19c ORACLE Database

Multimodel Database

May 2019

ORACLE

Purpose Statement

This document provides an overview of features and enhancements included in release 19c. It is intended solely to help you assess the business benefits of upgrading to 19c and to plan your I.T. projects.

DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

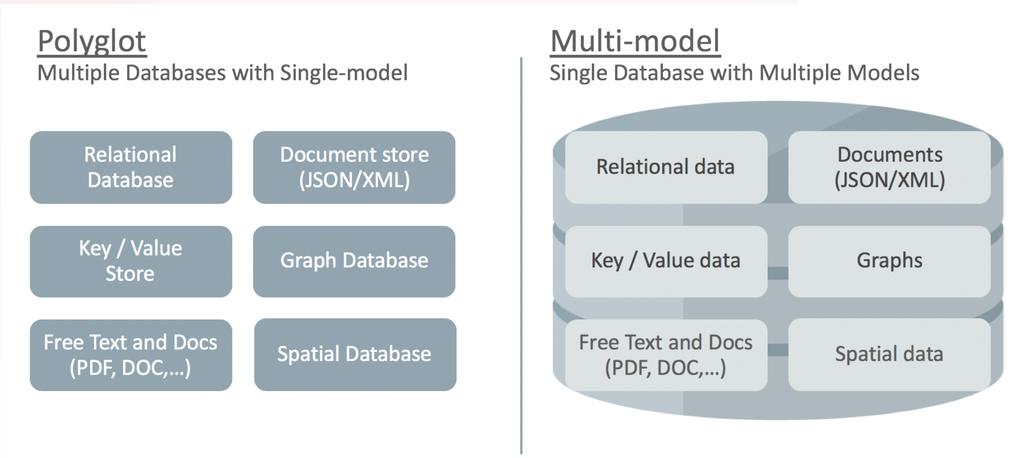
TABLE OF CONTENTS

INTRODUCTION

Over the nearly 40 years in the evolution of commercial relational database management systems, a consistent pattern has emerged as the capabilities, data types, analytics, and data models have been developed and adopted. With each new generation of computing architecture – from centralized mainframe, to client server, to internet computing, to the Cloud – new generations of data management systems have been developed to address new applications, workloads and workflows.

Today, the successful operation of corporations, enterprises, and other organizations relies on the management, understanding and efficient use of vast amounts of unstructured Big Data that may come from social media, web content, sensors and machine output, and documents. Traditional business applications – finance, order processing, manufacturing, and customer relationship management systems – that easily conform to standard data structures (such as rows and columns with well-defined schemas) also contribute to Big Data analysis. Increasingly, deriving business value for successful operations depends on management, analysis and understanding of information that is not readily accessible without human or machine-based interpretation. Common examples range from documents, XML, JSON, multimedia content, and web content to specialized information such as satellite and medical imagery, maps and geographic information, sensor data, and graph structures.

