパーセットが使用されます。XQuery では、これが、一連のノードに対する反復や結合処理の実行に使用される SQL に似た「FLWOR 式」によって補われています。FLWOR 式は、その名前の由来である FOR、LET、WHERE、ORDER BY、RETURN の 5 つの句によって作成されます。


XQuery を使用して新しい XML ドキュメントを作成することもできます。この言語は、必要な XML ドキュメントを合成するためにネストされた構造で組み合わせることができる要素コンストラクタと属性コンストラクタを定義します。

XQuery は、XQuery データ・モデル (XDM) に基づいています。XDM では、すべての値が 0 以上の項目のシーケンスとして表されます。シーケンス内の項目となるのは、XML ノードまたはアトム型の値です。アトム型の値となるのは、XML スキーマなどによって定義される任意の基本データ型の値です。型の完全なリストは、XML スキーマによって定義される基本データ型に基づきます。

XQuery 言語の最近の拡張には、XQuery-Update 勧告と XQuery Full 勧告が含まれます。XQuery-Update は、XML ドキュメントの内容の変更を可能にしますが、全面的な変換は実行しません。XQuery-Update は、挿入、削除、変更、および名前変更操作をサポートします。XQuery Full-Text は、ドキュメント、フラグメント、またはノード・レベルでのテキストの検索を可能にして、XML ドキュメントの複雑な全文検索を支援します。

SOAP

SOAP は、今日のアプリケーション開発者の間で非常に広く使用されているもう一つのスタンダードです。SOAP スタンドードにより、XML の活用範囲が広まり、サービスベースのインフラストラ



クチャが実現されます。SOAP サービスは、Web Services Description Language (WSDL) ドキュメントによって定義されます。SOAP サービスを起動するために、SOAP クライアントは、リクエスト・ドキュメントを SOAP サーバーに送信します。サービスの起動の結果は、レスポンス・ドキュメントを使用して SOAP クライアントに返されます。XML は、リクエスト・ドキュメントとレスポンス・ドキュメントの両方に使用されます。WSDL も XML ドキュメントです。サービスのエンドポイントの場所を指定し、またリクエスト・ドキュメントとレスポンス・ドキュメントの定義を提供します。WSDL は、埋込み XML スキーマを使用してこの情報を提供します。

XMLのユースケース

データ取得のためのXML

センサーやロガーによって生成されたデータ、またはアプリケーション・ロギングによって生成されたデータが、XML として保存されます。シナリオでは、非常に大きな XML が比較的短い期間に生成されます。XML は、少数の非常に大きなファイルとして格納される場合も、非常に多くの比較的小さいファイルとして格納される場合もあります。このデータは、ビジネスのアプリケーション・プロセスに統合される必要があります。

データ交換のためのXML

このシナリオでは、システムが、XML メッセージを使用して相互に通信します。XML は、あるシステムによって管理されるデータ（通常、リレーショナル・データ）から生成され、別の場所に転送されて、別のシステムによって管理されるデータ・ストア（通常、リレーショナル・データ・ストア）に統合されます。XML を交換メカニズムとして使用することにより、1 つのアプリケーション・システムが、そのデータの再構築を、そのデータへのアクセスを必要とするどのアプリケーションにも影響を与えずに行うことを可能にする抽象化レイヤーが提供されます。

一部のモデル（SOAP ベースのメッセージングなど）では、非常に多くの小さな（4~100 KB）XML ドキュメントがほぼリアルタイムで交換されます。他の事例では、大規模（100 KB から 10 GB 超）XML ファイル用の中小容量のボリュームが作成され、従来のバッチ生成および ETL 処理技術を使用して処理されます。多くの場合、これらのメッセージは業界の標準化団体で開発された業界標準の XML スキーマに準拠しています。

業界標準XMLデータ・モデルのためのXMLの永続性

XML の永続性が強化された原因の一つは、業界ベースの XML 標準の出現でした。これらのスタンダードは、発展するにつれて、それらを使用する構成組織すべてのニーズを満たすために、非常に複雑で多様性のある情報モデルを定義するようになります。これらの XML スキーマのうちのいくつかは何千もの表を含むリレーショナル・データ・モデルに変換されるモデルを記述することが、分析によって示されています。モデルの複雑さと多様性が増大するにつれて、リレーショナル・モデルと XML モデルの間で双方向の変換を実行するソフトウェアを開発するコストと時間は非常に大きなものとなります。

XML の問合せ技術と索引付け技術における最近の改善により、企業や組織は、XML コンテンツで直接作業することによってコストやリソースを節約できるかどうかを確かめようとしています。このために、専門的な方法で保持および管理される必要のある XML の量が大幅に増加しています。XML の永続性に対する要求の増大は、否認防止のために単純な XML メッセージでさえ保持する必要があることも原因となっています。

アプリケーションの状態の永続性のためのXML

XML モデルは、その固有の柔軟性により、データ・モデルが非常に変化しやすく、多くの場合には、データ・モデルが事前に分からないアプリケーションにおいてさえデータを保持できる方法として特に有望視されています。XML ベースの永続性を採用することにより、インストール・プロセスの一部として、また場合によっては実行時に、データ・モデルを即座に変更できるアプリケーション

