Oracle® Database

Database New Features Guide

20c

F33423-01

July 2020

ORACLE®

Oracle Database Database New Features Guide, 20c

F33423-01

Contents

1   Oracle Database Release 20c New Features

ORACLE

Preface

This document describes new features implemented in Oracle Database 20c.

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

Read Oracle Database New Features Guide if you want to learn about features, options, and enhancements that are new in Oracle Database 20c.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Database 20c documentation set:

- *Oracle Database Error Messages*
- *Oracle Database Administrator's Guide*
- *Oracle Database Concepts*
- *Oracle Database Reference*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Oracle Database Release 20c New Features

This chapter contains descriptions of all of the features that are new to Oracle Database Release 20c.

- Big Data and Data Warehousing Solutions
- Security Solutions
- Performance and High-Availability Options
- Oracle Sharding
- Tools and Languages
- Database Upgrade and Utilities

Big Data and Data Warehousing Solutions

- Query Optimization
- Oracle Spatial and Graph
- Oracle Text

Query Optimization

- In-Memory Deep Vectorization

In-Memory Deep Vectorization

Oracle Database optimizes hash joins by pushing operations down into the scans of both sides of the join.

In deep vectorization, the left scan performs the build, while the right scan conducts the probe and gathers and projects rows back to the parent of the hash join operator. Joins can account for a large percentage of SQL execution time for data warehouse workloads. Improving performance of hash joins by 100% or more can significantly improve performance.

Related Topics

- *Oracle® Database In-Memory Guide*

Oracle Spatial and Graph

- Self-service Low-code Spatial Studio
- Spatial Support for Database In-Memory
- Spatial Queries on Data Stored in Oracle Object Storage

- Spatial Network Data Model Contraction Hierarchy
- Property Graph: Graph Server and Client Kit
- Property Graph: New Features in PGQL
- Property Graph Visualization
- Property Graph: User-defined Graph Algorithms
- Property Graph: Optimized Graph Representation for Faster Performance
- RDF Graph: Native Unicode Storage and Processing

Self-service Low-code Spatial Studio

Oracle Spatial Studio is a self-service web application that makes it easy for you to create interactive maps and perform spatial analysis on your business data. You can also use Spatial Studio to publish spatial analyses as REST services and generate SQL statements for spatial analysis using low-code and UI components.

With Spatial Studio, you no longer need to write Javascript or SQL or use third-party tools to take advantage of the spatial capabilities in Oracle Database.

Related Topics

- *Oracle® Spatial and Graph Developer's Guide*

Spatial Support for Database In-Memory

You can now perform spatial filter operations (`SDO_FILTER`) on spatial tables stored in Database In-Memory.

This means that you no longer have to create and maintain spatial indexes when spatial tables are stored using Database In-Memory for faster query performance.

Related Topics

- *Oracle® Spatial and Graph Developer's Guide*

Spatial Queries on Data Stored in Oracle Object Storage

You can run a wide range of spatial analytics against spatial data, in formats including GeoJSON and Shapefile, stored in the Oracle Cloud Infrastructure Object Storage service. Standard spatial SQL queries, and all Oracle Spatial and Graph vector analytics functions and operators, are supported.

You no longer need to load your data into Oracle Database before performing spatial analysis on ad hoc data sets. You can keep these data sets in place and run spatial queries directly against them.

Related Topics

- *Oracle® Spatial and Graph Developer's Guide*

Spatial Network Data Model Contraction Hierarchy

Contraction hierarchy, a precomputed in-memory approach, speeds up path computations in the Oracle Spatial and Graph network data model.

By using the contraction hierarchy Java API, you can evaluate shortest path computations, drive time polygon analysis, and traveling salesperson analysis functions more efficiently. These functions will perform 10 to 100 times faster than with previous releases. You can run more network analysis functions using fewer CPUs, and support more concurrent requests using the same hardware.

Related Topics

- *Oracle® Spatial and Graph Topology Data Model and Network Data Model Graph Developer's Guide*

Property Graph: Graph Server and Client Kit

You no longer have to copy the Property Graph in-memory analytics server (PGX) and client tools and libraries from `$ORACLE_HOME`.

With the simplified packaging and the availability of the Oracle Graph Server and Client kit on OCI Marketplace, application developers can quickly and securely install and deploy the components required to work with Property Graphs. This makes it easier to start developing applications.

Related Topics

- *Oracle® Spatial and Graph Property Graph Developer's Guide*

Property Graph: New Features in PGQL

You can now do graph DDL and graph DML operations with the graph query language PGQL. Additionally, you can use PGQL for CHEAPEST path queries using cost functions.

This simplifies development by eliminating the need for configuration files and Java APIs when doing several common graph operations.

Related Topics

- *Oracle® Spatial and Graph Property Graph Developer's Guide*

Property Graph Visualization

A rich set of visualization features lets you interactively explore the graph, customize layouts, and highlight interesting relationships in your data.

Seeing graph data and relationships visually lets analysts, data scientists, and developers quickly understand and explore clusters, outliers, anomalies, patterns, communities, and critical connections in their data. This makes you more productive and helps you share and communicate results more clearly.

Related Topics

- *Oracle® Spatial and Graph Property Graph Developer's Guide*

Property Graph: User-defined Graph Algorithms

You can create or extend graph algorithms using Java syntax, in addition to the dozens of pre-built graph analytics that come with the product. These user-defined algorithms will execute as fast as native algorithms in the product because they are compiled with the same optimizations.

For unique and specialized use cases, customizing graph algorithms lets you add analysis that analysts and data scientists design specifically for your applications.

Related Topics

- *Oracle® Spatial and Graph Property Graph Developer's Guide*

Property Graph: Optimized Graph Representation for Faster Performance

The In-memory analytics server (PGX, or Property Graph AnalytiX) has an optimized representation of a property graph that uses less memory. Larger graphs can be analyzed in the same amount of memory.

Graph sizes are continuously growing larger. With this optimization you can analyze larger graphs using less memory than previously required. This not only enables analysis of more data, but also reduces system costs. The optimized graph representation gives you faster performance and is transparent; existing applications will run faster with no change.

Related Topics

- *Oracle® Spatial and Graph Property Graph Developer's Guide*

RDF Graph: Native Unicode Storage and Processing

When creating the RDF (Resource Description Framework) network, you can now store RDF data in a native unicode format for virtually all use cases.

This reduces the storage required and enhances query performance.

Related Topics

- *Oracle® Spatial and Graph RDF Semantic Graph Developer's Guide*

Oracle Text

- New NETWORK_DATASTORE Data Store Type for Oracle Text
- New DIRECTORY_DATASTORE Data Store Type for Oracle Text
- Facet Navigation Support for JSON Search Indexes
- JSON Support in Result Set Interface
- Improved Index Synchronization and Automatic Index Optimization
- Custom Range Bucketing in Result Set Interface
- New Oracle Text Index Type: Search Index
- Named Entity Recognition Improvements
- In-Memory Full Text Columns

New NETWORK_DATASTORE Data Store Type for Oracle Text

Starting with Oracle Database 20c, you can use a new data store type called `NETWORK_DATASTORE` instead of the deprecated `URL_DATASTORE` data type.

`NETWORK_DATASTORE` provides greater security because it enables URL access based on access control lists (ACLs), which support HTTP and HTTPS protocols.

Related Topics

• *Oracle® Text Application Developer's Guide*

New DIRECTORY_DATASTORE Data Store Type for Oracle Text

Starting with Oracle Database 20c, you can use a new data store type called `DIRECTORY_DATASTORE` instead of the deprecated `FILE_DATASTORE` type.

`DIRECTORY_DATASTORE` provides greater security because it enables file access to be based on directory objects.

Related Topics

• *Oracle® Text Application Developer's Guide*

Facet Navigation Support for JSON Search Indexes

JSON search indexing now supports facet navigation using a JSON result set.

Related Topics

• *Oracle® Text Application Developer's Guide*

JSON Support in Result Set Interface

The JSON Result Set Interface (RSI) enables you to perform queries in JSON and return results as JSON.

The RSI enables you to fetch a set of results (a "hitlist") together with summary data such as the total number of hits and facet navigation information. This feature provides easier integration with modern programming languages which support JSON.

Related Topics

• *Oracle® Text Application Developer's Guide*

Improved Index Synchronization and Automatic Index Optimization

Index synchronization performance is considerably improved, particularly when lots of sessions are doing DML operations simultaneously with the sync-on-commit option. Optimize index now optimizes all index entries (including `SDATA` values) rather than just the word-based index entries.

Related Topics

• *Oracle® Text Application Developer's Guide*

Custom Range Bucketing in Result Set Interface

You can now specify range buckets for faceted navigation. For example, you can group all items costing between $10 and $20.

