2

Getting Started with Oracle XML DB

Some preliminary design criteria are presented for consideration when planning your Oracle XML DB solution.

- Oracle XML DB Installation
  Oracle XML DB is installed automatically if Database Configuration Assistant (DBCA) is used to build Oracle Database using the general-purpose template.
- Oracle XML DB Use Cases
  Oracle XML DB is suited for any application where some or all of the data processed is represented using XML.
- Application Design Considerations for Oracle XML DB
  When planning an Oracle XML DB application it can be worthwhile to consider some preliminary design criteria.

Oracle XML DB Installation

Oracle XML DB is installed automatically if Database Configuration Assistant (DBCA) is used to build Oracle Database using the general-purpose template.

You can determine whether or not Oracle XML DB is already installed. If it is installed, then the following are true:

- Database schema (user account) `XDB` exists. To check that, run this query:

  ```
  SELECT * FROM ALL_USERS;
  ```

- View `RESOURCE_VIEW` exists. To check that, use this command:

  ```
  DESCRIBE RESOURCE_VIEW
  ```

> **See Also:**
>
> - Administration of Oracle XML DB for information about installing Oracle XML DB *manually*
> - *Oracle Database Security Guide*

Oracle XML DB Use Cases

Oracle XML DB is suited for any application where some or all of the data processed is represented using XML.

Oracle XML DB provides for high-performance database ingestion, storage, processing and retrieval of XML data. It also lets you quickly and easily generate XML from existing relational data. Applications for which Oracle XML DB is particularly suited include the following:

- Business-to-business (B2B) and application-to-application (A2A) integration
- Internet
- Content-management
- Messaging
- Web Services

A typical Oracle XML DB application has at least one of the following characteristics:

- Large numbers of XML documents must be ingested or generated
- Large XML documents must be processed or generated
- High-performance searching is needed, both within a document and across large collections of documents
- High levels of security are needed
- Fine-grained security is needed
- Data processing must use XML documents, and data must be stored in relational tables
- Programming must support open standards such as SQL, XML, XQuery, XPath, and XSL
- Information must be accessed using standard Internet protocols such as FTP, HTTP(S)/ WebDAV, and Java Database Connectivity (JDBC)
- XML data must be queried from SQL
- Analytic capabilities must be applied to XML data
- XML documents must be validated against an XML schema

Application Design Considerations for Oracle XML DB

When planning an Oracle XML DB application it can be worthwhile to consider some preliminary design criteria.

These include the following:

- The ways that you intend to store your XML data
- The structure of your XML data
- The languages used to implement your application
- The ways you intend to process your XML data

However, in general Oracle recommends that you start with the following Oracle XML DB features. For most use cases they are all that you need to consider.

- Storage model – compact schema-aware binary XML or transportable binary XML
- Indexing – XML search index, `XMLIndex` with structured component
- Database language – SQL, with SQL/XML functions
- XML languages – XQuery and XSLT
- Client APIs – OCI, thin JDBC, SQL .NET

- XML Data Storage
  There are several ways to store XML data in Oracle Database.
- The Structure of Your XML Data
  How structured your XML data is, and whether it is based on an XML schema, can influence how you store it.
- Languages Used to Implement Your Application
  You can program your Oracle XML DB applications in Java (JDBC, Java Servlets) or PL/SQL.
- XML Processing Options
  Oracle XML DB offers a full range of XML processing options.
- Oracle XML DB Repository Access
  Design considerations for applications that use Oracle XML DB Repository include access method, security needs, and whether you need versioning.
- Oracle XML DB Cooperates with Other Database Options and Features
  Oracle XML DB is an integrated part of Oracle Database, and works well with other database options and features.