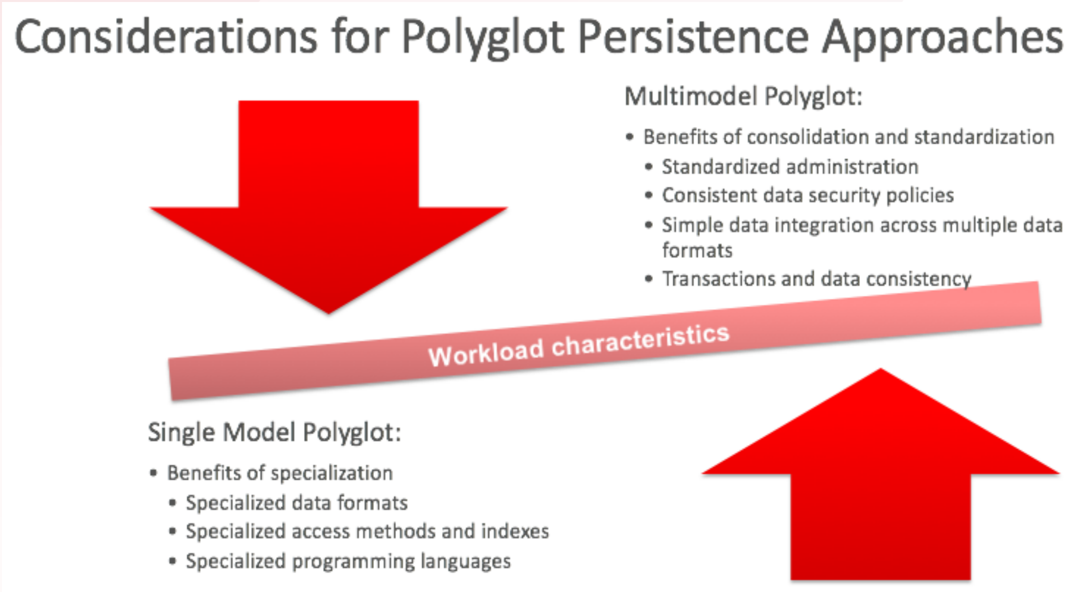
MULTIMODEL DATABASE ARCHITECTURE



The idea of having specific data models that address the needs of specific classes of applications has existed since the earliest days of computing. Transactional workloads (OLTP) are supported by data models that differ from those used in analytic workloads (OLAP). Document and multimedia data rely on formats like XML and JSON. Graph databases, spatial databases and key-value stores are used for connectivity analysis, geographic analysis and high- performance lookup, respectively. The concept that different database models are better suited to address the needs of different applications is now referred to as “Polyglot Persistence”.

One way to address these polyglot requirements is to have separate products that implement a specific database model to address specific applications. Examples of this include Oracle offerings such as Berkeley DB as a Key-Value store, Oracle NoSQL Database as a Key-Value and sharded database, Oracle TimesTen as an In-Memory Database, and Essbase for analytic processing. Numerous other open source and proprietary products are also available to support this single model Polyglot Persistence approach.

As commercial enterprise relational databases have developed over time, they have encompassed multiple data models and access methods within a single database management system. This concept is called Multimodel Polyglot Persistence and it allows many applications to use the same database management system while continuing to benefit from the unique data model necessary for a specific application.