Range buckets increase the usability of faceted navigation and avoid the need to manually create range-value sections in the indexed table.

Related Topics

• *Oracle® Text Application Developer's Guide*

New Oracle Text Index Type: Search Index

The search indexes for text columns support support sharding and all partition types. Index creation syntax is also simpler now.

The search index type maintains most of the features of the traditional `CONTEXT` index type, but has support for sharding and for multiple partitioning methods (the previous `CONTEXT` index type can only support range partitioning). It also has a simplified index creation syntax avoiding the semantic complexity necessary for `CONTEXT` indexes.

Related Topics

• *Oracle® Text Application Developer's Guide*

Named Entity Recognition Improvements

Some restrictions have been removed from Named Entity Recognition (previously called Entity Extraction). Entity definitions can now be nested.

You can now incorporate existing or user-defined rules into other rules. For example, you can define rule "percentage change" as one of several words (for example, 'climbed', 'dropped', 'rose', 'increased') followed by the built-in rule for percentage.

Related Topics

• *Oracle® Text Application Developer's Guide*

In-Memory Full Text Columns

Oracle Text technology supports an `INMEMORY TEXT` directive during table creation, enabling fast in-memory searching using a `CONTAINS` clause.

An in-memory table scan can now evaluate non-scalar data. When the In-Memory Column Store contains both scalar and non-scalar columns, On-line Transaction Processing (OLTP) applications that access both types of data can avoid accessing the row store, thereby improving performance.

Related Topics

• *Oracle® Database In-Memory Guide*

Security Solutions

• Security

• Oracle Advanced Security

• Oracle Database Vault

Security

• Force Upgraded Password File to be Case Sensitive

• Connect to Multiple Databases with Different Certificates from a single client

• Ability to Specify the Location of the CMU Wallet and dsi.ora File with a Database Property

- Addition of USER_APPLICATION_ROLES Data Dictionary View
- Predefined Unified Audit Policies for Security Technical Implementation Guides (STIG) Compliance
- SYSLOG Destination for Common Unified Audit Policies
- Unified Audit Policies Enforced on the Current User
- Unified Audit Policy Configuration Changes Effective Immediately
- Auditing for Oracle XML DB HTTP and FTP Services
- Oracle Blockchain Table
- Enterprise User Manager Support for Per-PDB Directory Service Connections

Force Upgraded Password File to be Case Sensitive

Starting in Oracle Database 20c, the parameter to enable or disable password file case sensitivity is removed. All passwords in new password files are case-sensitive.

Case-sensitive password files provide more security than older password files that are case insensitive. Oracle recommends that you use case-sensitive password files. However, upgraded password files from earlier Oracle Database releases can retain their original case-insensitivity. You can force your password files to be case-sensitive by migrating password files from one format to another.

Related Topics

- *Oracle® Database Security Guide*

Connect to Multiple Databases with Different Certificates from a single client

Starting with Oracle Database 20c, you can configure database clients to maintain multiple Secure Sockets Layer (SSL) sessions using different SSL certificates.

This feature enables multi-threaded clients to use multiple wallets with different certificates for simultaneous SSL sessions.

This enhancement is especially useful for database clients simultaneously connecting to multiple cloud databases, each with a different certificate.

Related Topics

- *Oracle® Database Security Guide*

Ability to Specify the Location of the CMU Wallet and dsi.ora File with a Database Property

You now can specify a location of centrally managed users (CMU) wallet and `dsi.ora` files for an individual PDB by using a database property on the PDB.

This enhancement enables a PDB administrator to specify a location to store these files rather than being limited by the sqlnet.ora `WALLET_LOCATION` parameter or having to use the default wallet locations. This feature works almost exactly as using `WALLET_LOCATION` or the default wallet location except that users with administrative privileges will not be able to start the database because directory objects are part of the database. To store the CMU wallet and dsi.ora files in the location path specified

by a database directory object, you must set the `CMU_WALLET` database property to this directory object.

After you have set the `CMU_WALLET` database property to a directory object for an individual PDB, you should store the CMU wallet and `dsi.ora` files for this PDB in the location specified by the directory object, so that they can be accessed through the corresponding file systems.

Related Topics

- *Oracle® Database Security Guide*

Addition of USER_APPLICATION_ROLES Data Dictionary View

Starting with this release, the `USER_APPLICATION_ROLES` data dictionary view provides a subset of roles that are available in `DBA_APPLICATION_ROLES` that only apply to the current user.

`USER_APPLICATION_ROLES` enables the current user to see all the application roles that have been granted to the user instead of the list of all possible application roles available in `DBA_APPLICATION_ROLES`.

Related Topics

- *Oracle® Database Reference*

Predefined Unified Audit Policies for Security Technical Implementation Guides (STIG) Compliance

Starting with this release, you can audit for Security Technical Implementation Guide (STIG) compliance by using new predefined unified audit policies.

These policies are as follows:

- `ORA_STIG_RECOMMENDATIONS`
- `ORA_ALL_TOPLEVEL_ACTIONS`
- `ORA_LOGON_LOGOFF`

Related Topics

- *Oracle® Database Security Guide*

SYSLOG Destination for Common Unified Audit Policies

Certain predefined columns of unified audit records from common unified audit policies can be written to the UNIX `SYSLOG` destination.

To enable this feature, you set `UNIFIED_AUDIT_COMMON_SYSTEMLOG`, a new CDB level init.ora parameter. This enhancement enables all audit records from common unified audit policies to be consolidated into a single destination.

This feature is available only on UNIX platforms, not Windows.

Related Topics

- *Oracle® Database Security Guide*

Unified Audit Policies Enforced on the Current User

Starting with this release, unified audit policies are enforced on the current user who executes the SQL statement.

In previous releases, unified audit policies were enforced on the user who owned the top-level user session (that is, the login user session) in which the SQL statement is executed.

Scenarios in which the current user is different from the login user include but are not limited to the following:

- Trigger execution

- Definer rights procedure execution

- Functions and procedures that are executed during the evaluation of views

Related Topics

- *Oracle® Database Security Guide*

Unified Audit Policy Configuration Changes Effective Immediately

Starting with this release, changes made to a unified audit policy become effective immediately in the current session and in all other on-going active sessions.

In previous releases, users who were affected by a changed unified audit policy had to log out of and then back into the session in order for the unified audit policy to take effect.

Related Topics

- *Oracle® Database Security Guide*

Auditing for Oracle XML DB HTTP and FTP Services

Starting with this release, you can create unified audit policies for database connections made using the database protocol servers for HTTP, HTTPS, and FTP.

A unified audit policy can track requests that have been made to servlets, such as those used by Oracle Enterprise Manager Express, Oracle Database Native Web Services, and HTTP and WebDAV operations that use Oracle XML DB Repository.

This enhancement enables you to track and monitor access to Oracle Database provided by the HTTP, HTTPS, and FTP protocols, including access by WebDAV clients.

Related Topics

- *Oracle® Database Security Guide*

Oracle Blockchain Table

Blockchain tables are append-only tables in which only insert operations are allowed. Deleting rows is either prohibited or restricted based on time. Rows in a blockchain table are made tamper-resistant by special sequencing & chaining algorithms. Users can verify that rows have not been tampered. A hash value that is part of the row metadata is used to chain and validate rows.

Blockchain tables enable you to implement a centralized ledger model where all participants in the blockchain network have access to the same tamper-resistant ledger.

A centralized ledger model reduces administrative overheads of setting a up a decentralized ledger network, leads to a relatively lower latency compared to decentralized ledgers, enhances developer productivity, reduces the time to market, and leads to significant savings for the organization. Database users can continue to use the same tools and practices that they would use for other database application development.

Related Topics

- *Oracle® Database Administrator's Guide*

Enterprise User Manager Support for Per-PDB Directory Service Connections

Enterprise User Security (EUS) is now able to connect a different directory service for each pluggable database (PDB). Previously in a multitenant database, all the containers connected to a single directory service.

This feature enables a large Oracle Database deployment to have different directory services. It also enables independent software vendors to allocate an individual PDB to connect to the appropriate customer's directory service.

Related Topics

- *Oracle® Database Enterprise User Security Administrator's Guide*

Oracle Advanced Security

- Ability to Set the Default Tablespace Encryption Algorithm
- Improved Performance for Wallets with Large Numbers of TDE Keys in Wallets and Oracle Key Vault
- Enhanced Database Availability with Zero Downtime to Switch Over to an Updated PKCS#11 Library

Ability to Set the Default Tablespace Encryption Algorithm

You now can set the `TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM` dynamic parameter to define the default encryption algorithm for tablespace creation operations.

