Oracle® Database

New Features Guide

12*c* Release 1 (12.1)

E49322-11

August 2015

Oracle Database New Features Guide is addressed to those of you familiar with previous releases of Oracle Database and who would like to become familiar with features, options, and enhancements that are new in this release of the database.

Oracle Database New Features Guide, 12*c* Release 1 (12.1)

E49322-11

Contents

3 Oracle Database 12*c* Release 1 (12.1) Documentation

List of Tables

Preface

This Preface contains these topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

Oracle Database New Features Guide is addressed to people familiar with previous releases of Oracle Database who would like to become familiar with features, options, and enhancements that are new in this release of the database.

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 11*g* Release 2 (11.2) documentation set:

- *Oracle Database Error Messages Reference*
- *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 12*c* Release 1 (12.1.0.2) New Features

This chapter contains descriptions of the features that are available in Oracle Database 12*c* Release 1 (12.1.0.2). This chapter contains the following sections:

1.1 Advanced Index Compression

Advanced Index Compression works well on all supported indexes, including those indexes that are not good candidates for the existing prefix compression feature;

including indexes with no, or few, duplicate values in the leading columns of the index.

Advanced Index Compression improves the compression ratios significantly while still providing efficient access to the index.

See Also:

Oracle Database Administrator's Guide for details

1.2 Approximate Count Distinct

The new and optimized SQL function, `APPROX_COUNT_DISTINCT()`, provides approximate count distinct aggregation. Processing of large volumes of data is significantly faster than the exact aggregation, especially for data sets with a large number of distinct values, with negligible deviation from the exact result.

The need to count distinct values is a common operation in today's data analysis. Optimizing the processing time and resource consumption by orders of magnitude while providing almost exact results speeds up any existing processing and enables new levels of analytical insight.

See Also:

Oracle Database SQL Language Reference for details

1.3 Attribute Clustering

Attribute clustering is a table-level directive that clusters data in close physical proximity based on the content of certain columns. This directive applies to any kind of direct path operation, such as a bulk insert or a move operation.

Storing data that logically belongs together in close physical proximity can greatly reduce the amount of data to be processed and can lead to better compression ratios.

See Also:

Oracle Database Data Warehousing Guide for details

1.4 Automatic Big Table Caching

In previous releases, in-memory parallel query did not work well when multiple scans contended for cache memory. This feature implements a new cache called big table cache for table scan workload.

This big table cache provides significant performance improvements for full table scans on tables that do not fit entirely into the buffer cache.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.5 FDA Support for CDBs

Flashback Data Archive (FDA) is supported for multitenant container databases (CDBs) in this release.

Customers can now use Flashback Data Archive in databases that they are consolidating using Oracle Multitenant, providing the benefits of easy history tracking to applications using pluggable databases (PDB) in a multitenant container database.

See Also:

Oracle Database Development Guide for details

1.6 Full Database Caching

Full database caching can be used to cache the entire database in memory. It should be used when the buffer cache size of the database instance is greater than the whole database size. In Oracle RAC systems, for well-partitioned applications, this feature can be used when the combined buffer caches of all instances, with some extra space to handle duplicate cached blocks between instances, is greater than the database size.

Caching the entire database provides significant performance benefits, especially for workloads that were previously limited by I/O throughput or response time. More specifically, this feature improves the performance of full table scans by forcing all tables to be cached. This is a change from the default behavior in which larger tables are not kept in the buffer cache for full table scans.

See Also:

Oracle Database Performance Tuning Guide for details

1.7 In-Memory Aggregation

In-Memory Aggregation optimizes queries that join dimension tables to fact tables and aggregate data (for example, star queries) using CPU and memory efficient `KEY VECTOR` and `VECTOR GROUP BY` aggregation operations. These operations may be automatically chosen by the SQL optimizer based on cost estimates.

In-Memory Aggregation improves performance of star queries and reduces CPU usage, providing faster and more consistent query performance and supporting a larger number of concurrent users. As compared to alternative SQL execution plans, performance improvements are significant. Greater improvements are seen in queries that include more dimensions and aggregate more rows from the fact table. In-Memory Aggregation eliminates the need for summary tables in most cases, thus simplifying maintenance of the star schema and allowing access to real-time data.