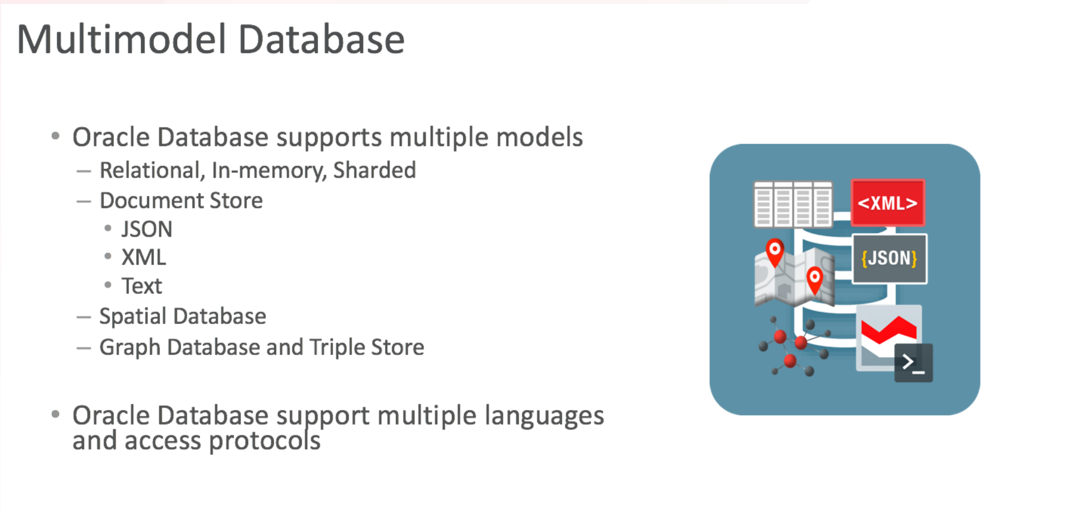
MULTIMODEL DATABASE FEATURES IN ORACLE 19c

Oracle Database 19c is a multimodel database management system that supports data formats, access methods and indexes, and programming languages required by specific workloads while exploiting common administration, security policies, and transactional and data consistency. It includes an in-memory columnar store, a sharded database model, as well as a document store, spatial database and graph database. This multimodel approach simplifies data integration across multiple data formats.

The ways in which these data models are managed in Oracle Database 19c vary based on how the data are created and used:

- Huge volumes of data in desktop office systems (documents, spreadsheets and presentations) and specialized workstations and devices (geospatial analysis systems and medical capture and analysis systems)
- Multi-terabyte archives and digital libraries in government, academia and industry
- Image data banks and libraries used in life sciences and pharmaceutical research
- Public sector, telecommunications, utility and energy geospatial data warehouses
- Integrated operational systems including business or health records, location and project data, and related audio, video and image information in retail, insurance, healthcare, government and public safety systems
- Graph data used in social networks, sensor analysis, recommendation systems, fraud detection, academic, pharmaceutical and intelligence research and discovery applications

For decades, Oracle database technology has been used to address the unique problems encountered when managing large volumes of all forms of information. Databases are often used to catalog and reference documents, images and media content stored in files through “pointer-based” implementations. To store this data inside database tables, Binary Large Objects, or BLOBs have been available as containers. Beyond simple BLOBs, Oracle Database has also incorporated a range of data models, intelligent data types and optimized data structures with operators to analyze and manipulate JSON and XML documents, multimedia content, text, graph, and geospatial information.



Organizations choose to store all forms of information in their Oracle database for many reasons:

- Robust Administration, Tuning and Management: Content stored in the database can be directly linked with associated data. Metadata and content are maintained in sync; they are managed under transactional control. The database also offers robust services for backup, recovery, physical and logical tuning.
- Simplicity of Application Development: Oracle's support for a specific type of content includes SQL language extensions, PL/SQL and Java APIs, as well as algorithms that perform common or valuable operations through built in operators. For certain content, Oracle Database includes specific query languages such as SQL, XQuery for XML, SPARQL for RDF graphs, DICOM access commands for medical imagery, and REST services to access database tables and JSON objects.
- High Availability: Oracle's Maximum Availability Architecture makes "zero data-loss" configurations possible for all data. Unlike common configurations where attribute information is stored in the database with pointers to unstructured data in files, only a single recovery procedure is required in the event of failure.
- Scalable Architecture: In many cases, the ability to index, partition, and perform operations through triggers, view processing, or table and database level parameters allows for dramatically larger datasets to be supported by applications that are built on the database rather than on file systems.
- Security: Oracle Database allows for fine-grained (row level and column level) security. The same security mechanisms are used for all forms of information. When using many file systems, directory services do not allow fine-grained levels of access control. It may not be possible to restrict access to individual users; in many systems enabling a user to access any content in the directory gives access to all content in the directory.

Oracle Database 19c includes Property Graph database and analytic features and a sharded database model, as well as enhancements to the NoSQL-style JSON store, the XML services, the Text analytics, the Spatial database capabilities and the RDF graph database features.

JSON IN ORACLE DATABASE

Modern application developers are choosing to store application data as documents instead of using entity relationship models backed by relational storage. The primary driver for this switch is the flexibility offered by JSON or XML based storage. This flexibility has enabled the application developer to be much more responsive to the needs of the business, as changes to the application data model no longer require changes to the database schema. This allows applications to be deployed and updated on a much faster cycle. The switch to document- based persistence has led to the adoption of NoSQL document stores for data persistence.