For example, if you set `TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM` to `AES256`, then future tablespace creation operations will use `AES256` as the default encryption algorithm. `TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM` applies to both offline and online tablespace encryption operations. In addition, when you create a new tablespace using Database Configuration Assistant (DBCA), you can set the default tablespace encryption algorithm by using the DBCA command line for silent installations.

Supported encryption algorithms are `AES128`, `AES192`, `AES256`, and `3DES168` If you do not set `TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM`, then the default encryption algorithm is the default that was used in previous releases: `AES128`.

Related Topics

- *Oracle® Database Advanced Security Guide*

Improved Performance for Wallets with Large Numbers of TDE Keys in Wallets and Oracle Key Vault

Oracle Database 20c introduces improved performance for Transparent Data Encryption (TDE).

This enhancement enables faster wallet loading and key rotations in multitenant databases. It allows for faster execution of TDE administration tasks and PDB cloning operations.

Enhanced Database Availability with Zero Downtime to Switch Over to an Updated PKCS#11 Library

Starting with this release, Oracle Database can switch over to an updated PKCS#11 library without incurring any system downtime.

This release introduces a new ADMINISTER KEY MANAGEMENT SWITCHOVER LIBRARY FOR ALL CONTAINERS statement, which will enable an Oracle database to switch over from the PKCS#11 library that it is currently using to the updated PKCS#11 library.

In previous releases, it was necessary to completely shut down any TDE-enabled database that used an online TDE master encryption key in Oracle Key Vault before an update to the Oracle Key Vault endpoint software could be installed. After the updated PKCS#11 library was installed, the TDE-enabled database would need to be started up again. This complete shut down followed by a start up of the database instance was necessary because long-running background processes of the database instance could not be told to unload the earlier PKCS#11 library and load the updated one.

Starting with this release, to switch over the database server to use an updated endpoint shared PKCS#11 library, you execute the ADMINISTER KEY MANAGEMENT SWITCHOVER TO LIBRARY 'fully_qualified_file_name_of_library' FOR ALL CONTAINERS; statement to initiate the switch over operation. Executing this command also updates the PKCS11_LIBRARY_LOCATION parameter to point to the location of the new PKCS#11 library.

Related Topics

- *Oracle® Database Advanced Security Guide*

Oracle Database Vault

- Ability to Prevent Local Oracle Database Vault Policies from Blocking Common Operations

Ability to Prevent Local Oracle Database Vault Policies from Blocking Common Operations

Starting with this release, a `DV_OWNER` common user in the CDB root can prevent local users from creating Oracle Database Vault controls on common objects in a pluggable database (PDB).

Blocking common users from common operations can prevent the execution of SQL commands that are necessary for managing the application or CDB database. To prevent this situation, a user who has the `DV_OWNER` role in the root can execute the `DBMS_MACADM.ALLOW_COMMON_OPERATION` procedure to control whether local PDB users can create Database Vault controls on common users' objects (database or application).

In previous releases, in a multitenant environment, a local Oracle Database Vault user could create Database Vault policies that could potentially block application or common operations. Blocking common users from common operations can prevent the execution of SQL commands that are necessary for managing the application or CDB database. To prevent this situation, a user who has the `DV_OWNER` role in the root can execute the `DBMS_MACADM.ALLOW_COMMON_OPERATION` procedure to control whether local PDB users can create Database Vault controls on common users' objects (database or application).

Related Topics

• *Oracle® Database Vault Administrator's Guide*

Performance and High-Availability Options

• Advanced Queuing

• Application Continuity

• Automatic Operations

• Oracle Database In-Memory

• Flashback

• Autonomous Health Framework

• Oracle Multitenant

Advanced Queuing

• Advanced Queuing: PL/SQL Enqueue and Dequeue Support for JMS Payload in Transactional Event Queues

• Advanced Queuing: PL/SQL Enqueue and Dequeue Support for non-JMS Payload in Transactional Event Queues

• Advanced Queuing: Transactional Event Queues for Performance and Scalability

• Advanced Queuing: Simplified Metadata and Schema in Transactional Event Queues

• Advanced Queuing: Kafka Java Client for Transactional Event Queues

• Advanced Queuing Support for JSON Data Type

Advanced Queuing: PL/SQL Enqueue and Dequeue Support for JMS Payload in Transactional Event Queues

PL/SQL APIs perform enqueue and dequeue operations for Java Message Service (JMS) payload in Transactional Event Queues. Similarly, the PL/SQL Array APIs are exposed to Transactional Event Queues JMS users. Since JMS support of heterogeneous messages, dequeue gets one of the five JMS message types back, but cannot predict what is the type of the next message received. Therefore, it can run into application errors with PL/SQL complaining about type mismatch. Oracle suggests that the application always dequeue from Transactional Event Queues using the generic type `AQ$_JMS_MESSAGE`. PL/SQL administration is also supported.

There are important high end customers who make use of PL/SQL APIs to enqueue and dequeue JMS payload in unsharded queues because of ease-of-use and to avoid client-server round trip time.

Oracle Database 20c introduces Transactional Event Queues (TEQ), which are partitioned message queues that combine the best of messaging, streaming, direct messages, and publish/subscribe. TEQ operates at scale on the Oracle database. TEQ provides transactional event streaming, and runs in the database in a scale of 10s to 100s of billions of messages per day on 2-node to 8-node Oracle RAC databases, both on-premise and on the cloud. TEQ has Kafka client compatibility, which means, Kafka producer and consumer can use TEQ in the Oracle database instead of a Kafka broker.

Advanced Queuing: PL/SQL Enqueue and Dequeue Support for non-JMS Payload in Transactional Event Queues

PL/SQL APIs can now perform enqueue and dequeue operations for ADT and RAW payload in Transactional Event Queues. Similarly, the PL/SQL array APIs are exposed to Transactional Event Queue users.

There are important high end customers who make use of PL/SQL APIs to enqueue and dequeue ADT or RAW payload in AQ queues. ADT payloads are important as it is a way to have different queue payloads needed by the applications.

Oracle Database 20c introduces Transactional Event Queues (TEQ), which are partitioned message queues that combine the best of messaging, streaming, direct messages, and publish/subscribe. TEQ operates at scale on the Oracle database. TEQ provides transactional event streaming, and runs in the database in a scale of 10s to 100s of billions of messages per day on 2-node to 8-node Oracle RAC databases, both on-premise and on the cloud. TEQ has Kafka client compatibility, which means, Kafka producer and consumer can use TEQ in the Oracle database instead of a Kafka broker.

Advanced Queuing: Transactional Event Queues for Performance and Scalability

Oracle Transactional Event Queues have their Queue tables partitioned into multiple Event Streams, which are distributed across multiple RAC nodes for high throughput messaging and streaming of events.

Partitioned tables form part of the foundation to scale and increase performance of Transactional Event Queues, especially on Oracle RAC or Exadata.

Oracle Database 20c introduces Transactional Event Queues (TEQ), which are partitioned message queues that combine the best of messaging, streaming, direct messages, and publish/subscribe. TEQ operates at scale on the Oracle database. TEQ provides transactional event streaming, and runs in the database in a scale of 10s to 100s of billions of messages per day on 2-node to 8-node Oracle RAC databases, both on-premise and on the cloud. TEQ has Kafka client compatibility, which means, Kafka producer and consumer can use TEQ in the Oracle database instead of a Kafka broker.

Advanced Queuing: Simplified Metadata and Schema in Transactional Event Queues

Transactional Event Queues have fewer tables than AQ and implements multiple memory optimizations or higher throughput. Customers will see higher message throughput just by switching from AQ to Transactional Event Queues.

This feature provides improvement in performance, scalability, and manageability.

Oracle Database 20c introduces Transactional Event Queues (TEQ), which are partitioned message queues that combine the best of messaging, streaming, direct messages, and publish/subscribe. TEQ operates at scale on the Oracle database. TEQ provides transactional event streaming, and runs in the database in a scale of 10s to 100s of billions of messages per day on 2-node to 8-node Oracle RAC databases, both on-premise and on the cloud. TEQ has Kafka client compatibility, which means, Kafka producer and consumer can use TEQ in the Oracle database instead of a Kafka broker.

Advanced Queuing: Kafka Java Client for Transactional Event Queues

Kafka Java Client for Transactional Event Queues (TEQ) enables Kafka application compatibility with Oracle database. This provides easy migration of Kafka applications to TEQ.

Customers don't have to manage a separate Kafka infrastructure, and this feature simplifies the event-driven application architectures with an Oracle converged database that now includes events data.

Starting from Oracle Database 20c, Kafka Java APIs can connect to Oracle database server and use Transactional Event Queues (TEQ) as a messaging platform. Developers can migrate an existing Java application that uses Kafka to the Oracle database. A client side library allows Kafka applications to connect to Oracle database instead of Kafka cluster and use TEQ messaging platform transparently.