See Also:

Oracle Database SQL Tuning Guide for details

1.8 In-Memory Column Store

In-Memory Column Store enables objects (tables or partitions) to be stored in memory in a columnar format. The columnar format enables scans, joins and aggregates to perform much faster than the traditional on-disk formats for analytical style queries. The in-memory columnar format does not replace the on-disk or buffer cache format, but is an additional, transaction-consistent copy of the object. Because the column store has been seamlessly integrated into the database, applications can use this feature transparently without any changes. A DBA simply has to allocate memory to In-Memory Column Store. The optimizer is aware of In-Memory Column Store, so whenever a query accesses objects that reside in the column store and would benefit from its columnar format, they are sent there directly. The improved performance also allows more ad-hoc analytic queries to be run directly on the real-time transaction data without impacting the existing workload.

The last few years have witnessed a surge in the concept of in-memory database objects to achieve improved query response times. In-Memory Column Store allows seamless integration of in-memory objects into an existing environment without having to change any application code. By allocating memory to In-Memory Column Store, you can instantly improve the performance of their existing analytic workload and enable interactive ad-hoc data extrapolation.

See Also:

Oracle Database Administrator's Guide for details

1.9 JSON Support

This feature adds support for storing, querying and indexing JavaScript Object Notation (JSON) data to Oracle Database and allows the database to enforce that JSON stored in the Oracle Database conforms to the JSON rules. This feature also allows JSON data to be queried using a `PATH` based notation and adds new operators that allow JSON `PATH` based queries to be integrated into SQL operations.

Companies are adopting JSON as a way of storing unstructured and semi-structured data. As the volume of JSON data increases, it becomes necessary to be able to store and query this data in a way that provides similar levels of security, reliability and availability as are provided for relational data. This feature allows information represented as JSON data to be treated inside the Oracle database.

See Also:

Oracle XML DB Developer's Guide for details

1.10 New FIPS 140 Parameter for Encryption

The new database parameter, `DBFIPS_140`, provides the ability to turn on and off the Federal Information Processing Standards (FIPS) 140 cryptographic processing mode inside the Oracle database.

Use of FIPS 140 validated cryptographic modules are increasingly required by government agencies and other industries around the world. Customers who have FIPS 140 requirements can turn on the `DBFIPS_140` parameter.

See Also:

Oracle Database Security Guide for details

1.11 PDB CONTAINERS Clause

The `CONTAINERS` clause is a new way of looking at multitenant container databases (CDBs). With this clause, data can be aggregated from a single identical table or view across many pluggable databases (PDBs) from the root container. The CONTAINERS clause accepts a table or view name as an input parameter that is expected to exist in all PDBs in that container. Data from a single PDB or a set of PDBs can be included with the use of `CON_ID` in the `WHERE` clause. For example:

```
SELECT ename FROM CONTAINERS(scott.emp) WHERE CON_ID IN (45, 49);
```

This feature enables an innovative way to aggregate user-created data in a multitenant container database. Reports that require aggregation of data across many regions or other attributes can leverage the CONTAINERS clause and get data from one single place.

See Also:

Oracle Database Administrator's Guide for details

1.12 PDB File Placement in OMF

The new parameter, `CREATE_FILE_DEST`, allows administrators to set a default location for Oracle Managed Files (OMF) data files in the pluggable database (PDB). When not set, the PDB inherits the value from the root container.

If a file system directory is specified as the default location, then the directory must already exist; Oracle does not create it. The directory must have appropriate permissions that allow Oracle to create files in it. Oracle generates unique names for the files. A file created in this manner is an Oracle-managed file.

The CREATE_FILE_DEST parameter allows administrators to structure the PDB files independently of the multitenant container database (CDB) file destination. This feature helps administrators to plug or to unplug databases from one container to another in a shared storage environment.

See Also:

Oracle Database Administrator's Guide for details

1.13 PDB Logging Clause

The PDB `LOGGING` or `NOLOGGING` clause can be specified in a `CREATE` or `ALTER PLUGGABLE DATABASE` statement to set or modify the logging attribute of the pluggable database (PDB). This attribute is used to establish the logging attribute of tablespaces created within the PDB if the `LOGGING` clause was not specified in the `CREATE TABLESPACE` statement.