の作成が可能になります。多くのアプリケーション開発者が、アプリケーションの要件が変わるたびにデータ・ストレージ・モデルを変更する必要のない拡張性を実現する方法として、XML を使用しています。

一部のアプリケーション開発者は、ハイブリッド・モデルを使用します。このモデルでは、データ・モデルのよく知られている部分が従来のリレーショナル表を使用して保持され、その他の十分に定義されていない部分や変化しやすい部分が XML を使用して保持されます。また、別のアプリケーション開発者は、アプリケーションのすべてのデータと状態が一連のよく知られた単純なグローバル識別子 (GUID) によってアクセスされる一連のドキュメントに含まれるドキュメント保管モデルを使用します。その他に、XML を使用して名前と値のペア・モデルによるストレージ・モデルが保持されるユースケースが存在します。これは、オブジェクト永続性レイヤー (Hibernate など) を使用して、アプリケーションによって定義されるオブジェクトとストレージ・レイヤーによって認識されるオブジェクトの間でのマッピングが行われる場合には、特によく見られます。

このハイブリッド・モデルは、コンテンツ管理システムのようなアプリケーションにおいて非常によく使用されます。この種のアプリケーションでは、アプリケーションによって管理されるオブジェクトに関連付けられるメタデータのクラスを業務ユーザーが定義できる必要があります。アプリケーションがデプロイされると、このプロセスが動的に実行されます。メタデータが定義されると、アプリケーションは、ストレージ・インフラストラクチャが効率的なデータ管理および問合せ機能の提供に関してメタデータをファーストクラス・オブジェクトとして扱うことを予期します。

オフィスの生産性および専門的オーサリング・ソフトウェアのためのXML

XML データ・モデルは、その固有の柔軟性と多様性により、オフィス生産性スイートや専門的オーサリング・ソフトウェアを使用して作成されるコンテンツを保管するためのメカニズムとして有視されています。この目的のために明示的に設計された標準である SGML が XML に先行するものであったことを考えれば、これは驚くべきことではありません。さまざまなベンダーのオフィス生産性ツール間での相互運用性を向上させる最近の動向により、この種のコンテンツを保管するための XML ベースのオープン・ソース標準が開発されました。オフィス生産性スイートの大半のベンダーは、現在、新しい XML ベースのコンテンツ・モデルをサポートする製品を出荷しています。そのため、この種のコンテンツが大幅に増加しています。Microsoft Office 2007 以降で使用されている DOCX、XSLX、および PPTX ファイル形式は、この動向の例です。'X'ファイル形式は、実際には、一連の XML ファイルとしてオフィス・ドキュメントの内容が格納される ZIP アーカイブです。

オフィス生産性ドキュメントを保持する主要な方法として XML が採用されることにより、XML ベース・コンテンツ管理 (CM) システムに対する関心が増えています。これらのシステムが従来の CM ソリューションと異なるのは、それらが管理するドキュメントのメタデータとコンテンツの両方を把握できるために、それらのユーザーが、前世代のデスクトップ生産性ソフトウェアにより作成されたドキュメントに拘束されてきた大量の情報を非常に効率よく使用できる点です。

Oracle XML DBの概要

Oracle XML DB は、すべてのバージョンの Oracle Database が備える機能の一つであり、高パフォーマンスのネイティブ XML ストレージおよび検索テクノロジーを提供します。Oracle XML DB は、XML、名前空間、DOM、XQuery、SQL/XML、XSLT を含むすべての主要な XML 規格を完全にサポートしています。XML 規格の完全サポートにより、Oracle XML DB は、ネイティブ XML アプリケーション開発を支援します。アプリケーション開発者は、データベースに格納されている XML コンテンツを、XML 中心の技術を使用して格納、管理、編成、および操作することができます。Oracle XML DB は SQL/XML スタンダードもサポートしているため、SQL 中心の開発技術によって、Oracle Database 12c Release 2 に格納されているリレーショナル・データから XML を直接公開できます。

Oracle XML DB のおもな特徴は次のとおりです。

- 効率的な XML の永続性
- 標準準拠の問合せおよび更新操作
- 強力で柔軟な索引付け
- Oracle のセキュリティおよびデータ管理機能との緊密な統合
- XML 中心、SQL 中心、およびドキュメント中心の開発のサポート
- XML と SQL の相互運用性

効率的なXMLストレージ

XML コンテンツは、XMLType データ型を使用してデータベースに格納されます。XMLType データ型は、複数のストレージ・モデルをサポートする抽象化です。これにより、Oracle XML DB は、このホワイト・ペーパーの最初の項で示した多岐にわたる使用モデルのすべてに関して XML ストレージを最適化できます。Oracle Database 12c Release 2 は、大別すると以下の 3 つの主要ストレージ・モデルをサポートします。