Advanced Queuing Support for JSON Data Type

Oracle Database Advanced Queuing now supports `JSON` data type.

Many client application and micro-services which use Advanced Queuing for messaging have better performance if they use `JSON` data type to handle JavaScript Object Notation (JSON) messages.

Related Topics

- *Oracle® Database PL/SQL Packages and Types Reference*

Application Continuity

- Transparent Application Continuity
- Reset Session State
- Transparent Application Continuity in the Oracle Cloud
- Planned Failover

Transparent Application Continuity

In this release, the following new features are provided for Transparent Application Continuity:

- Oracle Database clients use implicit request boundaries when connecting to the database using a service that has the attribute `FAILOVER_TYPE` set to `AUTO`.

- Planned Failover is introduced, which is failover that is forced by the Oracle database at points where the Oracle database decides that the session can be failed over and the session is unlikely to drain. This feature is also available for Application Continuity.

- There is also a new service attribute, `RESET_STATE`. The resetting of state is an important feature that clears the session state set by the application in a request at the end of request. This feature is also available for Application Continuity.

This improvement increases coverage of Transparent Application Continuity for applications that do not use an Oracle-supplied connection pool. Planned failover is used for shedding sessions during planned maintenance. It is also used for load rebalancing. Without `RESET_STATE`, application developers need to cancel their cursors, and clear any session state that has been set.

Related Topics

- *Oracle® Database Development Guide*

Reset Session State

The resetting of the session state is an important feature that clears the session state set by the application in a request at the end of request. Setting session state in a request leaves the session "dirty". Subsequent usages of that session can see that session state if it is not cleaned. Without `RESET_STATE`, application developers need to cancel their cursors, and clear session state that has been set. `RESET_STATE` is a service attribute available with Application Continuity (AC and TAC) 20c.

`RESET_STATE` is a powerful feature where the database cleans the session state for you at the end of each request. Having the database do this task allows developers to focus on functionality without being burdened with cleaning the session state between usages. As the application changes, the state is automatically cleaned for you. `RESET_STATE` is guaranteed cleaning, reduces development work, and future proofs your application as the application changes.

Related Topics

- *Oracle® Real Application Clusters Administration and Deployment Guide*

Performance and High-Availability Options

Transparent Application Continuity in the Oracle Cloud

The feature set introduced in Oracle 20c is strictly focused on enhancing Transparent Application Continuity, with the particular focus on enabling it by default in an Oracle database cloud environment. Improving runtime performance, planned failover (deterministic draining) and broader application coverage are all aimed at enabling Transparent Application Continuity by default in the Oracle Cloud.

The enhancements to Transparent Application Continuity for Oracle 20c focus on reducing overhead and resource consumption, in addition to broadening replay capabilities, so that fewer and fewer exceptions will prevent enabling of replay for database request n the Cloud. These enhancements will decrease the ongoing performance impact of Application Continuity and provide still better Continuous Availability for applications working against databases in the Oracle Cloud.

Related Topics

- *Oracle® Real Application Clusters Administration and Deployment Guide*

Planned Failover

Planned Failover is failover that is forced by the database at points where the database decides the session should be replayable and is unlikely to drain. Planned Failover is used automatically for shedding sessions during planned maintenance for batch and long-running operations that are unlikely to complete in the Drain window. Planned Failover is also used automatically for load rebalancing. Planned Failover is available with Application Continuity and Transparent Application Continuity.

Planned Failover prevents batch and long-running sessions receiving errors during maintenance windows. Planned Failover is a smart learning engine that fails these sessions when they are expected not to drain and failover will succeed.

Planned failover allows mixed workloads of OLTP and long-running, the OLTP drain, and long-running sessions are failed over. Planned failover shortens the draining time, as sessions are failed over proactively.

Related Topics

- *Oracle® Real Application Clusters Administration and Deployment Guide*

Automatic Operations

- SecureFiles Defragmentation
- Automatic Index Optimization
- Automatic Zone Maps

SecureFiles Defragmentation

SecureFiles defragmentation provides online defragmentation of allocated and freed space in SecureFiles segments, for all types of SecureFiles LOBs - compressed, deduplicated, encrypted. Defragmentation can be done automatically by a background process, and the segment advisor can estimate the fragmentation levels and how much space can be saved. Defragmentation can be done mostly in-place, with some temp segment space needed to hold intermediate results.

SecureFiles defragmentation provides a transparent way to defragment or shrink the space used by SecureFiles segments, helping to reclaim space and improve performance, without compromising concurrent access to SecureFiles data, and without a significant impact on performance.

Related Topics

- *Oracle® Database SecureFiles and Large Objects Developer's Guide*

Automatic Index Optimization

ADO Policies for Indexes extends existing Automatic Data Optimization (ADO) functionality to provide compression and optimization capability on indexes. Customers of Oracle Database are interested in leveraging compression tiering and storage tiering to satisfy their Information Lifecycle Management (ILM) requirements. Existing ADO functionality enables you to set policies that enforce compression tiering and storage tiering for data tables and partitions automatically, with minimal user intervention.

In a database, indexes can contribute to a significant amount of database space. Reducing the space requirement for indexes, without sacrificing performance, requires ILM actions similar to the existing Automatic Data Optimization feature for data segments. Using this new Index compression and optimization capability, the same ADO infrastructure can also automatically optimize indexes. Similar to ADO for data segments, this automatic index compression and optimization capability achieves ILM on indexes by enabling you to set policies that automatically optimize indexes through actions like compressing, shrinking and rebuilding indexes.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

Automatic Zone Maps

Automatic zone maps are created and maintained for any user table without any customer intervention. Zone maps allow the pruning of blocks and partitions based on the predicates in the queries, without any user intervention. Automatic zone maps are maintained for direct loads, and are maintained and refreshed for any other DML operation incrementally and periodically in the background.

Automatic zone maps are improving the performance of any query transparently and automatically without the need of any user action.

Related Topics

- *Oracle® Database Data Warehousing Guide*

Oracle Database In-Memory

- In-Memory Base Level
- Automatic In-Memory
- In-Memory Hybrid Scans
- Database In-Memory External Table Enhancements

In-Memory Base Level

Database In-Memory is an option to Enterprise Edition. Database In-Memory now has a new "Base Level" feature. This allows the use of Database In-Memory with up to a 16GB column store without triggering any license tracking.

The feature allows you to use Database In-Memory without having to license the option. The column store is limited to 16GB when using the Base Level. This helps to show the value of Database In-Memory without having to worry about licensing issues.

Related Topics

- *Oracle® Database In-Memory Guide*

Automatic In-Memory

Automatic In-Memory enables, populates, evicts, and recompresses segments without user intervention.

When `INMEMORY_AUTOMATIC_LEVEL` is set to `HIGH`, the database automatically enables and populates segments based on their usage patterns. Combined with support for selective column level eviction and recompression, In-Memory population is largely self-managing. This automation helps maximize the number of objects that can be populated into the In-Memory Column Store at one time.

Related Topics

- *Oracle® Database In-Memory Guide*

In-Memory Hybrid Scans

Oracle Database supports In-memory scans when not all columns in a table have been populated into the In-Memory Column Store (IM column store).

This situation can occur when columns have been specified as `NO INMEMORY` to save space. In-memory hybrid scans can access some data from the IM column store, and some data from the row store, improving performance by orders of magnitude over pure row store queries.

Related Topics

- *Oracle® Database In-Memory Guide*

Database In-Memory External Table Enhancements

For a partitioned or hybrid external table, the `INMEMORY` clause is supported at both the table and partition level. For hybrid tables, the table-level `INMEMORY` attribute applies to all partitions, whether internal or external.

This enhancement significantly broadens support for in-memory external tables.

Related Topics

- *Oracle® Database In-Memory Guide*

Flashback

- [PDB Point-in-Time Recovery or Flashback to Any Time in the Recent Past](#)

PDB Point-in-Time Recovery or Flashback to Any Time in the Recent Past

PDBs can be recovered to an orphan PDB incarnation within the same CDB incarnation or an ancestor incarnation.

Availability of PDBs is enhanced. Both flashback and point-in-time recovery operations are supported when recovering PDBs to orphan PDB incarnations.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

Autonomous Health Framework

- [Oracle ORAchk and Oracle EXAchk Support for Auto-Analysis](#)
- [Oracle ORAchk and Oracle EXAchk Support for REST API](#)
- [Oracle Trace File Analyzer Real-Time Health Summary](#)
- [Oracle Trace File Analyzer Log File Life Cycle Enhancements](#)
- [Oracle Trace File Analyzer Support for Efficient Multiple Service Request Data Collections](#)

Oracle ORAchk and Oracle EXAchk Support for Auto-Analysis

This release includes improvements to the accuracy and relevance of Oracle ORAchk and Oracle EXAchk alerts that enable you to respond more quickly.