Oracle Database 19c has been engineered to provide full support for this style of application development. The Simple Oracle Document Architecture (SODA) specification, introduced as part of Oracle Database 19c, describes an extremely simple API that allows the Oracle database to be used as a JSON document store. The SODA API provides support for creating and dropping document collections, create, retrieval, update and delete (CRUD) operations on documents, List and Query by Example (QBE) operations on document collections and various ancillary operations such as bulk insert and indexing. SODA allows application developers to create and deploy applications that manage data using JSON documents without any knowledge of SQL, JDBC and without requiring any assistance from an Oracle DBA. In addition to introducing SODA the database itself is now capable of enforcing JSON validity, indexing JSON content, and using these indexes to optimize operations on JSON content.

Choosing SODA allows application developers to get all the benefits of JSON based persistence without losing any of the benefits of Oracle's data management platform. It allows organizations to adopt NoSQL style development without introducing the complexity of having to manage multiple databases. They can continue to rely on the Oracle Database to provide them with high availability, scalability, security and recovery.

The other major benefit of choosing to use Oracle Database 19c as a NoSQL-style JSON document store is that you still have all the power of SQL when you need it. Application developers can create and deploy their applications without any knowledge of SQL using Query-by-Example techniques to query the application data. However, when it becomes time to use the data captured by the application in ways other than were envisaged by the application developer (ad-hoc queries) or perform reporting or analytics on the information contained in the JSON documents, Oracle Database 19c allows SQL to be used for this purpose.

Oracle Database 19c extends the SQL language allowing JSON documents to be queried as part of SQL operations. These extensions allow the full power of SQL to be applied to the content of your JSON documents in a simple and straightforward manner. They also enable join operations between JSON documents and join operations between JSON documents and all the other kind of content managed by the Oracle Database, including relational data, XML content, Spatial Content, Semantic Content and Text Content.

Oracle Database 19c for Oracle Cloud also includes the Oracle Data Guide for JSON, an exciting feature that helps with understanding the structure of the JSON documents the database is managing. The Oracle Data Guide for JSON dynamically tracks the structure of JSON documents, allowing you to easily generate relational views over your JSON documents that enable programmers and tools that have no understanding of JSON to work directly with your JSON documents.

Oracle Database 19c provides JSON support in the Database with these features:

- Generating JSON document directly from relational data.
- Partial update operations on JSON documents allowing a programmer to change the content of specific parts of a JSON document.
- Oracle In-Memory database and Oracle Exadata to optimize query operations on JSON documents.
- Using Oracle Spatial and Graph to query GeoJSON objects embedded in JSON documents and the use of JSON objects to store, index, and manage geographic data that is in JSON format.
- Conversion of and Oracle Spatial and Graph SDO_GEOMETRY object to a JSON geometry object, and geometry JSON object back to an SDO_GEOMETRY object.

PROPERTY GRAPH DATABASE AND ANALYTICS IN ORACLE SPATIAL AND GRAPH

Oracle Database has included special-purpose graph database capabilities for over a decade. The Network Data Model graph is widely used by government, utilities, energy and telecommunications organizations to model and perform connectivity-based analysis on spatial networks like roads, pipelines, and infrastructure. The standards- based RDF graph allows statistics bureaus, financial institutions, life sciences and pharmaceutical companies and government agencies to use semantic web capabilities to publish and share information as part of “linked data” initiatives.

As enterprise applications incorporate and analyze social network data, sensor and Internet of Things (IoT) information to discover patterns, relationships and anomalies, general-purpose property graph databases are becoming part of the information technology landscape. With Oracle Spatial and Graph, we have incorporated into Oracle Database a high performance, massively scalable property graph database. Unlike other property graph offerings, it includes dozens of built-in, powerful, parallel, in-memory analytics to simplify the work of software developers and data scientists and make them more efficient. These algorithms include ranking, centrality, recommendation, community detection, and path finding for social network, fraud detection, upsell and cross-sell, influencer detection, and churn analysis applications.