- バイナリ XML ストレージ
- オブジェクト・リレーショナル XML ストレージ
- リレーショナル・ストレージ

XMLType が、基礎となるストレージ・モデルの抽象化であるという事実により、アプリケーション開発者は、ディスクへの XML の格納方法とは無関係にアプリケーションのコードを作成できます。XMLType の CLOB ベースのストレージは、バイナリ XML ストレージを優先する Oracle Database 12c Release 2 以降では推奨されません。

バイナリXMLストレージ

バイナリ XML では、XML のネイティブ・バイナリ表現を使用して XML がデータベースに格納されます。この形式は、"コンパクトなスキーマ対応の XML" (CSX) として知られます。CSX 形式を使用すると、ストレージ要件が約半分に削減されます。CSX エンコードの最初の手順は、すべてのタグ・ベースのコンテンツのトークン化です。この最適化は、スキーマベースの XML と非スキーマ

ベースの XML の両方に適用されます。

スキーマベースの XML については、ネイティブのデータ型を使用して非テキスト値を表し、CSX 形式から不必要なトークン情報を排除することによって、さらにストレージが最適化されます。これにより、スキーマベースのバイナリ XML は、合計で 60~75% のストレージを削減できます。

バイナリ XML は、Oracle の SecureFiles Lob インフラストラクチャを使用してディスクに格納されます。Oracle SecureFiles Lob を使用すると、XML ドキュメントのストレージおよび検索操作に関して可能な最大限のスループットが実現されます。バイナリ XML は、SecureFiles のスライド挿入機能を利用して、XML コンテンツをノード・レベルで効率よく挿入、更新、および削除できます。ストレージをさらに節約する必要がある場合は、スキーマベースと非スキーマベースの両方のバイナリ XML を、Oracle Database の圧縮機能を使用してさらに圧縮できます。

可能な場合はいつでも、バイナリ XML に関する操作がストリーム技術を使用して実行されます。ストリーム技術により、従来の DOM ベースの XML コンテンツ処理に関連するメモリおよび CPU オーバーヘッドが回避され、パフォーマンスが向上します。ストリームは、XML コンテンツの解析とエンコードの両方およびスキーマベース XML の確認のための取得時にサポートされます。CSX 形式も X-Path ストリーム操作のために最適化されます。これにより、データベースは、大量の XML に対して複雑な X-Path 式を効率的に評価できます。複数のリーフ・ノードまたはフラグメントを、XML ドキュメントの 1 回の処理で抽出できます。

CSX 形式は、XML スキーマベースのコンテンツの非常に柔軟なサポートを提供します。異なる XML スキーマまたは 1 つの XML スキーマの各バージョンに関連付けられたドキュメントを、同じ表または列に格納できます。

オブジェクト・リレーショナルXMLストレージ

オブジェクト・リレーショナル・ストレージは、XML ドキュメントを一連の SQL オブジェクトとして保持します。SQL オブジェクト・モデルは、XML スキーマによって定義されるオブジェクト・モデルをコンパイルすることにより生成されます。このプロセスは、XML スキーマ登録と呼ばれます。これらのオブジェクトは、XML スキーマに準拠する XML ドキュメントを、忠実度を失わずにデータベースに格納することを可能にします。XML スキーマにおける要素の繰り返しは順序付けされたコレクションに自動的にコンパイルされ、再帰的な構造は XML スキーマのコンパイル・プロセスにおいて自動的に処理されます。

バイナリ XML ストレージと同様に、オブジェクト・リレーショナル・ストレージによっても、領域を大幅に節約できます。XML ドキュメントと SQL オブジェクト・モデルのコンテンツ間の明示的なマッピングにより、すべてのタグ付け情報を廃棄できます。すべての非テキスト値をネイティブ Oracle データ型として格納することによっても節約が実現されます。

XML を一連のオブジェクトとして取得して格納すると、XML に対して実行されるすべての操作が、その基礎となるオブジェクトに対する操作に書き換えられます。XML に対する XQuery および XPath ベースの操作は、リレーショナル・コンテンツに対する操作に使用されるものと同じ代数にコンパイルされます。これにより、Oracle Optimizer は、SQL を最適化する場合と同じ方法で XQuery を最適化できます。

オブジェクト・リレーショナル・ストレージは、XML スキーマ拡張の概念（W3C が勧告する、XML スキーマへの変更を処理する方法）をサポートします。開発された拡張スキーマは、データベースに登録できます。XML DB は、XML スキーマのコンパイル・プロセスにおいて元の XML スキーマから導出されるオブジェクト・モデルを調整します。

Oracle XML DB は、インプレース・スキーマ拡充もサポートします。これにより、すでにデータベースに格納されているどのインスタンス・ドキュメントもアンロードおよびリロードすることなく、定義済みの変更のサブセットを、登録済みの XML スキーマに適用できます。必要な変更を加えることが、新しい XML スキーマに従った場合にインスタンス・ドキュメントの既存のコーパスが無効になることを意味しないかぎり、インプレース・スキーマ拡充がサポートされます。

リレーショナルXMLストレージ

XML DB では、SQL/XML と XQuery を使用して、リレーショナル・データを XMLType 値として公開するビューを作成できます。この XMLType は、バイナリまたはオブジェクト・リレーショナル・ストレージを利用する XMLType とまったく同じ方法で、XQuery および XPath を使用して問い合わせることができます。ほとんどの場合、これらのビューへの更新は、INSTEAD OF トリガーを使用して処理できます。