Related Topics

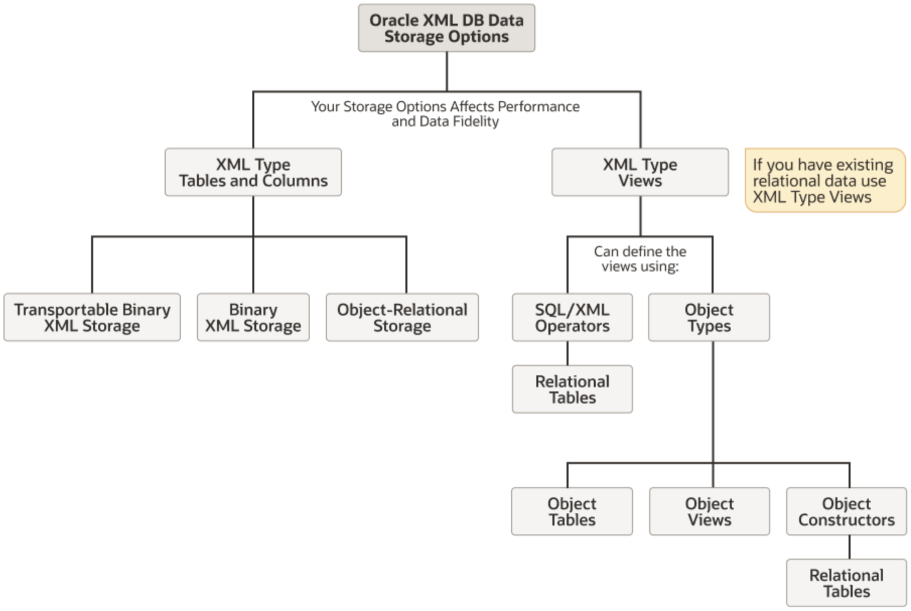
- Choice of XMLType Storage and Indexing
  Important design choices for your application include what `XMLType` storage model to use and which indexing approaches to use.
- SQL/XML Standard Functions
  Oracle XML DB provides the SQL functions that are defined in the SQL/XML standard.
- XQuery and Oracle XML DB
  The XQuery language is one of the main ways that you interact with XML data in Oracle XML DB. Support for the language includes SQL*Plus command`XQUERY` and SQL/XML functions `XMLQuery`, `XMLTable`, `XMLExists`, and `XMLCast`.
- Transformation and Validation of XMLType Data
  There are several Oracle SQL functions and `XMLType` APIs for transforming `XMLType` data using XSLT stylesheets and for validating `XMLType` instances against an XML schema.
- C DOM API for XMLType
  The C DOM API for `XMLType` lets you operate on `XMLType` instances using a DOM in C.
- Java DOM API for XMLType
  The Java DOM API for `XMLType` lets you operate on `XMLType` instances using a DOM. You can use it to manipulate XML data in Java, including fetching it through Java Database Connectivity (JDBC).
- Oracle XML DB and Oracle Data Provider for .NET
  Oracle Data Provider for Microsoft .NET (ODP.NET) is an implementation of a data provider for Oracle Database. It uses Oracle native APIs to offer fast and reliable access to Oracle data and features from any .NET application.

XML Data Storage

There are several ways to store XML data in Oracle Database.

Storage of `XMLType` tables and views is outlined in Figure 2-1.

Figure 2-1   Oracle XML DB Storage Options for XML Data


Oracle XML DB Data Storage Options
Your Storage Options Affects Performance and Data Fidelity
XML Type Tables and Columns
XML Type Views
If you have existing relational data use XML Type Views
Transportable Binary XML Storage
Binary XML Storage
Object-Relational Storage
Can define the views using:
SQL/XML Operators
Object Types
Relational Tables
Object Tables
Object Views
Object Constructors
Relational Tables

If you have existing relational data, you can access it as XML data by creating `XMLType` views over it. You can use the following to define the `XMLType` views:

- SQL/XML functions. See Generation of XML Data from Relational Data and XQuery and Oracle XML DB.
- Object types: object tables, object constructors, and object views.

Regardless of which storage options you choose for your application, Oracle XML DB provides the same functionality. Though the storage model you use can affect your application performance and XML data fidelity, it is totally independent of how frequently you query or update your data and what APIs your application uses.

Related Topics

- XMLType Storage Models
  `XMLType` is an *abstract* data type that provides different *storage models* to best fit your data and your use of it. As an abstract data type, your applications and database queries gain in flexibility: the same interface is available for all `XMLType` operations.

The Structure of Your XML Data

How structured your XML data is, and whether it is based on an XML schema, can influence how you store it.

If your XML data is *not* XML Schema-based, then, regardless of how structured it is, you can store it in an `XMLType` table or view as binary XML, or you can store it as a file in an Oracle XML DB Repository folder. You cannot store it object-relationally.

If your XML data is XML Schema-based then you must store it as binary XML or object-relationally.

Related Topics

- Choice of XMLType Storage and Indexing
  Important design choices for your application include what XMLType storage model to use and which indexing approaches to use.

Languages Used to Implement Your Application

You can program your Oracle XML DB applications in Java (JDBC, Java Servlets) or PL/SQL.

Related Topics

- Java DOM API for XMLType
  The Java DOM API for XMLType lets you operate on XMLType instances using a DOM. You can use it to manipulate XML data in Java, including fetching it through Java Database Connectivity (JDBC).
- Guidelines for Oracle XML DB Applications in Java
  Design guidelines are presented for writing Oracle XML DB applications in Java. This includes guidelines for writing and configuring Java servlets for Oracle XML DB.
- PL/SQL APIs for XMLType
  There are several PL/SQL packages that provide APIs for XMLType.
- PL/SQL Access to Oracle XML DB Repository
  PL/SQL packages DBMS_XDB_CONFIG and DBMS_XDB_REPOS together provide the Oracle XML DB resource application program interface (API) for PL/SQL. You use the former to configure Oracle XML DB and its repository. You use the latter to perform other, non-configuration operations on the repository.

