

XML QUERY (XQUERY) SUPPORT IN ORACLE DATABASE 10G RELEASE 2

KEY FEATURES

NEW IN ORACLE 10G RELEASE 2

- A native XQuery engine built into the database
- Based on the latest (April 2005) draft of the W3C XQuery standard
- XQueries can be issued against content in the database (XMLType tables, Relational tables), files, HTTP servers, etc.
- Fully transactional, just like the SQL engine
- Scalable and RAC-able
- Supports several Oracle extensions, e.g. for Full-Text search functionality that is currently lacking in XQuery.
- Can combine SQL and XQuery in the same query

DATASOURCES

- XMLType Tables
 - Local or remote XML Views
 - Files in the XML DB Repository
 - Files on local or remote file-systems
 - Relational Tables
 - Local or remote Relational Views
 - XML/XHTML documents accessible through Web Services
 - RSS feeds
 - Your own repository accessed via a customized connector
 - J2EE Connector Architecture (JCA) sources accessed via Oracle iAS (using the mid-tier XQuery engine and the XML Query Services)
- ... and more.

HIGH PERFORMANCE AND SCALEABILITY

- Native XQuery engine written in C integral with the database kernel
- Integrated with the database query optimizer and execution

XML Query (XQuery) is a standards effort undertaken by the World Wide Web consortium to enable data to be extracted from XML documents. XQuery is designed to work with the XML data model, and be a comprehensive query language for data that is expressed in XML – just as SQL has been the query language for much of the world’s structured data expressed as relational tables, and as keyword searches have powered much of the information access on the Internet. XQuery is also useful for constructing XML data and documents as the return result of query expressions.

Oracle Database 10g Release 2 introduces native support of XQuery. In addition, Oracle also supports a mid-tier XQuery engine that is well integrated with the database XQuery capabilities. Using these, you can

- *Search with any language – XQuery/ XPath/ XSL/ Full-Text as well as SQL*
- *Search anywhere – mid tier or backend*
- *Search anything – files, relational content, message, web-service etc.*
- *... and search everything – structured tables, semi-structured XML, unstructured documents*
- *Search any size - a native, scalable implementation*
- *Search any time - an unbreakable solution, built on Oracle’s robust enterprise platform.*

Storing and Retrieving XML

With Oracle Database 9i Release 2, we introduced a native XML store – XML DB – integrated with and part of the traditional Oracle Database. XML DB provides a high-fidelity storage and retrieval of XML documents, and is useful for a certain classes of data-management tasks, viz.:

Content Management – standards-based life-cycle management of content expressed in XML (e.g. technical manuals, multi-media messages, legal statutes, regulatory filings)

Exchange and Storage – generation of template-based business documents expressed as XML (e.g. purchase orders, bills, invoices, reports) to exchange between applications, and storage of such documents natively in the database

Data Integration – querying across different types of information assets – database records, files, web servers, news feeds.

With Oracle Database 10g Release 2, the XML DB technologies are ideal for a variety of applications which incorporate the above tasks, such as:

- Product Documentation and Technical Manuals
- Product Specification and Design Documents
- Web Site Content
- News and Information Services

engine

- Directly generates sub-SQL execution primitives
- Oracle Real Application Clusters for scale-out

FLEXIBLE, EASY-TO-INTEGRATE QUERY FRAMEWORK

- Integrated with SQLPlus, XDK, XML DB and Database Java Virtual Machine
- Supports Oracle extensions for Full-Text search
- Mid-tier version of the XQuery engine tracking XQJ

AREAS OF LIMITED SUPPORT

- Pending finalization of the W3C XQuery Recommendation, refer to the documentation on some minor limitations in the proposed XQuery functions and operators, certain boundary conditions, implicit time zones etc.
- Till XQuery becomes a W3C recommendation Oracle's XQuery implementation in future versions may not be backward compatible with Oracle Database 10g Release 2.

- Legislative Documents
- Contracts
- Financial Reporting
- Document Publishing.

From the perspective of developers building such applications, the initial use of XML as a storage format is broadening into the need to query the stored XML. The common reasons typically cited for using XQuery to query XML are:

Conciseness and simplicity – it is estimated that accessing XML data through XQuery results in one-fifth the code of XSLT-oriented approaches and one-twentieth the code of DOM-based approaches

Heterogeneous queries – since XML can map to a large number of different data models, one could search databases, files, web-services simultaneously using XQuery.

XML data model – XQuery is based on XML Schema. The XML Schema-based data model is best suited for representing variable, unpredictable, and irregular – i.e. ‘semi-structured’ data. When querying such data, XQuery is an obvious choice.

XML construction – XQuery can construct XML as the result of evaluating query expressions, in many cases more expressively and efficiently than XSLT.

Database or Mid-Tier?

Since databases already have a rich (SQL) query execution capability, it is often wrongly surmised that XQuery is needed only in the middle-tier.

Certainly, when you want to integrate heterogeneous data sources, a middle-tier approach is feasible in that you have to address non-database sources. In practice, however, one typically sees that the organizations’ important data lives best in databases – either relationally, or as content in CM systems, or as persistent messages (email etc.) The advantages of transactional consistency, high availability, cluster-based scale-out, secure backup and recovery, auditing – are not available if your data is left in files. The evolutionary arrow of content- and message-management does lead firmly towards databases.

Further, one of the major objections of storing anything other than structured data in a database was that databases could only perform relational algebra; with XQuery support, combined with support for the XML data model as well as an XML Repository inside the database, Oracle has become the leading platform for storing all your data.

Operational data and content is best stored in databases, modeled as XML or SQL, and queried by whichever metaphor is more appropriate. In cases where there are relatively smaller amounts of metadata, configuration files or in-memory cached data, a lightweight mid-tier XQuery engine might be more appropriate. Oracle provides both a database and a middle tier implementation of XQuery, and the latter covers all the lightweight query requirements of application-server-based queries.

Database XQuery Use-Cases

There are four typical ways to use XQuery in the database:

Querying XML Documents in the Oracle XML DB Repository: Increasingly, Content

Management applications seek to migrate their content stores from file-systems to databases. Databases provide transactional robustness, superior scalability, and, with XQuery, a natural way to query documents stored in Oracle. This usage of XQuery typically dovetails with XML Content Management applications of Oracle XML DB.

Querying a Relational Table or View as if it were XML data: XML must be frequently generated from relational data for purposes of Exchange. In Oracle 9i Release 2 and Oracle 10g Release 1, we provided SQL/XML operators to help generate XML from relational data. With Oracle Database 10g Release 2, you can also use the Oracle XQuery function *ora:view* to create an XML view over the relational data, on the fly.

Querying native XMLType data in Oracle XML DB: Applications often need to store XML with high fidelity – i.e. not lose any ordering, namespace etc. The XML data model may contain information that is not supported in the basic relational model. Oracle XMLDB provides a native XMLType to store XML with DOM-fidelity or whitespace-fidelity, depending on your needs. For use-cases involving XML Storage, XQueries need to operate on natively stored XMLType data.

Querying across Databases and Filesystems, or Websites: In certain cases, data in the database may refer to content stored in a file-system, perhaps through an URL. Where you control all the data, it will always help to consolidate the file content into the XML DB Repository, converting this case to first case (Querying XML Documents in the Oracle XML DB Repository.) However, in some cases you may not have enough control over the data to move it into your database – as when dealing with a URL on a partner site. Oracle's XQuery implementation can do the following: (i) call out from the database (ii) read in a file using the *doc* function and thereafter (iii) perform a query across both data in the database and a document read in from an external source. This is akin to Data Integration.