ORACLE

XML DB and XML features in Oracle Database

Releases 19c and 21c,
Autonomous Database, Cloud and On-Premise

February 2022

Table of contents

Introduction to XML and Oracle XML DB

XML is a well-established way to persist and exchange business critical information. XML is an open standard, managed by the W3C, and under the control of no single vendor. Many industry segments have developed XML based standards for representing information. These standards are typically based on XML Schema, a W3C developed standard for defining the expected contents of a given XML document or file. XML based standards can be found in healthcare, financial services, manufacturing, publishing, law enforcement and the public sector. XML also provides the foundation for SOAP based application development. In many situations, government regulators mandate the use of such standards when exchanging information. Consequently, organizations often need to deal with large volumes of XML data and are forced to adopt XML platforms that manage XML data with a similar degree of rigor and security as conventional operational data.

> To meet this need, Oracle developed Oracle XML DB: a high-performance, native XML storage and retrieval technology that is part of all versions of Oracle Database. Oracle XML DB allows an organization to manage XML data and content in the same way that it manages traditional relational data allowing organizations to save costs and improve return on investment by using a single platform to manage and secure all of their mission critical data.

The W3C has developed a number of standards that are used when working with XML content:

XML Schema

As the complexity of the XML increases it becomes necessary to have a standardized language to describe what the expected content of an XML document should be. The XML Schema standard specifies a vocabulary that makes it possible to define a collection of type definitions and element declarations that accurately and unambiguously describe the content and structure of a class of XML documents. XML Schema defines many primitive data types that can be combined to form other more complex and user-defined objects.

XML Schema has been widely adopted since it allows organizations to accurately describe what information is being exchanged, and to validate that the information being exchanged conforms to the agreed specification. Many industry standards bodies have used XML Schema to define the persistence and exchange models needed to exchange information between customers, suppliers and partners.

XQuery and XPath

XQuery is the natural query language for XML content, in the same way that SQL is the natural query language for relational content. XQuery uses a superset of the W3C's XPath expression syntax to address specific parts of an XML document. It supplements this with a SQL-like "FLWOR expression which is constructed from the five clauses after which it is named: FOR, LET, WHERE, ORDER BY, RETURN. XQuery can also be used to create new XML documents. The language defines Element and Attribute constructors that can be combined together in nested structures to synthesize the required XML document.

XQuery-Update makes it possible to modify the content of an XML document without performing a complete transformation. XQuery update supports insert, delete, modify and rename operations. XQuery Full-Text and support for performing complex full-text searches of an XML document, with the ability to search for text at the document, fragment or node level.

XML Webservices and SOAP

The SOAP standard makes extensive use of XML to provide a service-based infrastructure A SOAP service is defined using a Web Services Description Language (WSDL) document. To invoke a SOAP service a SOAP client sends a request document to the SOAP server. The results of invoking a service are returned to the SOAP client using a response document. XML is used for both the request and

response documents. The WSDL is also an XML document. It specifies where the end-point for the service is located as well as providing the definition of the request and response documents. The WSDL uses an embedded XML Schema to provide this information.

XML Use-cases

XML has the advantage of being inherently self-describing and is human and machine readable. Its self-describing nature makes it a very good for representing dense data as well as sparse or extremely variable data. The XML data model is also highly extensible, allowing organizations to easily customize XML content models to meet specific information storage and retrieval requirements. The primary use cases for XML are

- Data capture: XML is used to store the data generated by sensors and loggers
- Data Exchange: XML is used to exchange information between loosely coupled systems.
- XML Persistence: Persistence based on industry-standard data models
- XML Persistence: Persistence of application objects, meta-data and state.
- XML Persistence: Persistence of content created using popular productivity software.
- XBRL financial reporting and analytics

XML for Data Capture

Data generated by sensors and loggers, or by application logging is stored as XML. In scenario an extremely large volume of XML is generated in a relatively short time frame. The XML may be stored as a small number of very large files or a very large number of relatively small files. This data needs to be integrated into the business's application processes

XML for Data Exchange

In this scenario systems use XML messages to communicate with each other. XML is generated from (typically relational) data managed by one system, transported to some other location and then ingested into the (typically relational) data stores managed by the other system. The use of XML as an exchange mechanism provides an abstraction layer that allows one application system to re-organize its data without impacting any applications that require access to that data.

In some models, such as SOAP based messaging, extremely large numbers of small (4Kb-100Kb) XML documents are exchanged in near real-time. In other cases, small to medium volumes of large (100Kb to10+GB) XML files are created and processed using more traditional batch generation and ETL processing techniques. Often these messages are compliant with industry standard XML Schemas that have been developed by Industry standards bodies.

XML Persistence for Industry Standard XML data-models

One of the drivers for the growth in XML persistence has been the emergence of industry-based XML standards. As these standards evolve, they define extremely complex and highly variable information models in-order to meet the needs of all of the constituent organizations they serve. Analysis has shown that some of these XML Schemas describe models that would translate in relational data models containing 1000's of tables. As the degree of complexity and variability in the model increases, the cost and time required to develop software that performs bi-directional translation between the relational model and the XML model becomes prohibitive.