スタンダード・ベースのXML問合せおよび更新

データベースに格納された XML に対するすべての操作は、XQuery と XPath を使用して実行されます。Oracle XML DB は、Oracle Database に格納された XMLType 値を、XQuery を使用して操作するための包括的な支援を提供します。SQL 標準に対する SQL/XML 拡張は、一連の演算子、XMLQuery、XMLTable、および XMLExists を定義して、SQL 文のコンテキストでの XQuery 操作の実行を可能にします。これにより、XML 操作で、他のデータベース操作と同じトランザクション・セマンティクスを使用できるようになります。SQL/XML 拡張は、XMLCast 演算子も定義します。これにより、XML スキーマによって定義される基本データ型と SQL によってサポートされるスカラー型の間での変換が可能になります。Oracle Database 12c は、W3C の XQuery Update 勧告を使用して XML コンテンツを更新できる新機能を備えています。

Oracle XML DB は、JSR-225 と呼ばれる XQJ のサポートも提供しています。これは、XQuery 操作を実行するための JDBC のようなインターフェースを提供する Java スタンダードです。これにより、Java プログラマーは、Oracle Database 12c Release 2 に格納された XML データを直接操作する純粋な XQuery ベースのアプリケーションを開発できます。Oracle XML DB は、データベースが SOAP ベースの環境の直接の参加者となることを可能にする SOAP エンドポイントを提供します。このエンドポイントは、XQuery サービスおよび SQL サービスと、PL/SQL パッケージ、プロシージャ、およびファンクションをサービスとして起動するメカニズムをサポートします。

Oracle XML DB は、DBMS_XMLDOM および DBMS_XMLPARSER パッケージによる XML コンテンツの DOM レベルでのアクセスおよび操作を提供します。XML コンテンツの XSL 変換は、SQL 演算子の XMLTransform と DBMS_XSLTRANSFORM PL/SQL パッケージによってサポートされます。

これらの機能の使用方法について詳しくは、Oracle XML DB の開発者ガイドを参照してください。

XMLの索引付け

Oracle XML DB は、多数の異なる XML 索引付けモデルをサポートします。これにより、すべての種類の XML コンテンツの効率的な索引付けが可能になります。

Oracle のバイナリ XML 形式は、XML コンテンツを効率的に索引付けできるように設計されています。このバイナリ XML 形式は、XQuery 操作を最適化してフラグメントおよびリーフ・レベルでの XML コンテンツへのアクセスを実現するために一から設計されました。XML ストレージの索引付けには、次の 3 つの異なる手法があります。

- 非構造化 XML 索引
- 構造化 XML 索引
- XML 全文索引

非構造化 XML 索引は、主として、XML の構造と問合せのいずれかが事前に分からないユースケース用に設計されています。そのもっとも基本的な形式では、ドキュメントのすべてのノードに索引が付けられます。これにより、索引付けされる XML ドキュメントのコーパスに対する可能なすべての問合せを最適化できるようになります。完全な索引を保持すると、特に XML が大量である場合、非常に多くのリソースが消費される可能性があります。Oracle XML DB では、索引付けをドキュメントのノードのサブセットに制限すること、または非同期方式で索引付けを実行することにより、非構造化 XML 索引のリソース使用率を調整できます。

構造化 XML 索引は、XML の構造と実行される一連の問合せが十分に分かっているユースケースに主眼が置かれています。構造化 XML 索引では、問合せに回答するために必要なデータが、内部的に、一連のリレーショナル表に投影されます。XQuery 式が実行される際、XQuery プロセッサは、これらの表に対してリレーショナル問合せを実行し、XQuery 式を満たす XML ドキュメントを特定します。リーフ・レベルの抽出を必要とする XQuery 式も、索引からデータを直接抽出することによって最適化できます。

XML 全文索引は、Oracle Database 12c の新機能です。この機能により、Oracle Database に格納された XML ドキュメントに対する XQuery Full-Text 操作が最適化されます。この機能については、後で、このホワイト・ペーパーの新機能に関する項において詳しく説明します。

オブジェクト・リレーショナル・ストレージの索引付けは、任意の複雑なリレーショナル階層の索引付けに似ています。Oracle では、オブジェクト階層の双方向横断を最適化するために必要な B ツリー索引が自動的に作成されます。DBA は、他のアクセス・パスをサポートするために、別の B ツリー索引を必要に応じて自由に追加できます。

XML最適化コンテンツ・リポジトリ

Oracle XML DB には、高パフォーマンスの XML 中心のリポジトリが組み込まれています。このリポジトリにより、使い慣れたファイル/フォルダのメタファによる Oracle Database に格納されたコンテンツの編成およびアクセスが可能になります。リポジトリには、SQL から直接アクセスすることもできますし、標準インターネット・プロトコル（HTTP、FTP、WebDAV など）を使用してアクセスすることもできます。これにより、データベースに格納された XML コンテンツに一般的なデスクトップ・ツール（Microsoft Office、Windows エクスプローラなど）から直接アクセスできます。