As part of Oracle Database, the graph model resides in database tables and can be queried and filtered using SQL or a variety of supported APIs. To perform advanced analysis, the graph is loaded into memory where in-memory analyst (PGX) algorithms are applied. The analytics can either be executed within a Java application or executed in the multi-user, multi-graph in-memory analyst server environment on Oracle WebLogic Server, Apache Tomcat or Eclipse Jetty. The output of graph analysis can be another graph, such as a bipartite, filtered, undirected, sorted or simplified edges graph.

The property graph algorithms can be invoked by Oracle R Enterprise (a feature of Oracle Advanced Analytics to perform statistical analysis in Oracle Database). Graph data can be indexed using Oracle Text indexing; text queries are automatically translated into SQL SELECT statements with a "contains" clause. Graph data can be queried with SQL, and graph queries can include spatial filtering (such as finding results within a certain distance of a location). Multi-level security can be enforced with graph level access control, and Oracle Label Security can be used for fine-grained access control to individual graph elements.

Oracle Spatial and Graph 19c property graph capabilities include:

- Enhancements to PGQL, a SQL-like declarative language for querying graph data stored in Oracle Database and finding in-memory subgraph instances that match a given query pattern, including richer support for PATH queries and scalar sub-queries.

- New property graph analytics are supported for SQL-based collaborative filtering that can help make recommendations.
- TinkerPop3 API support.
- REST API and additional APIs for In-memory analyst (PGX).
- More in-memory analytics have also been added, including new variants of Infomap, Pagerank, a Personalized SALSA for making recommendations, K-Core for finding subgraphs by properties, Diameter, Radius, and Eccentricity to analyze distances in a graph, and PRIM for finding the minimum spanning tree of edges connecting all vertices in an undirected graph.
- Support for undirected graphs, Apache Zeppelin and an execution and scheduling manager to better control in-memory analyst tasks and resources.

SPATIAL DATABASE AND ANALYTICS IN ORACLE SPATIAL AND GRAPH

Oracle Spatial and Graph delivers the most comprehensive spatial database offering available in the industry today, including the highest performance native support for vector and raster analysis operations, topology and network models, 3D data, geocoding, routing, and OGC-standard Web Services. It is designed to meet the advanced geospatial requirements of the largest enterprise business and government applications such as business intelligence, land management, telecommunications, utilities, defense, and homeland security, using parallel architectures, in-memory spatial indexes, and hundreds of spatial algorithms, operators and functions. With open, native spatial support, Oracle Spatial and Graph eliminates the cost and complexity of separate, proprietary systems while enabling the use of all leading GIS tools. This extends Oracle's industry-leading security, performance, scalability, and manageability to mission critical spatial assets. It is the choice of the most demanding GIS and geo- enabled applications in the world.

With Oracle Database 19c, Oracle Spatial and Graph includes features for working with microservices and Big Data data sets for cloud and sensor-based applications, including:

- Native support for GeoJSON and of JSON objects to store, index, and manage geographic data that is in JSON format. Conversion of and Oracle Spatial and Graph SDO_GEOMETRY object to a JSON geometry object, and geometry JSON object back to an SDO_GEOMETRY object.
- Location data enrichment services tag large collections of documents with location references. The process associates authoritative location terms (place names, addresses, and latitude / longitude) to text found in database tables. Users can perform spatial and text analysis on these enriched text sources.
- A map visualization component enables developers to incorporate highly interactive maps and spatial analysis into business applications. Application content can be combined with maps and data from a variety of web services and data formats such as GeoJSON. This HTML5 map visualization is deployed in a Java EE container or in Oracle Java Cloud Service.
- A Location Tracking Server provides fast, continuous location monitoring of thousands of objects within a tracking network, in the database.
- Spatial support for distributed and Oracle XA transactions used in large scale web and cloud-based applications with distributed architectures
- Oracle REST Data Services support for spatial operations in Oracle Database for modern RESTful development