XML for Office Productivity and Technical Authoring Software

The flexibility and variability inherent in the XML data model makes it an attractive mechanism for storing content generated using Office Productivity Suites and Technical Authoring software. This is not surprising, given that the precursor of XML was SGML, a standard designed explicitly for this purpose. The drive towards interoperability resulted in the development of open source, XML based standards. The DOCX, XSLX and PPTX file formats used by Microsoft Office are examples.

The adoption of XML as the primary method of persisting office productivity documents created XML-based content management (CM) systems. These systems differentiate themselves from traditional CM solutions by being able to understand both the metadata and the content of the documents that they manage, allowing their users make much more effective use of the volume of information that was traditionally trapped in the documents produced by previous generations of desktop productivity software.

XBRL for financial reporting and analytics

XBRL is the open international standard for digital business reporting. XBRL provides a language in which reporting terms are defined and then used to represent financial statements or other kinds of compliance and business reports. XBRL relies on XML and XML Schema to define rules and semantics. XBRL provides significant benefits in the preparation, analysis, and communication of business information. With growing adoption of XBRL, and with financial reports being generated on a regular basis, there is a growing volume of XBRL content to be stored, managed, and queried.

Introduction to Oracle XML DB

Oracle XML DB is a high-performance, native XML storage and retrieval technology that is delivered as a part of all versions of Oracle Database. Oracle XML DB provides full support for all of the key XML standards, including but not limited to XML, XML Schema, Namespaces, DOM, XQuery, SQL/XML and XSLT. By providing full support for XML standards, Oracle XML DB supports native XML application development. Developers are able to use XML-centric techniques to store, manage, organize, and manipulate XML content stored in the database. Oracle XML DB also supports the SQL/XML standard, which allows SQL-centric techniques to be used to publish XML directly from relational data.

The key features of XML DB include

- Efficient XML persistence
- Standards compliant query and update operations
- Powerful and flexible XML indexing
- Support for XML-centric, SQL-centric and Document-centric development and analytics
- XML and SQL interoperability, for example. to run SQL analytics over XML data.
- Deep integration with other Oracle Database features including but not limited to security, availability, performance, partitioning, multi-tenant, Golden Gate, RAC, Exadata, etc.

Binary XML Storage

XML content is stored in the database using a dedicated data type: *XMLType* which uses an efficient binary encoding schema: *Binary XML* is stored on disc using Oracle's SecureFile Lob infrastructure which delivers the maximum possible throughput for storage and retrieval. Binary XML also leverages the SecureFile sliding insert feature, allowing efficient, node-level, insert, update and delete operations on XML content. If further storage savings are desired, Binary XML can be compressed using the compression features of the Oracle Database.

Wherever possible operations on Binary XML are performed using streaming techniques. Streaming techniques improve performance by avoiding the memory and CPU overhead associated with traditional DOM based processing of XML content. Streaming is supported during ingestion for both parsing and encoding of XML content as well as queries and updates. This allows the database to efficiently evaluate complex X-Path expressions against large volumes of XML. Multiple leaf nodes or fragments can be extracted in a single pass of the XML document. Documents associated with different XML Schemas, or versions of an XML Schema can be stored in the same table or column making the storage very schema-flexible.

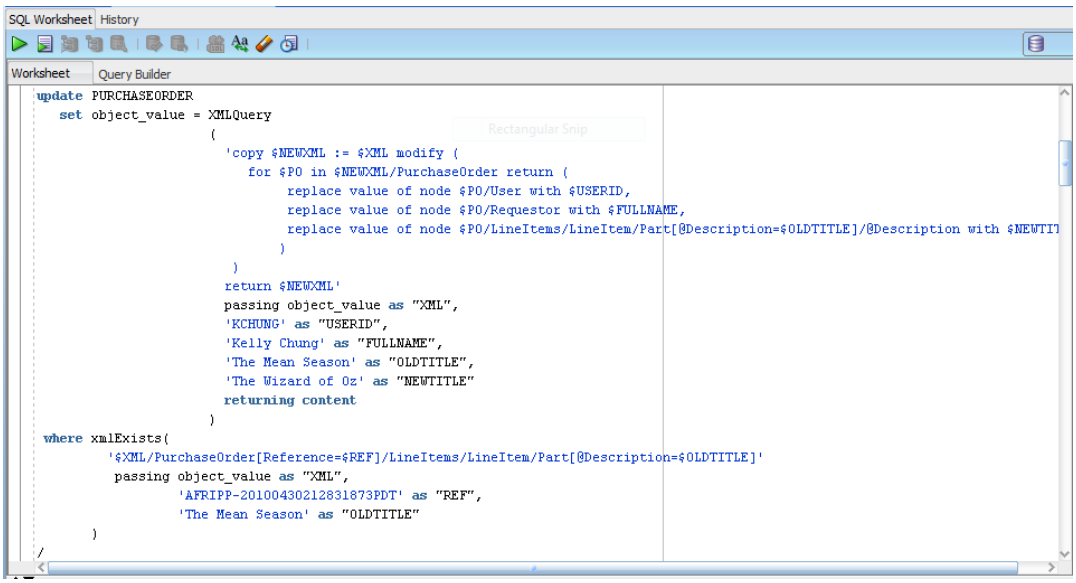
XML over relational data

With XML DB it is possible to use SQL/XML and XQuery to create views which expose relational data as XMLType values. These XMLTypes can then be queried using XQuery and X-Path in exactly the same manner as XMLTypes that make use of Binary XML storage. In most cases, updates to these views can be handled using instead of triggers.