リポジトリは、基本的なコンテンツ管理操作（チェックアウト/チェックイン機能によるバージョンングなど）をサポートするとともに、データベース・トリガーがリレーショナル表に対する操作への対応を可能にするのと同じ方法で、

フォルダ/ファイルに対する操作への対応を可能にするイベント・モデルをサポートします。

この Oracle XML DB Repository により、Oracle Database 12c Release 2 は、コンテンツ中心の XML アプリケーションの開発とデータ中心の XML アプリケーションの開発のニーズを満たすことができます。

Oracle Database 12cのOracle XML DB拡張機能

Oracle Database 12c Release 2 のリリースにより、オラクルは、業界最高の XML サポートを拡張して、すべてのタイプの XML コンテンツの保管、管理、および問合せに最適なプラットフォームを維持しています。Oracle Database 12c Release 2 には、新しいリリースごとに新機能を提供するとともにパフォーマンスとスケーラビリティを改善するオラクルの継続的な取組みが反映されています。Oracle Database 12c Release 2 以降、Oracle XML DB と Oracle XML DB Repository は、Oracle Database の必須機能となっています。Oracle XML DB のインストール方法や、特定のデータベース・インスタンスで Oracle XML DB を使用できるかどうかといったことについても、今後は心配する必要がありません。Oracle Database 12c で提供される新機能には、以下のサポートが含まれます。

- XQuery Update
- XQuery Full-Text
- Oracle Multitenant オプション
- HTTPS および Oracle Enterprise Manager Database Express (Oracle EM Express)
- ダイジェスト認証

Oracle Database 12 における XML 処理のためのパフォーマンス、可用性、およびスケーラビリティの改善には、以下のものが含まれます。

- XMLTable の最適化
- パーティション化の改良点
- ロジカル・スタンバイおよび Oracle GoldenGate との統合
- ローリング・アップグレードのサポート

使いやすさの改善には、以下のものが含まれます。

- XML 管理性パッケージ

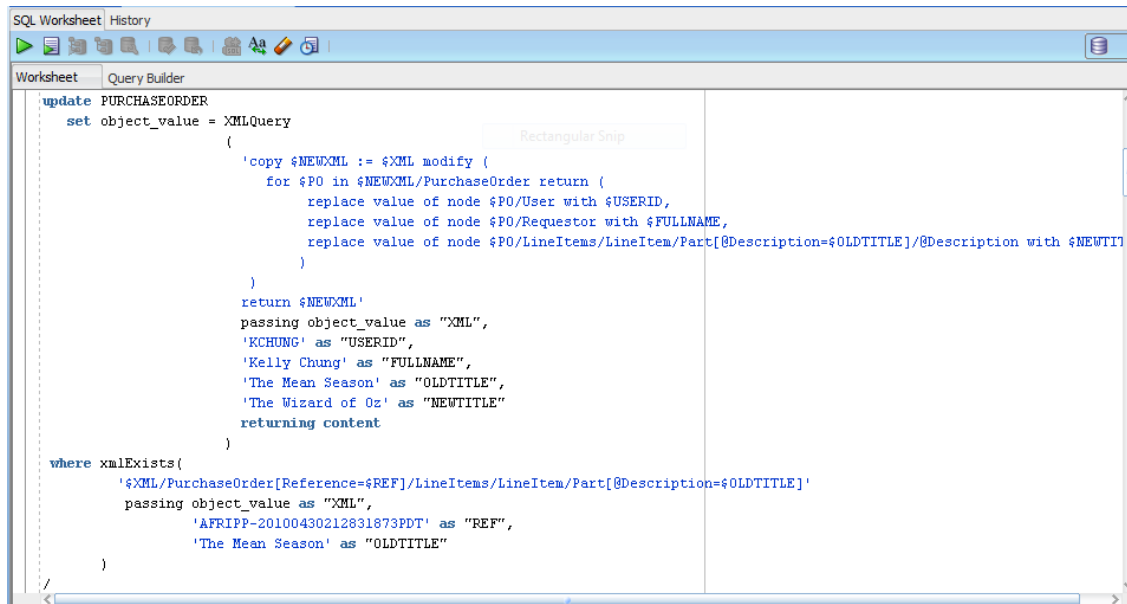
XQuery Update

Oracle Database 12c では、Oracle XML DB により、W3C の XQuery Update 勧告のサポートが追加されます。XQuery Update により、XML ドキュメントに対する以下の操作が可能になります。

- ノードの挿入
- ノードの削除
- ノードの変更
- ノード名の変更

Oracle Database 12c では、XQuery Update 操作は、既存の XMLQuery 演算子を使用して起動します。この演算子は、SQL の SELECT 文または UPDATE 文の一部として使用できます。SQL UPDATE 文の場合は、標準データベース・トランザクション・セマンティクスが適用されます。可能な場合には、XQuery Update 操作がターゲット XML ドキュメントの部分更新に変換されます。