Oracle ORAchk and Oracle EXAchk raise alerts if they detect a failed check. In earlier releases, Oracle ORAchk and Oracle EXAchk map the alerts statically as Info, Warning, and Critical. With this auto-analysis enhancement using applied machine learning, Oracle ORAchk and Oracle EXAchk adjust the levels and their thresholds dynamically to become more relevant and actionable.

It is critical that you maintain the operational availability and performance of the Oracle software stack. Oracle ORAchk and Oracle EXAchk monitor real-time configuration of both the Oracle software and host operating system and hardware.

Related Topics

- *Oracle® Autonomous Health Framework User's Guide*

Oracle ORAchk and Oracle EXAchk Support for REST API

In this release, support for REST API adds a remote interface to the existing Oracle ORAchk and Oracle EXAchk command-line interfaces (CLI).

You can manage Oracle software deployments remotely from centralized consoles and web interfaces. By supporting the REST interfaces, Oracle ORAchk and Oracle EXAchk integrate into these applications and help support fleet or cloud management.

Related Topics

• *Oracle® Autonomous Health Framework User's Guide*

Oracle Trace File Analyzer Real-Time Health Summary

Oracle Trace File Analyzer generates a real-time health summary report, which shows performance degradation due to faults and workload issues.

Similar to the status scorecard of the deployment configurations that Oracle ORAchk and Oracle EXAchk generate, Oracle Trace File Analyzer also provides a readily consumable and trackable scoring for operational status. The health summary consists of scores in the categories of availability, health, workload, and capacity broken down from cluster-wide through the database, instance, service, and hardware resource.

Related Topics

• *Oracle® Autonomous Health Framework User's Guide*

Oracle Trace File Analyzer Log File Life Cycle Enhancements

Oracle Trace File Analyzer archives log files before purging them upon each rotation.

The Oracle Database and Oracle Grid Infrastructure deployments generate a large number of logs and trace files. Oracle Trace File Analyzer does not archive these files. You have to create custom jobs if you need this history for support or auditing purposes. The enhancement in this release builds in the desired archiving functionality and thus removes the need for custom scripts.

Related Topics

• *Oracle® Autonomous Health Framework User's Guide*

Oracle Trace File Analyzer Support for Efficient Multiple Service Request Data Collections

Oracle Trace File Analyzer collects multiple Service Request Data Collections into a single collection even if it detects multiple issues or errors at the same time.

Service Request Data Collection mode of operation enables you to collect only the log and trace files that are required for diagnosing a specific type of problem. Even with this optimization, Oracle Trace File Analyzer collects the same subset of files if it detects multiple issues or errors at the same time. The enhancement further optimizes the collection of multiple Service Request Data Collections into a single collection and thus removes duplication.

It is essential to collect log and trace files upon detection of issues before the files are rotated or purged. However, collecting log and trace files involves resource overhead, which may be critically low due to these issues. The enhancement in this release reduces the resource overhead and disk space needed at a critical time.

Related Topics

• *Oracle® Autonomous Health Framework User's Guide*

Oracle Multitenant

• Namespace Integration with Database

- MAX_IDLE_BLOCKER_TIME Parameter
- Expanded Syntax for PDB Application Synchronization

Namespace Integration with Database

Database Nest is an infrastructure that provides operating system resource isolation and management, file system isolation, and secure computing for CDBs and PDBs. This infrastructure enables a database instance to run in a protected, virtualized environment.

Sharing instance-level and operating system resources can lead to security and isolation constraints, especially in large-scale cloud deployments. Vulnerabilities can be external, such as compromised applications, unauthorized access of resources, and shared resources. An example of an internal vulnerability is a compromised Oracle process.

Database Nest isolates a database instance from other databases and applications running on the same host, and also isolates PDBs from each other and from the CDB. The feature is implemented as a Linux-specific package that provides hierarchical containers, called nests. A CDB resides within a single parent nest, while PDBs reside within the individual child nests created within the parent.

Linux processes in a PDB nest have their own process ID (PID) number spaces and cannot access PIDs in other nests. Process isolation provides a last level of defense in a security breach if a malicious user compromises a process.

Related Topics

- *Oracle® Multitenant Administrator's Guide*

MAX_IDLE_BLOCKER_TIME Parameter

`MAX_IDLE_BLOCKER_TIME` sets the number of minutes that a session holding needed resources can be idle before it is a candidate for termination.

`MAX_IDLE_TIME` sets limits for all idle sessions, whereas `MAX_IDLE_BLOCKER_TIME` sets limits only for idle sessions consuming resources. `MAX_IDLE_TIME` can be problematic for a connection pool because it may continually try to re-create the sessions terminated by this parameter.

Related Topics

- *Oracle® Multitenant Administrator's Guide*

Expanded Syntax for PDB Application Synchronization

The `ALTER PLUGGABLE DATABASE APPLICATION ... SYNC` statement now accepts multiple application names and names to be excluded. For example, a single statement issued in an application PDB can synchronize `app1` and `app2`, or synchronize all applications except `app3`.

The expanded syntax enables you to reduce the number of synchronization statements. Also, the database replays the statements in correct order. Assume that you upgrade `ussales` from v1 to v2, and then upgrade `eusales` from v1 to v2, and then upgrade `ussales` from v2 to v3. The statement `ALTER PLUGGABLE DATABASE APPLICATION ussales, eusales SYNC` replays the statements in sequence, upgrading `ussales` to v2, then `eusales` to v2, and then `ussales` to v3.

Related Topics

- *Oracle® Multitenant Administrator's Guide*

Oracle Sharding

- Oracle Sharding

Oracle Sharding

- Multi-Shard Query, Data Loading, and DML Enhancements
- Sharding Advisor Schema Analysis Tool
- Create a Sharded Database from Multiple Existing Databases (Federated Sharding)

Multi-Shard Query, Data Loading, and DML Enhancements

If any shards are unavailable during query execution, then the enhanced multi-shard query attempts to find alternate shards to operate on, and the query resumes without issuing a failure condition. Bulk data loading and DML can operate on multiple shards simultaneously.

Multi-shard queries are more fault-tolerant. Bulk data loading and DML operations can occur across all shards simultaneously, making these operations much faster.

Related Topics

- *Oracle® Database Using Oracle Sharding*

Sharding Advisor Schema Analysis Tool

Sharding Advisor is a standalone command-line tool that helps you redesign a database schema so that you can efficiently migrate an existing, non-sharded Oracle Database to an Oracle sharding environment. Sharding Advisor analyzes your existing database schema and produces a ranked list of possible sharded database designs.

Using the Sharding Advisor recommendations, you can experience a smoother, faster migration to Oracle Sharding. Sharding Advisor analysis provides you with the information you need to

- Maximize availability and scalability
- Maximize query workload performance
- Minimize the amount of duplicated data on each shard

Related Topics

- *Oracle® Database Using Oracle Sharding*

Create a Sharded Database from Multiple Existing Databases (Federated Sharding)

Convert a set of existing databases running the same application into a sharded database, without modifying the database schemas or the application. The databases

can be geographically distributed and can have some differences in their individual schemas.

You can more easily issue queries across multiple independent databases running the same application when they are combined into a sharded database.

Related Topics

- *Oracle® Database Using Oracle Sharding*

Tools and Languages

- Analytical SQL and Statistical Functions
- Oracle Application Express
- Oracle Machine Learning for SQL
- Globalization
- Java in Oracle Database
- JDBC
- JSON Document Store
- NET Services
- Oracle Call Interface
- PL/SQL
- SQL

Analytical SQL and Statistical Functions

- Bitwise Aggregate Functions
- New Analytical and Statistical Aggregate Functions
- Enhanced Analytic Functions

Bitwise Aggregate Functions

New aggregate functions `BIT_AND_AGG`, `BIT_OR_AGG`, and `BIT_XOR_AGG` enable bitwise aggregation of integer columns and columns that can be converted or rounded to integer values.

Bitwise aggregation functions enable bitwise type processing directly in SQL. Use of these new functions improves overall query performance by eliminating unnecessary data movement and by taking full advantage of other database capabilities such as parallel processing.

Related Topics

- *Oracle® Database SQL Language Reference*

New Analytical and Statistical Aggregate Functions

New analytical and statistical aggregate functions are available in SQL:

`CHECKSUM` computes the checksum of the input values or expression.

KURTOSIS functions KURTOSIS_POP and KURTOSIS_SAMP measure the tailedness of a data set where a higher value means more of the variance within the data set is the result of infrequent extreme deviations as opposed to frequent modestly sized deviations. Note that a normal distribution has a kurtosis of zero.