Standards-based XML Query and Update

Oracle XML DB provides comprehensive support for *XQuery* and *XPath* to query and manipulate XMLType values stored in the Oracle Database. The *SQL/XML* extension to the SQL standard defines a set of operators, *XMLQuery*, *XMLTable* and *XMLExists* that allow XQuery operations to be executed in the context of a SQL statement. This allows XML operations to use the same transaction semantics as other database operations. The SQL/XML extension also defines the *XMLCast* operator which allows translation between the primitive types defined by XML Schema and the scalar types supported by SQL. The W3C standard *XQuery-update* is supported.

Oracle XML DB also provides DOM Level access and manipulation of XML content via the *DBMS_XMLDOM* and *DBMS_XMLPARSER* packages. XSL transformation of XML content is supported via the SQL operator *XMLTransform* and the *DBMS_XSLTRANSFORM* PL/SQL package.



XQuery Update operation as part of SQL Update statement

XML Indexing

Oracle's Binary XML format is designed to enable efficient indexing of XML content. The binary XML format was designed from the ground up to allow optimization of XQuery operations which result in fragment and leaf level access to XML content. There are two distinct indexing techniques:

- Structured XML Index
- XML Full-Text Index

Structured XML Index is focused on use cases where the structure of the XML, and the set of queries that will be executed are well known. Behind the scenes, Structured XML Index projects the data required to answer the queries into a set of relational tables. When an XQuery an expression is executed, the XQuery processor executes a relational query over these tables to determine which XML documents satisfy the XQuery expression XQuery expressions that involve leaf level extraction, can also be optimized by extracting data directly from the index. SQL/JSON operator like XMLTable (allowing relational access to XML data) have been optimized to leverage indexes.

The XML Full-text index optimizes XQuery Full-Text operations on XML documents:

XQuery Full-Text and XML Full Text Indexing

XML can be used to represent many different kinds of information: from highly structured data (fixed schemas) through semi-structured data (flexible schemas) to markup data (text with embedded tags) and unstructured data (untagged free-flowing text). As the XML becomes less structured the need to be able to search using Information Retrieval (IR) techniques grows and Oracle XML DB provides an XQuery implementation with support for XQuery-Full Text):

A full-text search looks for tokens and phrases rather than substrings. For instance a substring search for items that contain the string "sport" will return an item that contains "Transport". A full-text search for the token "sport" will not. Full-text search also needs to support language-based searches. An example of a language-based search is "find me all items that contain a token with the same linguistic stem as 'sport'" (finds "sport" and "sports" but not "transport") or "find me all items that contain the tokens Oracle and Database within 3 tokens of each other."

With the support of the structured index and a full text index Oracle allows to efficiently query the full spectrum of XML documents - from structured to unstructured.

Oracle XML DB XBRL Extension

The XBRL Extension to Oracle XML DB extends Oracle Database to serve as a comprehensive platform for managing XBRL content. It lets you create XBRL repositories and project XBRL data relationally or query it in various ways. It can help you improve operations on aggregated business and financial reports such as extraction, transformation, and loading (ETL); business intelligence (BI); and online analytical processing (OLAP). The XBRL Extension to Oracle XML DB provides the following features.

- Native database storage of XBRL data.
- Database enforcement of integrity, based on XBRL rules.
- Ability to query XML data using XBRL semantics.
- Relational representation of XBRL content. Ability to expose XBRL content to relational applications and SQL queries.
- PL/SQL transforming procedures that generate derived XBRL views based on XBRL relational representations, network generation APIs, or dimensional information.
- Scalable XBRL services: reports, network generation, transformations.
- Online analysis based on XBRL dimensions, both explicit and typed.
- Integration with Oracle Business Intelligence Suite Enterprise Edition (OBIEE).

XML DB Support on Oracle Autonomous Database

Binary XML, XML indexing, XQuery, etc are fully supported on Oracle Autonomous Databases. Limits are listed here: https://docs.oracle.com/en/cloud/paas/autonomous-database/adbdl/

XML DB Documentation

https://docs.oracle.com/en/database/oracle/oracle-database/19/adxdb/ (Release 19c)

https://docs.oracle.com/en/database/oracle/oracle-database/21/adxdb/ (Release 21c)

https://www.oracle.com/database/technologies/appdev/xmldb.html

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 blogs.oracle.com  facebook.com/oracle  twitter.com/oracle