- Support for sharded databases with spatial data types
- More spatial web services features for scalability and usability

RDF SEMANTIC GRAPH TRIPLE STORE FEATURES IN ORACLE SPATIAL AND GRAPH

The RDF Semantic Graph feature of Oracle Spatial and Graph is a special purpose graph for linked data and semantic web applications conforming to World Wide Web Consortium standards, common in health sciences, finance, media, and intelligence communities. Oracle delivers advanced RDF Semantic Graph data management and analysis with native support for the Resource Description Framework (RDF) and Web Ontology Language (OWL) standards for representing and defining semantic data and SPARQL, a query language designed specifically for graph analysis. Application developers benefit from the industry's leading open, scalable graph data platform and its fine-grained security.

Application developers can add meaning to data and metadata by defining a set of terms and the relationships between them. These sets of terms ("ontologies") enable query, analysis and actions based on semantic content, rather than simply data values. Ontologies are used to build applications that utilize domain-specific knowledge. Ontological data sets, often containing 100s of millions of data items and relationships, can be stored in groups of three, or "triples" using the RDF data model. Oracle enables scaling to billions of triples to meet the needs of the most demanding applications.

RDF graph analysis enables discovery of relationships across data sets and documents and integration and access by applications to systems with disparate metadata.

Oracle Spatial and Graph RDF Semantic Graph features include:

- RDF Views on Relational Tables removing the need to duplicate data and the associated storage previously required to perform RDF graph queries on relational data sets. Semantic graph queries on RDF views can integrate relational data and RDF Semantic Graph triple data stored in Oracle. Semantic queries on these views can be written in the SPARQL query language or by embedding SPARQL in an Oracle SQL SEM_MATCH table function.
- RDF Semantic Graph "Named Graph" support as defined by the World Wide Web Consortium (W3C).
- Support for Analytic Operations and Tools. RDF Semantic Graph supports SPARQL 1.1 path expressions for simple and complex paths. RDF Semantic Graph can also be used in conjunction with the Network Data Model Java API to provide fast in-memory graph analytics, including shortest path, reachability, within-cost, and nearest-neighbor analysis of RDF graphs. Results from graph queries can be materialized as views for use with Oracle Advanced Analytics to enable the use of Oracle Data Mining clustering, classification, regression, anomaly detection, and decision tree algorithms as well as Oracle R Enterprise algorithms.
- RDF Semantic Graph support for schema-private semantic networks.
- Support for XML Schema, Text and Spatial Data Types to add, drop, and alter data type indexes and to enable the filtering of semantic queries written in SPARQL or SQL using XML schema, text, and spatial attributes.
- RDF Semantic Graph document indexing Enhancements:
  - Batch indexing of documents.
  - Flexible framework for managing entity extraction engines and associated rules.

– Local partitioned indexing.

– Operator to calculate the relevance of found documents.

SHARDED DATABASE MODEL

Sharding, a form of horizontal database partitioning that allows for linear horizontal scaling, has become increasingly popular for many customer-facing web applications, such as e-commerce, mobile, and social media. Such applications have a well-defined data model and data distribution strategy (hash, range, list, or composite) and primarily access data using a sharding key. Examples of sharding keys include customer ID, account number, and country_id.

Oracle Database 19c supports sharding. In this architecture stand-alone databases are used as individual shards in the data model. OLTP transactions that access data associated with a single value of the sharding key are the primary use-case for a sharded database. Examples of this are lookup and update of a customer’s records, subscriber documents, financial transactions, e-commerce transactions, and the like. Because all of the rows that have the same value of the sharding key are guaranteed to be on the same shard, such transactions are always single-shard and executed with the highest performance and provide the highest level of consistency. Multi-shard operations are supported, but with a reduced level of performance and consistency. Such transactions include simple aggregations, reporting, and the like, and play a minor role in a sharded application relative to workloads dominated by single-shard OLTP transactions.