SKEWNESS functions SKEWNESS_POP and SKEWNESS_SAMP are measures of asymmetry in data. A positive skewness is means the data skews to the right of the center point. A negative skewness means the data skews to the left.

All of these new aggregate functions support the keywords ALL, DISTINCT, and UNIQUE.

With these additional SQL aggregation functions, you can write more efficient code and benefit from faster in-database processing.

Related Topics

- *Oracle® Database SQL Language Reference*

Enhanced Analytic Functions

Window functions now support the EXCLUDE options of the SQL standard window frame clause. The query_block clause of a SELECT statement now supports the window_clause, which implements the window clause of the SQL standard table expression as defined in the SQL:2011 standard.

Supporting the full ANSI standard enables easier migration of applications that were developed against other standard-compliant database systems.

Related Topics

- *Oracle® Database Data Warehousing Guide*

Oracle Application Express

- Faceted Search Page
- JET Chart Enhancements
- Enhancements Shared LOVs and Popup LOVs
- Universal Theme Enhancements
- Dark Mode
- New Data Upload
- New Form Region
- REST Read / Write Enhancements

Faceted Search Page

Create easy filterable reports by creating a faceted search page.

The new faceted search page provides a powerful new way to filter and display data. A faceted search page features a faceted region that enables users to narrow down the search results and display the results in either a report or cards view.

Related Topics

- *Oracle® Application Express App Builder User's Guide*

JET Chart Enhancements

Oracle Application Express includes new declarative attributes for JET Charts to support the styling of chart text.

New declarative attributes enable you to declaratively style the chart title, legend text, and axes titles, and labels. Two new declarative attributes, Stack Category and Stack Label, have also been added to JET charts to support the rendering of a stacked chart.

Related Topics

* *Oracle® Application Express App Builder User's Guide*

Enhancements Shared LOVs and Popup LOVs

Oracle Application Express includes a number of productivity enhancements to Shared LOVs and Popup LOVs.

Shared List of Values (LOV) now include support for additional data sources, declarative column mappings, and multiple display column. Popup LOVs now support multiple display columns, search-as-you-type, improved multiple value selection, and a simplified user interface.

Related Topics

* *Oracle® Application Express App Builder User's Guide*

Universal Theme Enhancements

The Universal Theme includes UI refinements, accessibility improvements, new template options, and Theme Roller enhancements.

Universal Theme contains many accessibility improvements including a new Skip to Main Content link. The Universal Theme also includes two additional styles for the tree-based Navigation Menu and a new Inline Popup region template. Finally, a number of components and styles have been visually tweaked and refined to provide a more streamlined user interface.

Related Topics

* *Oracle® Application Express App Builder User's Guide*

Dark Mode

Decrease eye strain by enabling Dark Mode.

The development environment can now render with a darker color scheme, which reduces eye strain and is especially helpful for developing late into the night.

Related Topics

* *Oracle® Application Express App Builder User's Guide*

New Data Upload

The data upload functionality has been modernized to support native Excel, CSV, XML and JSON documents.

Data upload in SQL Workshop features a new drag and drop user interface that provides support for uploading native Excel, CSV, XML and JSON documents into a new table or into existing tables. These same capabilities can be accessed from the Create Application Wizard by selecting the From a File option.

Related Topics

- *Oracle® Application Express App Builder User's Guide*

New Form Region

A new Form region type provides superior functionality compared to the old legacy form pages.

A new Form region type provides support for internal and external data sources (that is, Local Database, REST Enabled SQL Service, or Web Source). Forms can be created on SQL queries, or by creating PL/SQL Functions returning SQL Queries. In addition, this new Form region type supports more than two Primary Key Columns and includes more control on Lost Update detection and Row Locking. Finally you can migrate existing legacy form pages to the new form region with the Upgrade Application function in App Builder Utilities.

Related Topics

- *Oracle® Application Express App Builder User's Guide*

REST Read / Write Enhancements

Read and write data using REST Enabled SQL references or Web Source Modules.

Define a REST Enabled SQL reference once and use it across a workspace. Create Web Source Modules to define the source within each application. Both approaches are fully declarative, using Data Discovery with parameters. Forms and interactive grids now include built-in support for the internal or external data source you specify.

Related Topics

- *Oracle® Application Express SQL Workshop Guide*

Oracle Machine Learning for SQL

- Oracle Machine Learning XGBoost Algorithm
- Oracle Machine Learning MSET-SPRT Algorithm
- Adam Optimization Solver for the Neural Network Algorithm

Oracle Machine Learning XGBoost Algorithm

XGBoost is a highly-efficient, scalable gradient tree boosting machine learning algorithm for regression and classification.

The `DBMS_DATA_MINING.ALGO_XBGOOST` algorithm prepares training data, builds and persists a model, and applies the model for prediction. You can use it as a stand-alone predictor or incorporate it into real-world production pipelines for a wide range of problems such as ad click-through rate prediction, hazard risk prediction, web text classification, and so on.

Related Topics

- *Oracle® Machine Learning for SQL Concepts*

Oracle Machine Learning MSET-SPRT Algorithm

The Multivariate State Estimation Technique-Sequential Probability Ratio Test (MSET-SPRT) algorithm is a nonlinear, nonparametric anomaly detection technique for monitoring critical processes.

The `DBMS_DATA_MINING.ALGO_MSET_SPRT` algorithm detects subtle anomalies while producing minimal false alarms. It calibrates expected behavior from available, historical data of the normal operational sequence of monitored signals. It incorporates the learned behavior of the system into a persistent MSET-SPRT model. You can apply the model to new records to detect anomalous behavior.

Related Topics

- *Oracle® Machine Learning for SQL Concepts*

Adam Optimization Solver for the Neural Network Algorithm

Adam is an optimization solver for the Neural Network algorithm that is computationally efficient, requires little memory, and is well suited for problems that are large in terms of data or parameters or both.

Adam is a popular extension to stochastic gradient descent. It uses mini-batch optimization and can make progress faster while seeing less data than the other Neural Network optimization solver, Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) with line search.

Related Topics

- *Oracle® Machine Learning for SQL Concepts*

Globalization

- Unicode 12.1 Support
- New German Linguistic Sorts for Capital Sharp S Support
- New Era Support for Japanese Imperial Calendar

Unicode 12.1 Support

The National Language Support (NLS) data files for AL32UTF8 and AL16UTF16 character sets are updated to match version 12.1 of the Unicode Standard character database. The support for Unicode Collation Algorithm (UCA) is also updated to conform with UCA 12.1.

This enhancement enables Oracle Database to conform to the latest version of the Unicode Standard.

Related Topics

- *Oracle® Database Globalization Support Guide*

New German Linguistic Sorts for Capital Sharp S Support

Two new linguistic sorts (`XGERMAN_S` and `XGERMAN_DIN_S`) are added to Oracle Database to support Latin Capital Letter Sharp S as the uppercase form of Latin Smaller Letter Sharp S.

This feature expands the database linguistic support to meet the requirements of the German market.

Related Topics

- *Oracle® Database Globalization Support Guide*

New Era Support for Japanese Imperial Calendar

The new Japanese era Reiwa, which went into effect on May 1, 2019, is now supported in Oracle Database for the Japanese Imperial Calendar.

The Japanese Imperial Calendar is used widely in Japan in addition to the Gregorian Calendar. Supporting the new era is an essential requirement for software distributed in the Japanese market.

Related Topics

- *Oracle® Database Globalization Support Guide*

Java in Oracle Database

- Oracle JVM Security Enhancements

Oracle JVM Security Enhancements

Oracle Java virtual machine (JVM) security has been enhanced to make it well suited for pluggable databases (PDBs). The enhancements also include support for the new Java module system and safeguarding of Oracle JVM against security vulnerabilities.

The enhanced Oracle JVM security makes it safe to use Oracle JVM with pluggable databases (PDB) and shields it from security vulnerabilities like the Spectre attack. In addition, it simplifies developing Java in the database applications using the new module system.

Related Topics

- *Oracle® Database Java Developer's Guide*

JDBC

- JDBC Datasource for Simplifying Sharded Databases Access
- JDBC Reactive Extensions

JDBC Datasource for Simplifying Sharded Databases Access

You can now use a new Java Database Connectivity (JDBC) datasource for accessing sharded databases without furnishing the sharding keys.

Java applications refer to `jdbc:oracle:sharding` in connection IURL

This feature simplifies sharded database access and unlimited scalability for existing Java applications.

Related Topics

- *Oracle® Database JDBC Developer's Guide*

JDBC Reactive Extensions

The Reactive Extensions are a set of methods that extend the JDBC standard to offer asynchronous database access with back pressure support. The Reactive Extensions implement the Publisher and Subscriber types defined by `java.util.concurrent.Flow`. Flow is the JDK's standard representation of a reactive stream.