If a PDB `LOGGING` clause is not specified in the `CREATE PLUGGABLE DATABASE` statement, the logging attribute of the PDB defaults to `LOGGING`.

This new clause improves the manageability of PDBs in a multitenant container database (CDB).

See Also:

Oracle Database Administrator's Guide for details

1.14 PDB Metadata Clone

An administrator can now create a clone of a pluggable database only with the data model definition. The dictionary data in the source is copied as is but all user-created table and index data from the source is discarded.

This feature enhances cloning functionality and facilitates rapid provisioning of development environments.

See Also:

Oracle Database Administrator's Guide for details

1.15 PDB Remote Clone

The new release of Oracle Multitenant fully supports remote full and snapshot clones over a database link. A non-multitenant container database (CDB) can be adopted as a pluggable database (PDB) simply by cloning it over a database link. Remote snapshot cloning is also supported across two CDBs sharing the same storage.

This feature further improves rapid provisioning of pluggable databases. Administrators can spend less time on provisioning and focus more on other innovative operations.

See Also:

Oracle Database Administrator's Guide for details

1.16 PDB Snapshot Cloning Additional Platform Support

With the initialization parameter `CLONEDB` set to `true`, snapshot clones of a pluggable database are supported on any local, Network File Storage (NFS) or clustered file systems with Oracle Direct NFS (dNFS) enabled. The source of the clone must remain read-only while the target needs to be on a file system that supports sparseness.

Snapshot cloning support is now extended to other third party vendor systems.

This feature eases the requirement of specific file systems for snapshot clones of pluggable databases. With file system agnostic snapshot clones, pluggable databases can be provisioned even faster than before.

See Also:

Oracle Database Administrator's Guide for details

1.17 PDB STANDBYS Clause

The `STANDBYS` clause allows a user to specify whether a pluggable database (PDB) needs to be a part of the existing standby databases. The `STANDBYS` clause takes two values: `ALL` and `NONE`. While ALL is the default value, when a PDB is created with `STANDBYS=NONE`, the PDB's data files are offlined and marked as `UNNAMED` on all of the standby databases. Any of the new standby databases instantiated after the PDB has been created needs to explicitly disable the PDB for recovery to exclude it from the standby database. However, if a PDB needs to be enabled on a standby database after it was excluded on that standby database, PDB data files need to be copied to the standby database from the primary database and the control file needs to be updated to reflect their paths after which a user needs to connect to the PDB on that standby database and issue `ALTER PLUGGABLE DATABASE ENABLE RECOVERY` which automatically onlines all of the data files belonging to the PDB.

This feature increases consolidation density by supporting different service-level agreements (SLAs) in the same multitenant container database (CDB).

See Also:

Oracle Database Administrator's Guide for details

1.18 PDB State Management Across CDB Restart

The `SAVE STATE` clause and `DISCARD STATE` clause are now available with the `ALTER PLUGGABLE DATABASE` SQL statement to preserve the open mode of a pluggable database (PDB) across multitenant container database (CDB) restarts.

If SAVE STATE is specified, open mode of specified PDB is preserved across CDB restart on instances specified in the `INSTANCES` clause. Similarly, with the DISCARD STATE clause, the open mode of specified PDB is no longer preserved.

These new SQL clauses provide the flexibility to choose the automatic startup of application PDBs when a CDB undergoes a restart. This feature enhances granular control and effectively reduces downtime of an application in planned or unplanned outages.

See Also:

Oracle Database Administrator's Guide for details

1.19 PDB Subset Cloning

The `USER_TABLESPACES` clause allows a user to specify which tablespaces need to be available in the new pluggable database (PDB). An example of the application of this clause is a case where a customer is migrating from a non-multitenant container database (CDB) where schema-based consolidation was used to separate data belonging to multiple tenants to a CDB where data belonging to each tenant is kept in a separate PDB. The `USER_TABLESPACES` clause helps to create one PDB for each schema in the non-CDB.

This powerful clause helps convert cumbersome schema-based consolidations to more agile and efficient pluggable databases.

See Also:

Oracle Database Administrator's Guide for details

1.20 Rapid Home Provisioning

Rapid Home Provisioning allows deploying of Oracle homes based on gold images stored in a catalog of pre-created homes.