Oracle Sharding sharded databases are useful for Cloud and other applications that benefit from linear scalability, fault containment, and geographical distribution of data. It can simplify rolling upgrades because applying configuration changes on one shard at a time does not affect other shards and allows administrators to first test the changes on a small subset of data. Sharding is well suited to deployment in the cloud. Shards may be sized as required to accommodate whatever cloud infrastructure is available and still achieve required service levels. Oracle Sharding supports on-premises, cloud, and hybrid deployment models.

ORACLE XML DB

XML has been widely adopted in just about every industry. XML based standards can be found in the Healthcare, Manufacturing, Financial Services, Government and Publishing sectors. The introduction of XML-based standards, such as XBRL, has led to XML becoming the de-facto mechanism for exchanging information among application systems. This has led to a growth in the use of XML as a persistence model for mission critical data.

To meet this need, Oracle developed Oracle XML DB. This is a high-performance, native XML storage and retrieval technology that is delivered with all versions of Oracle Database. It provides full support for all of the key XML standards, including XML, Namespaces, DOM, XQuery, SQL/XML and XSLT. Oracle XML DB is the first platform to deliver true hybrid relational / XML capabilities, making it possible to bring the full power of the SQL language to bear on XML content and the full power of the XML paradigm to relational data. Oracle XML DB includes the XML Developer's Kit (XDK), a versatile set of components that enables you to build and deploy C, C++, and Java software programs that process XML. You can assemble these components into an XML application that serves your business needs.

Oracle Database 19c extends its industry leading XML support ensuring that Oracle Database remains the best platform for storing, managing and querying all possible types of XML content. Features in Oracle Database 19c offer improved performance and scalability and enable complete support for the flexibility that makes the XML data model so attractive to so many different organizations.

Oracle Database 19c features for XML Developers include these XQuery capabilities:

- Support for XQuery Update, allowing users to efficiently update large XML Documents by performing fragment and node-level modifications using the W3C Query language.
- Support for XQuery Full-Text Specification, allowing document centric applications to take full advantage of full text searching and indexing.
- Support for XQuery API for Java (XQJ) as an API which is the Java Specification Request (JSR) for executing XQuery statements from Java programs.

Oracle Database 19c also includes core Oracle XML DB features:

- Over 10x faster query and index maintenance.
- Extended Partitioning Support for Binary XML Storage and Indexing
- Oracle XML DB and domain index support of hash and interval partitioned tables.
- Repository supports digest authentication, provides more robust security for users using HTTP to access content stored in the database.
- Repository allows WebDAV, HTTP, and FTP to be used to access content stored in DBFS.
- Oracle XDK supports W3C DOM Level 3 Core API's and reduces the memory footprint associated with using XML schemas.
- Integrated Oracle XQuery Implementation to unify the Oracle and BEA XQuery engines creating a single Java- based XQuery engine.
- Oracle XSLT/XPath engine interoperability to enable the use of non-XDK-based data models with the Oracle XDK/J XSLT/XPath engine, which supports interoperability between these Oracle engines and third-party XML processors.
- A Standalone XQuery Virtual Machine, allowing High performance XQuery operations to be performed on XML content stored outside the Oracle Database.

With Oracle Database 19c developers can take advantage of the features and power of the XSL 2.0 specification when developing XML-based applications. They are also able to make use of the feature set of XQuery 3.0.

ORACLE TEXT

Oracle Text is the leading text searching, retrieval and management system to be integrated into a database environment. With Oracle Database 19c, Oracle Text includes many features that improve index and query performance and improve usability, including:

- Near real-time indexing to support applications with frequently updated indexes.
- Improved performance for highlighting and snippet generation and native support of snippet information from the result set interface.
- Query Filter Cache feature allows you to cache the results of a particular query, or part of a query, and use those results to filter future searches for better performance.
- The number of MDATA and field sections allowed is now almost unlimited.

- Indexes can be modified without having to rebuild the index from scratch.