Oracle XML DB は、XQuery Update の'コピー変更'形式がサポートされます。次の図は、これの簡単な例を示しています。



```
update PURCHASEORDER
set object_value = XMLQuery
(
  'copy $NEWXML := $XML modify (
    for $PO in $NEWXML/PurchaseOrder return (
      replace value of node $PO/User with $USERID,
      replace value of node $PO/Requestor with $FULLNAME,
      replace value of node $PO/LineItems/LineItem/Part[@Description=$OLDTITLE]/@Description with $NEWTITLE
    )
  )
  return $NEWXML'
  passing object_value as "XML",
  'KCHUNG' as "USERID",
  'Kelly Chung' as "FULLNAME",
  'The Mean Season' as "OLDTITLE",
  'The Wizard of Oz' as "NEWTITLE"
  returning content
)
where xmlExists(
  '$XML/PurchaseOrder[Reference=$REF]/LineItems/LineItem/Part[@Description=$OLDTITLE]'
  passing object_value as "XML",
  'AFRIPP-20100430212831873PDT' as "REF",
  'The Mean Season' as "OLDTITLE"
)
```

図1：SQLのUPDATE文の一部としてのXQuery Update操作

XML コンテンツを更新するための Oracle 独自の演算子（UpdateXML、InsertChildXML、DeleteXML、AppendChildXML、InsertXMLBefore など）は、Oracle Database 12c では推奨されません。XQuery-Update Facility は、Oracle Database に格納された XML コンテンツを更新するために包括的なスタンダード準拠のメカニズムを提供します。これらには、推奨されなくなった演算子によって提供される機能がすべて含まれます。

なお、XQuery Update のサポートは、Oracle Database 11g でも 11.2.0.3.0 以降のパッチ・セットを適用することで有効になります。XQuery-Update 勧告について詳しくは、次の URL を参照してください。

<http://www.w3.org/TR/xquery-update-10/>

XQuery Full TextとXML全文索引付け

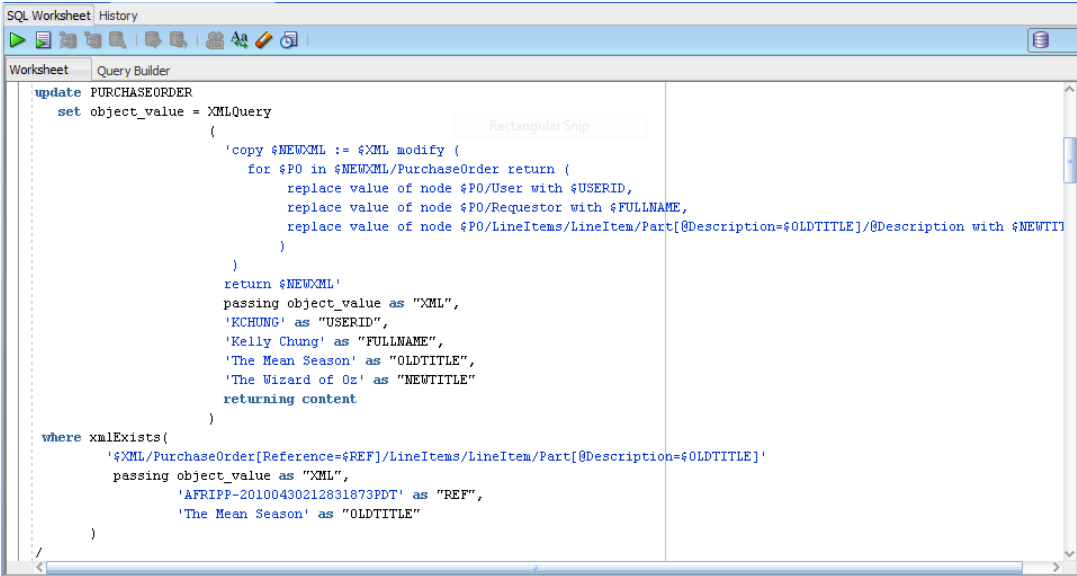
XML は、さまざまな種類の情報を表すために使用できます。この情報の性質は、高度に構造化されたデータ（固定されたスキーマ、数や日付などの既知の型）から、半構造化データ（柔軟なスキーマおよび型）、マークアップ・データ（タグが埋め込まれたテキスト）、非構造化データ（タグの付いていない自由に流動するテキスト）にいたる全範囲に及びます。XML の構造化の程度が低下するにつれて、情報検索（IR）技術を使用した検索の必要性が増大します。

全文検索は、サブstring検索とは大きく異なります。全文検索では、サブstringではなくトークンやフレーズが検索されます。たとえば、"sport"という文字列を含む項目のサブstring検索では、"Transport"を含む項目も返されます。"sport"というトークンの全文検索では、そのような結果にはなりません。XQuery 1.0 勧告では、サブstring照合を実行する"contains"という演算子がすでに定義されています。W3C XQuery and XPath Full Text Facility 1.0（XQuery-FT）により、XQuery 1.0 および XPath 2.0 の構文とセマンティクスが拡張され、全文検索機能が提供されます。