Provisioning time for Oracle Database is significantly improved through centralized management while the updating of homes is simplified to linkage. Oracle snapshot technology is used internally to further improve the sharing of homes across clusters and to reduce storage space.

See Also:

Oracle Clusterware Administration and Deployment Guide for details

1.21 Zone Maps

For full table access, zone maps allow I/O pruning of data based on the physical location of the data on disk, acting like an anti-index.

Accessing only the relevant data optimizes the I/O necessary to satisfy a query, increasing the performance significantly and reducing the resource consumption.

See Also:

Oracle Database Data Warehousing Guide for details

2

Oracle Database 12*c* Release 1 (12.1.0.1) New Features

This chapter contains descriptions of all of the features that are new to Oracle Database 12*c* Release 1 (12.1.0.1). This chapter contains the following sections:

2.1 Application Development

The following sections describe the new application development features for Oracle Database 12*c* Release 1 (12.1).

2.1.1 Enhanced Developer Productivity with Oracle Application Express

The following sections describe Oracle Application Express features.

2.1.1.1 Accessibility

Improvements have been made in the area of accessibility in existing themes and HTML templates.

By improving accessibility to applications developed in Oracle Application Express, it becomes easier for applications to meet regulatory compliance for access by users with disabilities.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.2 Automatic Purging of Workspaces

Facilities in Oracle Application Express now monitor the activity of workspaces and the applications in those workspaces over time. Administrators of unused workspaces are notified by e-mail that their workspaces and applications have not been used and are subject to being purged. After being placed in a dormant state for a period of time, a DBA and an instance administrator can approve the list of workspaces to be cleaned up, which results in the removal of the workspaces, applications and, optionally, schemas and tablespaces from an Oracle Application Express instance.

For organizations with large installations of Oracle Application Express, this feature releases unused database resources.

See Also:

Oracle Application Express Administration Guide for details

2.1.1.3 Dynamic Actions

New in this release, Oracle provides a declarative facility to incorporate JavaScript and AJAX into an application to provide rich client-side interactivity within an Oracle Application Express application. Replacing hand crafted JavaScript and AJAX with declarative definitions greatly improves the quality, consistency, and manageability of rich client-side interactivity.

This feature enables developers to declaratively define client-side behaviors without needing to know JavaScript and AJAX.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.4 End User Data Upload

With this new feature, end users can upload data into an existing table (within an application). Developers can define into which table or tables the data can be uploaded, including the unique keys to determine if a record is to be inserted or updated. In addition, the developer can specify look-ups for columns so that, for example instead of entering the `DEPTNO` or `STATUS_ID`, the end user can enter the `Department Name` or `Status Code`.

This feature allows developers to let their end users be more self-sufficient.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.5 Error Handling

Error handling and user-defined exception processing has been improved to allow developers to present user-friendly messages to users instead of database messages.

This improvement enables developers to control the error messages that are displayed to end users so they do not see errors such as `ORA-00001: unique constraint (<owner>.<name>) violated PK violated`.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.6 Expanded Interactive Reports

End users can now choose between report, icon or detail views for interactive reports. Additional support has been added for compound filters, group by, e-mail notifications, and the ability to save shared reports and download to a standalone searchable HTML file.

These enhancements provide improved interactive reports and the ability for end users to share their saved reports.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.7 Improved Charting Engine

Integrating the AnyChart 6 charting engine for the creation of improved Flash charts and the introduction of HTML5 charts, results in better looking charts that load faster. Maps and Gantt charts have also been introduced into the Oracle Application Express wizard-based chart creation. HTML5 charts are required for mobile devices that do not support Flash.

The new reporting engine is faster with improved graphics and more declarative features. This enhances the charting capabilities while making development easier.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.8 Mobile Applications

You now have the ability to declaratively define mobile applications and mobile application components including HTML5 charts, HTML5 item types, and mobile calendars. This feature also allows you to facilitate applications having both desktop and mobile user interfaces with automatic detection. The mobile applications are built using jQuery Mobile.

This enhancement makes development of mobile applications fast and declarative. Instead of building separate applications for different mobile operating systems (for example, iOS, Android, Blackberry, and Windows), the same application can be run on any mobile device by incorporating jQuery Mobile.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.9 Modernized Application Builder

Numerous usability improvements have been added to Oracle Application Express including integrated application-wide search, an advisor that inspects customer applications for common errors and security issues, dashboards throughout the product, and improved Administration screens.