XML Processing Options

Oracle XML DB offers a full range of XML processing options.

The following are available and should be considered when designing your Oracle XML DB application:

- XML Generation and XMLType views. Whether you need to generate (or regenerate) XML data. See Generation of XML Data from Relational Data.
- Whether your application is data-centric or document-centric, or both. See Overview of How To Use Oracle XML DB.
- DOM fidelity, document fidelity. XMLType storage, whether object-relational, compact schema-aware binary XML, or transportable binary XML, preserves *DOM fidelity*. That is, A DOM created from an XML document stored as XMLType is identical to a DOM created from the original document. However, there could be differences in insignificant whitespace. See DOM Fidelity, SYS_XDBPD$ and DOM Fidelity for Object-Relational Storage, and PL/SQL APIs for XMLType.

  If you need to preserve *document fidelity* (insignificant whitespace) in addition to DOM fidelity, then store two copies of your original document: one as an XMLType instance for database use and XML processing, the other as a CLOB instance to provide document fidelity.
- XPath searching. You can use XPath syntax embedded in a SQL statement to query XML content in the database. See Query and Update of XML Data, Access to Oracle XML DB Repository Data, and Repository Access Using RESOURCE_VIEW and PATH_VIEW.
- How often XML documents are accessed, updated, and manipulated. See Query and Update of XML Data.

- Whether you need to update fragments or whole documents. You can use XPath expressions to specify individual elements and attributes of your document during updates, without rewriting the entire document. This is more efficient, especially for large XML documents. See Updating XML Data.
- Which kinds of indexing best suit your application and data. See Indexes for XMLType Data.
- XSLT. Whether you need to transform the XML data to HTML, WML, or other languages, and, if so, how your application does this. While storing XML documents in Oracle XML DB, you can optionally ensure that their structure complies with (validates against) specific XML schemas. See Transformation and Validation of XMLType Data.

Oracle XML DB Repository Access

Design considerations for applications that use Oracle XML DB Repository include access method, security needs, and whether you need versioning.

There are two main repository access methods:

- Navigation-based access or path-based access. This is suitable for both content/document and data oriented applications. Oracle XML DB provides the following languages and access APIs:
  - SQL access through resource and path views. See Repository Access Using RESOURCE_VIEW and PATH_VIEW.
  - PL/SQL access using package `DBMS_XDB` or packages `DBMS_XDB_ADMIN`, `DBMS_XDB_CONFIG` and `DBMS_XDB_REPOS`. See PL/SQL Access to Oracle XML DB Repository .
  - Protocol-based access using HTTP(S)/WebDAV or FTP, most suited to content-oriented applications. See Repository Access Using Protocols.
- Query-based access. This can be most suited to data oriented applications. Oracle XML DB provides access using SQL queries through the following APIs:
  - Java access (through JDBC). See Java DOM API for XMLType.
  - PL/SQL access. See PL/SQL APIs for XMLType.

These options for accessing repository data are also discussed in Access to Oracle XML DB Repository Data.

You can also consider the following access criteria:

- What levels of security you need. See Repository Access Control.
- Whether you need to version the data. See Resource Versions.

Oracle XML DB Cooperates with Other Database Options and Features

Oracle XML DB is an integrated part of Oracle Database, and works well with other database options and features.

- Oracle Database Advanced Queuing (AQ) – merge XML payloads. See XML Data Exchange Using Oracle Database Advanced Queuing and *Oracle Database Advanced Queuing User's Guide*
- Oracle GoldenGate and Oracle Active Data Guard – replicate and safeguard XML data, or perform a rolling upgrade. See *Oracle GoldenGate* and *Oracle Data Guard Concepts and Administration*

- Oracle Exadata Storage Server Software – high-performance, scalable, and highly available use of XML data. See *Oracle Exadata Storage Server Software User's Guide*.
- Oracle Real Application Clusters (Oracle RAC) – Use XML data with clusters of database instances. See *Oracle Real Application Clusters Administration and Deployment Guide*
- Oracle Multitenant option – Use XML data with a multitenant architecture, where each pluggable database has its own Oracle XML DB Repository. See *Oracle Multitenant Administrator's Guide*
- Compression and Encryption – You can compress or encrypt binary XML data that uses SecureFiles LOB storage. For XML data stored object-relationally, you can compress or encrypt XML elements and attributes individually.
- Parallel Execution – Execution of the following operations can be carried out in parallel:
  - A query of `XMLType` data
  - DML for `XMLType` data stored as binary XML using SecureFiles LOBs
  - A direct load for an `XMLType` table on which an Oracle Text `CONTEXT` index is defined

> **See Also:**
>
> *Oracle Database Concepts*