全文検索は、言語ベース検索をサポートする必要があります。言語ベース検索の例は、"sport'と同じ語幹を持つトークンを含むすべての項目を検索する" ("sport"や"sports"は検出されるが"transport"は検出されない) や"Oracle と Database というトークンを相互の 3 トークン以内に含むすべての項目を検索する"といったものです。これらの検索はどちらも、単純なサブストリングのアプローチでは実装できません。

全文検索のもう一つの課題は、結果の順序付けとランク付けです。全文検索は、精密科学ではありません。検索結果は、最重要の結果が最初に示される順番で表示されるようにランク付けされる必要があります。

Oracle Database 12c では、Oracle XML DB により、XQuery-FT をサポートするように XQuery 実装が拡張されます。オラクルは、この重要な W3C 標準のサポートを提供した最初の大手データベース・ベンダーです。次の図は、XQuery-FT 検索の簡単な例を示しています。



```
update PURCHASEORDER
set object_value = XMLQuery
(
  'copy $NEWXML := $XML modify (
    for $PO in $NEWXML/PurchaseOrder return (
      replace value of node $PO/User with $USERID,
      replace value of node $PO/Requestor with $FULLNAME,
      replace value of node $PO/LineItems/LineItem/Part[@Description=$OLDTITLE]/@Description with $NEWTIT
    )
  )
  return $NEWXML'
  passing object_value as "XML",
  'KCHUNG' as "USERID",
  'Kelly Chung' as "FULLNAME",
  'The Mean Season' as "OLDTITLE",
  'The Wizard of Oz' as "NEWTITLE"
  returning content
)
where xmlExists(
  '$XML/PurchaseOrder[Reference=$REF]/LineItems/LineItem/Part[@Description=$OLDTITLE]'
  passing object_value as "XML",
  'AFRIPP-20100430212831873PDT' as "REF",
  'The Mean Season' as "OLDTITLE"
)
```

図2 : XQuery Update Full Text検索

XQuery-FT は、Oracle Database の Oracle Text 機能を利用して全文検索を最適化します。Oracle Text は、SQL 言語のユーザーに CONTAINS 演算子による強力な全文検索機能を提供する、Oracle Database の十分に確立された機能です。Oracle Text のパフォーマンスの鍵となるのは、"ctxsys.context"索引です。これにより、データベースは、膨大なテキスト・コンテンツを効率的に索引付けし、検索できます。Oracle Database 12c では、XQuery-FT によって使用される XML データ・モデルを完全に把握するように、この索引が拡張されました。この索引は、CONTAINS ベースの検索をサポートするのと同じ方法で XQuery-FT ベースの検索をサポートできるようになっています。

XQuery and XPath Full Text Facility 1.0 勧告について詳しくは、次の URL を参照してください。

<http://www.w3.org/TR/xpath-full-text-10/>

Oracle Multitenantアーキテクチャとの統合

Oracle Multitenant オプション・データベースを使用すると、複数の Oracle データベースを単一のデータベース・インスタンスに統合でき、それにより、多数の Oracle データベースの管理が非常に容易になります。マルチテナント環境では、マルチテナントのコンテナ・データベース (CDB) が、1 つ以上のプラグブル・データベース (PDB) の管理を担当します。

Oracle XML DB は、マルチテナント・アーキテクチャに完全に統合されています。CDB と各 PDB は、Oracle XML DB Repository のプライベート・インスタンスを持ちます。PDB レベルでの XDB サービスの構成は、依然として PDB の `xdbconfig.xml` ファイルによって管理されます。Oracle XML DB Protocol Server は、2 つ以上の PDB が HTTP サービスと FTP サービスを提供するマルチテナント構成をサポートします。

HTTP プロトコルと FTP プロトコルの性質により、HTTP または FTP サービスを提供する PDB ごとに一意のポートを割り当てる必要があります。共有サーバーは PDB ではなく CDB のプロパティであるため、マルチテナント構成に必要な共有サーバーの数は、インスタンスによって提供されるすべての PDB の要件を十分に満たすものである必要があります。新しい PDB をマルチテナント構成に組み込む場合は、HTTP ポート番号と FTP ポート番号に新しい値を割り当てないとデータベースを開くことができない可能性があります。

データベースへの匿名アクセスを有効にする場合は、まず、ANONYMOUS アカウントのロックを CDB レベルで解除する必要があります。ANONYMOUS のロックを CDB で一度解除すると、必要に応じて、各 PDB でロックしたり解除したりできます。ANONYMOUS が Oracle XML DB Repository 内のパブリック・コンテンツにアクセスできるかどうかは、依然として、PDB の `xdbconfig.xml` ファイル内の `ALLOW_ANONYMOUS_REPOSITORY_ACCESS` 要素によって制御されます。

HTTPSプロトコル構成とOracle EM Express

Oracle Database 12c では、Database Console アプリケーションが Oracle EM Express によって置き換えられています。Oracle EM Express は、Oracle XML DB プロトコルを使用してデータベースから直接提供されます。データベースを開いたらすぐに Oracle EM Express を使用できるように、HTTPS プロトコルがデフォルトで有効になっています。HTTPS プロトコルのデフォルト・ポートは 5500 であり、Oracle EM Express にアクセスするには `/em'` として構成されたサブレットを使用します。マルチテナント構成では、Oracle EM Express は、CDB においてデフォルトで有効になります。