The Reactive Extensions use non-blocking mechanisms for creating Connections, executing SQL, fetching rows, committing, rolling back, closing Connections, and reading and writing BFILEs, BLOBs, and CLOBs.

The extensions bring scalability, high throughput and Reactive Streams support to Java applications that use the Oracle Database in the Cloud and on-premises. This extension works as a Service Provider for the Reactive Streams libraries including: Reactor, RxJava and Akka Streams.

Related Topics

- *Oracle® Database JDBC Developer's Guide*

JSON Document Store

- New JSON Data Type
- New Oracle SQL Function JSON_TRANSFORM
- SQL/JSON Syntax Improvements
- JSON Scalar Allowed at Top Level of JSON Document (RFC 8259 Support)

New JSON Data Type

`JSON` is a new SQL and PL/SQL data type for JSON data. It provides a substantial increase in query and update performance.

JSON data type is optimized for query and DML processing. It can yield database performance improvements for processing JSON data.

You can use JSON data type and its instances in most places where a SQL data type is allowed, including:

- As the column type for table or view DDL
- With SQL/JSON functions and conditions, and with PL/SQL procedures and functions
- In Oracle dot-notation query syntax
- For creation of functional and search indexes

Oracle Call Interface and Java Database Connectivity (JDBC) clients now provide APIs that can work directly with binary JSON data, significantly saving network costs and server CPU cycles.

Related Topics

- *Oracle® Database JSON Developer's Guide*

New Oracle SQL Function JSON_TRANSFORM

You can use SQL function `JSON_TRANSFORM` to update parts of a JSON document. You specify which parts to modify, the modifications, and any new values.

`JSON_TRANSFORM` makes it easier for an application to modify a JSON document, without having to parse and rebuild it. In most cases, it also avoids a round-trip between the server and client for the whole document.

Related Topics

- *Oracle® Database JSON Developer's Guide*

SQL/JSON Syntax Improvements

You can now express more complex SQL/JSON queries and express some queries more succinctly.

- New SQL function `JSON_SCALAR` accepts a scalar instance of a SQL data type and returns a scalar JSON value as an instance of JSON data type.

- New JSON path-language item methods support `JSON_SCALAR: float()`, `double(), binary(), ymInterval(),` and `dsInterval()`.

- The JSON path-language and dot-notation syntax support new, aggregate item methods: `avg(), count(), minNumber(), maxNumber(), minString(), maxString(), sum()`.

- You can now express more complex SQL/JSON queries and express some queries more succinctly, and SQL/JSON path-expression syntax for array steps is improved.

Related Topics

- *Oracle® Database JSON Developer's Guide*

JSON Scalar Allowed at Top Level of JSON Document (RFC 8259 Support)

JSON documents in Oracle Database can now have a top-level JSON scalar value. Previously they had to have a JSON object or array value.

This feature helps Oracle JSON support be compliant with RFC 8259.

This behavior is not enabled by default. To enable it, set database initialization parameter `compatible` to 20. The default value for Oracle Database Release 20c is `12.2`.

If the parameter is at least `20,` then JSON data that is stored either textually (`VARCHAR2, CLOB, BLOB`) or as `JSON` data type respects RFC 8259: it allows top-level scalars in documents. For a JSON column you can, however, use an `IS JSON` check constraint with keywords `DISALLOW SCALARS` to disallow documents having top-level scalar values.

Related Topics

- *Oracle® Database JSON Developer's Guide*

NET Services

- [Oracle CMAN Traffic Director Mode Support for All Types of Database Links](#)
- [Migration of Oracle CMAN Sessions with High Availability](#)
- [REST APIs for Oracle CMAN Administration, Proxy Protocol, Enhanced Rule List, and Bandwidth Management](#)

Oracle CMAN Traffic Director Mode Support for All Types of Database Links

Starting with Oracle Database 20c, Oracle Connection Manager (CMAN) Traffic Director Mode is extended to support all types of dedicated database links including Fixed User, Connected User, and Current User.

Oracle Connection Manager in Traffic Director Mode enhances application scalability, performance, security, tenant isolation, and high availability (zero downtime during planned and unplanned database outages).

Related Topics

- *Oracle® Database Net Services Administrator's Guide*

Migration of Oracle CMAN Sessions with High Availability

Client/server sessions can be migrated from one Oracle Connection Manager (CMAN) instance to another Oracle CMAN instance during a planned upgrade or while patching Oracle CMAN. Live sessions can be migrated with data in-transit.

Operations that are running either on a client or on a server continue to run seamlessly during the migration with zero downtime. You can also add new client connections during the migration.

Related Topics

- *Oracle® Database Net Services Administrator's Guide*

REST APIs for Oracle CMAN Administration, Proxy Protocol, Enhanced Rule List, and Bandwidth Management

You can use REST APIs to manage Oracle Connection Manager (Oracle CMAN) instances. Proxy protocol provides additional security and access control. Enhanced rule list allows scalability with segregation of rule_lists for each service. You can also manage distribution of bandwidth across services using Oracle CMAN.

This feature offers REST API management, security with proxy protocol, scalability with enhanced rule list, and per service bandwidth management.

Related Topics

- *Oracle® Database Net Services Administrator's Guide*

Oracle Call Interface

- New C Client Interface APIs for JSON Data Type
- Support for C99 Syntax in Pro*C/C++ Precompiler
- Oracle Call Interface Session Pool Improvements
- Improvements to OCI Data Interface for LOBs
- Oracle Call Interface API to build and quote TNS Connection Strings

New C Client Interface APIs for JSON Data Type

This feature introduces new Oracle Call Interface (OCI) APIs to operate with JSON descriptor. You can read and write textual JSON or binary JSON (in Oracle binary format) from/to a buffer or a stream. You can bind and define a JSON descriptor to write and read JSON data from the database.

With the new JSON APIs, you can now preserve abstraction without a need to serialize the data to text..

Related Topics

- *Oracle® Call Interface Programmer's Guide*

Support for C99 Syntax in Pro*C/C++ Precompiler

The Pro*C/C++ Precompiler now supports the C99 standard, the ISO/IEC 9899:1999 standards specification for C programming.

C99 syntax and semantics allow application developers to use richer functionality.

Related Topics

- *Pro*C/C++ Programmer's Guide*

Oracle Call Interface Session Pool Improvements

The Oracle Call Interface Session Pool has been enhanced.

Monitoring and administration has been improved.

When the number of sessions exceeds the minimum pool size, idle session cleanup now occurs even if there is no pool activity.

There is better OCISessionGet wait timeout accuracy.

When using Oracle Sharding, the pool has better balancing of session across shards.

You can now manage Oracle Call Interface Session Pool better with performance, auto-tuning, and administration improvements.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

Improvements to OCI Data Interface for LOBs

This feature makes the OCI Data Interface for LOBs more user friendly.

The OCI Data Interface for LOBs is efficient because it reduces the number of round trips between the client and the database. This feature is further improved by removing the need for NULL callbacks and usage of 'ind' variable when the LOB length exceeds the sb2 size.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

Oracle Call Interface API to build and quote TNS Connection Strings

A new Oracle Call Interface API can now be used to build and quote Oracle Database Transparent Network Substrate (TNS) connection strings.

The Oracle Call Interface enhancement enables validation of values, which you can use for Oracle Net Service connection string attributes.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

PL/SQL

- New Pragma SUPPRESSES_WARNING_6009
- PL/SQL Type Attributes in User-Defined Types
- PL/SQL Aggregate Qualified Expressions Enhancements
- New PL/SQL Iterator Constructs
- PL/SQL Support For New JSON SQL Data Type

New Pragma SUPPRESSES_WARNING_6009

The `SUPPRESSES_WARNING_6009` pragma allows more robust error handling and better encapsulation and modularization.

The PL/SQL compiler issues warning `PLW-06009` if it determines that an `OTHERS` exception handler does not, in all cases, end in either an explicit `RAISE` statement or in a call to the PL/SQL supplied procedure `RAISE_APPLICATION_ERROR`. The compiler's behavior is too aggressive for some programming styles when programmers supply their own reporting subroutines. This new pragma quiets the warning.

Related Topics

- *Oracle® Database PL/SQL Language Reference*

PL/SQL Type Attributes in User-Defined Types

You can use attributes of PL/SQL scalar data types, such as `BOOLEAN` and `PLS_INTEGER`, in non-persistable object types.

You can use non-persistable object types in your PL/SQL code if you have no desire to persist instances of these types. This is useful when you are developing programs following Oracle's object oriented programming model.