Oracle Database 19c also includes partition-specific near real-time indexes with improved management, read-only, auto-list and multi-list partitioning support, fine-grained cursor invalidation, the ability to add document formats and a range of other improvements.

ORACLE SECUREFILES

SecureFiles is designed to handle file data in Oracle Database, and delivers file system-like performance for basic query and insert operations. The optimized algorithms in SecureFiles make it up to 10x faster than previous LOB support (now called BasicFiles). SecureFiles can take advantage of several advanced Oracle Database capabilities that are not possible with file systems:

- In an Oracle Real Application Clusters environment, SecureFiles offers high levels of scalability that go far beyond what is offered in file systems
- SecureFiles allows for easy migration from older LOBs using Online Table Redefinition without affecting existing applications
- Applications no longer have to deal with multiple interfaces for manipulating relational and associated file data
- With SecureFiles, all information can be part of a database transaction, freeing the application from the complexity of guaranteeing atomicity, read consistency and other backup and recovery requirements
- SecureFiles also extends Transparent Data Encryption (TDE) capability to LOB data. The Oracle database supports automatic key management for all LOB columns within a table and transparently encrypts/decrypts data, backups and redo/undo log files.

Storage Optimization in SecureFiles

Also available with SecureFiles are advanced file system features such as Deduplication and Compression. Deduplication eliminates multiple, redundant copies of SecureFiles data and is completely transparent to applications. Oracle Database automatically detects multiple, identical SecureFiles data and stores only one copy, thereby saving storage space. Deduplication simplifies storage management resulting in significantly better performance, especially for copy operations.

SecureFiles data can be compressed using industry standard compression algorithms resulting in significant savings in storage and improved performance. Oracle Database automatically determines if the SecureFiles file is compressible or if compression savings are beneficial. SecureFiles uses a server-wide default LOB compression algorithm and provides for varying levels of compression. Each compression level represents a tradeoff between compression factor and speed. Organizations can choose the compression level which best suits their needs based on storage and CPU usage constraints. SecureFiles files are compressed and uncompressed automatically, transparent to applications.

Both Deduplication and Compression are part of the Advanced Compression capability of Oracle Database 19c.

SecureFiles Features in Oracle Database 19c

- Parallel DML Support for SecureFiles LOBs provides improved performance.
- SecureFiles LOB is the default storage option for LOBs, instead of BasicFiles LOB.

- The maximum size of SQL data types VARCHAR2, NVARCHAR2, and RAW change respectively from 4,000 bytes and 2,000 bytes to 32,767 bytes each. The corresponding PL/SQL data types remain at 32,767 bytes.
- Data Pump uses SecureFiles as default LOB storage. When you import tables, you can recreate all LOB columns as SecureFiles LOBs, thus converting BasicFiles LOBs to SecureFiles LOBs as part of data pump imports.
- SecureFiles supports components that enable HTTP, WebDAV, and FTP access to DBFS over the Internet, using various XML DB server protocols

CONCLUSION

In combination with Advanced Analytics and Data Warehouse features of Oracle Database, the Multimodel Database capabilities enable Oracle Database 19c to deliver scalable, high performance management and analysis capabilities for many Big Data and other application workflows in cloud and on-premises deployments. Traditional business applications – finance, order processing, manufacturing, and customer relationship management systems that easily conform to standard data structures (such as rows and columns with well-defined schemas), as well as applications based on analysis of web, social, spatial, mobile and sensor data, now increasingly incorporate Big Data for cloud applications. With Oracle Database 19c we have focused on offering data models like Property Graph and Sharded databases as well as delivering dramatically faster performance, moving more of the application logic and analytics into the database with cloud-ready JSON and REST services to simplify application development and enable analysis on dramatically larger datasets.

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries
TELE + 1.650.506.7000 + 1.800.ORACLE1
FAX + 1.650.506.7200
oracle.com

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

Integrated Cloud Applications & Platform Services

White Paper **Multimodel Database**
May 2019