HTTPS プロトコルを使用してリポジトリにアクセスするとブラウザには必ず、セキュリティ証明書に関するエラーがあると表示されます。ブラウザには、以下の問題が示されます。

この Web サイトによって提供されるセキュリティ証明書は、信頼できる認証局によって発行されたものではありません。

この Web サイトによって提供されるセキュリティ証明書は、異なる Web サイトのアドレスに関して発行されています。

これらのエラー・メッセージは、無視しても問題ありません。これらのエラーが表示されないようにするには、Oracle Database に付属のデフォルトのセキュリティ証明書の代わりに、ユーザー独自のセキュリティ証明書をインストールする必要があります。

ダイジェスト認証

Oracle Database 12c では、Oracle XML DB の HTTP サーバーによってダイジェスト・ベースの認証がサポートされています。これにより、通信時に HTTP パスワードが安全に暗号化されるようになりました。ダイジェスト認証は HTTP でのみサポートされることに注意してください。FTP プロトコルの使用時には、パスワードが危険にさらされないように引き続き注意する必要があります。ダイジェスト認証を有効にすると、HTTP 認証のユーザー名に関して大文字と小文字が区別されるようになるという副作用があります。これは、ほとんどの構成において、HTTP でダイジェスト認証を使用する場合にデータベース・ユーザー名を大文字で入力しなければならないことを意味します。

バイナリXMLのためのXMLTableの最適化

バイナリ XML ストレージ・モデル上での XMLTable 操作のパフォーマンスを向上させる重要な作業が実施されました。この作業では、Oracle Database 12c でのバイナリ XML に関する XMLTable 操作が Oracle Database 11g (11.2.0.3.0) と比較して 2 倍以上の速さで作動することを可能にする新しい行ソースを作成する必要がありました。XMLTable 操作の速度の向上に加えて、これらの最適化により、構造化 XML 索引に関連した索引メンテナンス操作のパフォーマンスも改善されます。

パーティションおよびパラレル処理の改善

構造化 XML 索引に関するパラレル索引メンテナンスが全面的にサポートされました。レンジ、リスト、およびハッシュのパーティション化スキーマが、構造化および非構造化 XML 索引に対してサポートされました。これにより、XML を使用する顧客は、今日のマルチプロセッサ構成で実現された非常に高い並列処理度をフル活用できます。

レプリケーションと可用性

Oracle Database 12c では、広く使用されている Oracle のロジカル・スタンバイ・オプションによって、すべての種類の XMLType 永続性を使用できます。さらに、すべての形式の XMLType ストレージの Oracle GoldenGate によるレプリケーションもサポートされました。

ローリング・アップグレード

Oracle XML DB がインストールされた Oracle データベースは、ローリング・アップグレードに参加できるようになりました。これにより、データベース・サービスを停止させる必要がなくなり、Oracle Real Application Clusters (Oracle RAC) 構成でのデータベース・アップグレードのスケジュール設定が非常に容易になります。

XML管理性パッケージ

Oracle Database 12c は、XML 管理性パッケージがデフォルトでインストールされる Oracle Database の最初のリリースです。これらのパッケージにより、Oracle XML DB 関連の共通タスクの実行が非常に容易になります。現時点では、以下の 2 つの管理性パッケージが存在します。

- DBMS_XMLSCHEMA_ANNOTATE：このパッケージにより、データベースに登録する前の XML スキーマのプログラムの注釈が可能になります。これにより、異なるバージョンの XML スキーマを使用する場合に、一連の一貫性のある注釈を使用することが容易になります。
- DBMS_XMLSTORAGE_MANAGE：このパッケージにより、XML ストレージの管理に関する共通タスク（コレクション表の名前の変更、オブジェクト・リレーショナル・ストレージに関する B ツリー索引の作成など）が簡素化されます。

まとめ


企業や組織は、取り扱う必要のある XML コンテンツの増加という問題に直面しています。これらの企業や組織は、リレーショナル・データに必要とされるものと同程度の厳密さとセキュリティで XML を管理できるプラットフォームを導入する必要があります。データの管理については、各種のコンテンツを管理するために個別の製品が使用される、'ストーブ・パイプ型'ソリューションを回避する必要があります。ストーブ・パイプ型のソリューションでは、データの共有が困難になるだけでなく、各ソリューションの取得と操作のために多額の追加コストが必要になります。

Oracle XML DB は、企業や組織の XML コンテンツを管理するために必要なすべての機能とパフォーマンスを提供します。Oracle XML DB とその他のテクノロジー（Oracle Text、Oracle Spatial and Graph など）により、すべてのミッション・クリティカルなデータを、可用性と信頼性に優れたスケーラブルな単一のプラットフォームすなわち Oracle Database 12c で管理できます。

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 © 2017, 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。0417

ホワイト・ペーパー・タイトル
2017年4月
著者：[オプション]
共著者：[オプション]