Related Topics

- *Oracle® Database PL/SQL Language Reference*

PL/SQL Aggregate Qualified Expressions Enhancements

Starting with Oracle Database Release 20c, three new types of iterator choice association are added for use in qualified expressions. The basic iterator choice association extends the current iterator choice association by allowing a full iterator as the index. The index iterator choice association provides an index expression along with the value expression. The sequence iterator choice association allows a sequence of values to be added to the end of a collection. In each case, the expressions specified may reference the iterands.

Aggregates and their necessary adjunct, qualified expressions, improve program clarity and programmer productivity.

Related Topics

- *Oracle® Database PL/SQL Language Reference*

New PL/SQL Iterator Constructs

PL/SQL is enhanced to help you program iteration controls using new iterators in loops and in qualified expressions.

The new iterator constructs are clear, simple, understandable, and efficient.

Related Topics

- *Oracle® Database PL/SQL Language Reference*

PL/SQL Support For New JSON SQL Data Type

You can use the new JSON SQL data type in PL/SQL.

The new JSON data type allows you to pass JSON data from SQL to PL/SQL and back to SQL (static and dynamic). PL/SQL now supports binding directly JSON data from client side interfaces such as Oracle Call Interface (OCI) and Java Database Connectivity (JDBC), as well, as from PL/SQL to callouts. JSON can be a differentiating type in overload resolution.

Related Topics

- *Oracle® Database PL/SQL Language Reference*

SQL

- Placeholders in SQL DDL Statements
- SQL Macros
- Expression Support for Initialization Parameters
- Enhanced SQL Set Operators

Placeholders in SQL DDL Statements

SQL DDL statements can now contain placeholders instead of hard coded values for some content. For example, placeholders may be used where a username or password are required in a `CREATE USER` statement. Oracle Call Interface programs can substitute values into the DDL statement placeholders before the statements are sent to Oracle Database. This is similar to data binding, but occurs in Oracle Client.

Application security is improved because values do not need to be hard coded in SQL DDL.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

SQL Macros

You can create SQL Macros (SQM) to factor out common SQL expressions and statements into reusable, parameterized constructs that can be used in other SQL statements. SQL macros can either be scalar expressions, typically used in `SELECT` lists, `WHERE`, `GROUP BY` and `HAVING` clauses, to encapsulate calculations and business logic or can be table expressions, typically used in a `FROM` clause.

SQL macros increase developer productivity, simplify collaborative development, and improve code quality.

Related Topics

- *Oracle® Database PL/SQL Language Reference*

Expression Support for Initialization Parameters

You can specify an expression when setting the value of an initialization parameter.

In previous releases, you were required to specify an absolute value when setting an initialization parameter. You can now specify an expression that takes into account the current system configuration and environment. This is especially useful in Oracle Autonomous Database environments.

Related Topics

- *Oracle® Database Reference*

Enhanced SQL Set Operators

The SQL set operators now support all keywords as defined in ANSI SQL. The new operator `EXCEPT [ALL]` is functionally equivalent to `MINUS [ALL]`. The operators `MINUS` and `INTERSECT` now support the keyword `ALL`.

Full ANSI compliance provides greater compatibility with other database vendors and makes migration to Oracle Database easier than before.

Related Topics

- *Oracle® Database SQL Language Reference*

Database Upgrade and Utilities

- [Oracle Database Utilities](#)
- [Upgrades and Migration](#)

Oracle Database Utilities

- [Oracle Data Pump Includes and Excludes in the Same Operation](#)
- [Oracle Data Pump Resumes Transportable Tablespace Jobs](#)
- [Oracle Data Pump Parallelizes Transportable Tablespace Metadata Operations](#)
- [Oracle Data Pump Provides Optional Index Compression](#)
- [Oracle Data Pump Exports from Oracle Autonomous Database](#)
- [Oracle Data Pump Checksums Support Cloud Migrations](#)
- [Oracle SODA for C APIs Enhancements](#)
- [Oracle SQL*Loader Supports Native JSON Data Type](#)
- [Oracle SQL*Loader Support for Object Store Credentials](#)

Oracle Data Pump Includes and Excludes in the Same Operation

Starting with Oracle Database 20c, Oracle Data Pump can include and exclude objects in the same export or import operation.

Oracle Data Pump provides powerful, flexible inclusion and exclusion of objects for a job. Now, Oracle Data Pump commands can include both `INCLUDE` and `EXCLUDE` parameters in the same operation. By enabling greater specificity about what is being migrated, this enhancement makes it easier to migrate to Oracle Cloud, or to another on-premises Oracle Database.

Related Topics

- *Oracle® Database Database Utilities*

Oracle Data Pump Resumes Transportable Tablespace Jobs

Starting with Oracle Database 20c, Oracle Data Pump resumes transportable tablespace export and import jobs that are stopped.

Oracle Data Pump has the capacity to resume transportable tablespace export and import jobs. Due to errors, or other problems, you can find that transportable tablespace export or import jobs are stopped. Oracle Data Pump's capacity to resume these stopped jobs helps to save you time, and makes the system more available.

Related Topics

- *Oracle® Database Database Utilities*

Oracle Data Pump Parallelizes Transportable Tablespace Metadata Operations

Starting with Oracle Database 20c, Oracle Data Pump improves Transportable Tablespace metadata operations with parallelism.

Oracle Data Pump now supports parallel export and import operations for Transportable Tablespace (TTS) metadata. This is the information that associates the tablespace data files with the target database in a TTS migration. Parallelism improves TTS export and import performance, especially when there are millions of database objects in the data files, including tables, indexes, partitions, and subpartitions.

Related Topics

- *Oracle® Database Database Utilities*

Oracle Data Pump Provides Optional Index Compression

In Oracle Database 20c, Oracle Data Pump supports optional index compression on imports, including for Oracle Autonomous Database.

Oracle Data Pump supports adding, changing and eliminating table compression. Oracle Database 20c supports index compression as well by introducing a new `TRANSFORM` parameter clause, `INDEX_COMPRESSION_CLAUSE`. This clause enables you to control whether index compression is performed during import. Adding this clause also enables you to specify index compression on import with the autonomous services.

Related Topics

- *Oracle® Database Database Utilities*

Oracle Data Pump Exports from Oracle Autonomous Database

Starting with Oracle Database 20c, Oracle Data Pump can perform exports from Oracle Autonomous Database into dump files in a cloud object store.

Oracle Data Pump supports dump file export from cloud services to the object store. You can now migrate data from services you manage in Oracle Autonomous Database.

Related Topics

- *Oracle® Database Database Utilities*

Oracle Data Pump Checksums Support Cloud Migrations

To check Oracle Data Pump dumpfiles for validity, you can now use checksums that are added to the dumpfile.

Oracle Data Pump is used for migrating application data from on-premises Oracle Database instances into the Oracle Cloud, and also for copying dumpfiles to on-premises.

Starting with Oracle Database 20c, a checksum is now added to the dumpfile. You can use the checksum to help to confirm that the file is valid after a transfer to or from the object store and also after saving dumpfiles on on-premises and that it has no accidental or malicious changes.

Related Topics

- *Oracle® Database Database Utilities*

Oracle SODA for C APIs Enhancements

This feature introduces new SODA APIs for saving document to a collection, truncating a document collection, and specifying the array fetch size to read documents from a collection, which reduces network round-trips.

The enhancements improve interoperability of the SODA APIs with the database and utilities.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

Oracle SQL*Loader Supports Native JSON Data Type

Oracle SQL*Loader enables JSON file data loading into Oracle Database native JSON objects.

Starting with Oracle Database 20c, SQL*Loader conventional and direct path supports loading JSON data into the new native JSON data type in Oracle Database.

Related Topics

- *Oracle® Database Database Utilities*

Oracle SQL*Loader Support for Object Store Credentials

Starting with Oracle Database 20c, Oracle SQL*Loader accesses data in an object store by presenting user-defined credentials.

Oracle SQL*Loader now enables you to to specify credentials that you define, so that you can access and read data from files in an object store into Oracle Database.

Related Topics

- *Oracle® Database Database Utilities*

Upgrades and Migration

- Oracle Database Automates Database Upgrades with AutoUpgrade

Oracle Database Automates Database Upgrades with AutoUpgrade

Oracle Database AutoUpgrade enables you to upgrade one or many databases without human intervention, all with one command, and with a single configuration file.

AutoUpgrade enables you to upgrade one or many Oracle Database instances at the command-line, using a single command and a single configuration file. AutoUpgrade runs the preupgrade tasks, performs automated fix-ups where needed, runs the database upgrade, and finishes by completing post-upgrade tasks. It includes automatic retry and fallback, the option to schedule upgrades for future points in time, and the ability to set, change, or remove initialization parameters as desired. Using AutoUpgrade can save you time and money by enabling you to upgrade hundreds of databases with one command, and avoid the need for high-maintenance upgrade solutions.

Related Topics

- *Oracle® Database Database Upgrade Guide*