The improvements to the Application Builder make the tool more intuitive and easier to learn.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.10 Packaged Applications

A collection of productivity applications allows users to immediately start utilizing their database investment.

Being provided with a number of productivity and sample applications, developers can start using Oracle Application Express as soon as it is installed to improve their business processes. They can also unlock these applications to learn about Oracle Application Express development best practices for developing such applications.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.11 Plug-Ins

This features enables development of and the ability to share custom region types, item types, dynamic actions, authentication, and authorizations. This dramatically broadens the reach of Oracle Application Express applications and provides a library of features for Oracle Application Express. When developers require functionality not available with native components, this architecture allows them to extend their applications in a manner that is both supported and maintained.

This feature provides a supported means by which to extend the built-in Oracle Application Express capabilities.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.12 Tabular Forms

Expanded tabular form functionality allows developers to declaratively define validations and processes using column values. This enhancement also adds support in tabular forms for additional display types (for example, checkboxes, popup Key LOVs, and radio groups).

Rather than having to code custom code and use a custom item type to be able to perform validations on tabular forms, developers can now reference columns within validations and processes.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.13 Team Development

A suite of tools natively integrated into Oracle Application Express to help developers plan and manage their application development of Oracle Application Express applications is now available. This also includes features to gather feedback in an Oracle Application Express application and process it as a to-do item, a bug, or a feature request.

This feature allows development teams to streamline their development process.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.14 Themes and Templates

Each of the modern themes in Oracle Application Express has been revised and modernized. This allows applications to appear more modern, make use of gradients, provide more XHTML-conformant templates, have more enhanced browser compatibility, and improved accessibility. Theme 25 is a new theme designed to utilize Responsive Design so that regions and items automatically adjust based on the size of the window. Theme 26 mirrors the theme used for the new packaged applications introduced in Oracle Application Express 4.2. A completely new theme has been included for mobile smart phones to allow developers to readily build applications designed to run on any mobile device.

The revised themes allow for more modern looking applications that are easier to customize as they are DIV based instead of being based on HTML tables.

> **See Also:**
>
> *Oracle Application Express Application Builder User's Guide* for details

2.1.1.15 Time Stamp and Time Zone Support

Support is added for `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE` and `TIMESTAMP WITH LOCAL TIME ZONE` data types throughout Oracle Application Express. Declarative functionality is also added to automatically derive an end user's time zone and set it in the Oracle Application Express session, enabling the easy creation of time zone-sensitive applications.

The ability to utilize time stamps and time zones throughout the application is important for any application that records dates and times. This is especially true for applications that are accessed globally.

> **See Also:**
>
> *Oracle Application Express Application Builder User's Guide* for details

2.1.1.16 Use of ROWID

`ROWID` can now be used for automatic DML processing (as an alternative to identifying the primary key columns).

Using `ROWID` instead of a constrained number of primary key columns allows developers to utilize the standard wizards when defining forms and reports based on tables with more than two primary key columns. This is particularly important in commercial off-the-shelf (COTS) applications, such as PeopleSoft.

> **See Also:**
>
> *Oracle Application Express Application Builder User's Guide* for details

2.1.1.17 Web Services Support

Web services support has been modernized within Oracle Application Express. Some specific features include:

- Creating a PL/SQL API to interact with Web services.
- Exposing report regions, DML processes as Representational State Transfer (REST) Web services.
- Support of binary data types in Web services support.
- Allow the inclusion of custom Simple Object Access Protocol (SOAP) headers with Web Services Description Language (WSDL) based Web services.

- Improving the WSDL parsing engine.
- Support of SOAP 1.2 in wizard-based Web services support.

These enhancements provide the ability to integrate Oracle Application Express with Representational State Transfer (REST) Web services and integrate applications into a Simple Object Access Protocol (SOAP) environment.

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.1.18 Websheets

Websheets are a new class of application development within Oracle Application Express, lowering the bar even further to manage data in an Oracle database from a Web browser. Using only a Web browser, end users can define pages, data grids and reports. With the data grids, they can do inline editing, add lists of values, and add validations and then select the community that can see and edit their data.

This feature allows business users to combine textual content (similar to a wiki) with data (for example, data grids and queries against tables in their Oracle schema).

See Also:

Oracle Application Express Application Builder User's Guide for details

2.1.2 Enhanced Globalization Support

Oracle provides enhanced support for building fully globalized enterprise applications including the latest Unicode Standard compliance, database migration to the Unicode character set, linguistic collation support, and infrastructure for application data multilingual support. The following sections describe the enhanced globalization support features.

2.1.2.1 Expanded Database Locale Support

A set of new locales (approximately 10 languages and 30 territories) is now supported in Oracle Database 12*c* Release 1 (12.1) to improve the overall locale coverage and address customer requirements.

This feature improves the database locale coverage to provide behavior that meets local users' cultural conventions.

See Also:

Oracle Database Globalization Support Guide for details

2.1.2.2 Replacement of CSSCAN and CSALTER with DMU

The Database Migration Assistant for Unicode (DMU) provides a streamlined end-to-end solution for migrating databases from legacy character sets to the Unicode character set. It is shipping with Oracle Database 12*c* Release 1 (12.1) and becomes the officially supported method for migration to the Unicode character set. The legacy Database Character Set Scanner (CSSCAN) and CSALTER utilities are removed from the database installation and have been desupported. The DMU also supports migrating selected prior database releases of 10.2, 11.1, and 11.2. More details are available at the OTN DMU page located at:

http://www.oracle.com/technetwork/database/database-technologies/globalization/dmu/overview/index.html

See Also:

Oracle Database Globalization Support Guide for details

2.1.2.3 Unicode 6.1 Support

The National Language Support (NLS) data files for AL32UTF8 and AL16UTF16 character sets have been updated to match version 6.1 of the Unicode Standard character database.

With this enhancement, as of August 2012, Oracle Database conforms to the latest version of the Unicode Standard.

See Also:

Oracle Database Globalization Support Guide for details

2.1.2.4 Unicode Collation Algorithm Conformance

Database linguistic sorting and searching support has been enhanced to conform to the Unicode Collation Algorithm (UCA) and ISO 14651 international collation standard.

A UCA-compliant implementation achieves better multilingual sorting behavior for all languages and increases industry compatibility.

See Also:

Oracle Database Globalization Support Guide for details

2.1.3 General

The following sections describe new general features.

2.1.3.1 Import and Export of Workspace Manager Schema

It is now possible to import and export Workspace Manager schema (all of the schemas that contain a version-enabled table or a parent table in a referential integrity constraint of a version-enabled table, as well as any internal Workspace Manager metadata). In addition, full import and export of databases with Workspace Manager enabled tables is now supported across different versions of Oracle Database.

This greatly simplifies the upgrade, management and administration of databases with Workspace Manager enabled tables.

See Also:

Oracle Database Workspace Manager Developer's Guide for details

2.1.3.2 Performance Improvements to Workspace Operations and Views

`DIFF` and `CONF` views have been reorganized to enable Oracle Database optimizer to generate more efficient SQL plans when working with Workspace enabled tables. In addition, user-defined hints in Workspace Manager views may now be included.

Changes have also been made to `MergeWorkspace` and DML inserts to reduce execution time.

These changes enable Workspace Manager enabled tables to scale efficiently and support extremely large tables (up to 200 million rows) and to deliver improved query response times.

2.1.4 Improved Oracle SQL and PL/SQL

The following sections describe the improved Oracle SQL and PL/SQL features.

2.1.4.1 An Invoker's Rights Function Can Be Result Cached

Through Oracle Database 11*g* Release 2 (11.2), only definer's rights PL/SQL functions could be result cached. Now, invoker's rights PL/SQL functions can also be result cached. (The identity of the invoking user is implicitly added to the key of the result.)

At times, it may be appropriate to use an invoker's rights PL/SQL function to issue one or more `SELECT` statements. This feature improves performance.

See Also:

Oracle Database PL/SQL Language Reference for details

2.1.4.2 An Object of Type LIBRARY Can Be Defined Using an Object of Type DIRECTORY

In previous releases, an object of the `LIBRARY` type could only be defined by using an explicit path. However, now the `DIRECTORY` type can be the single point of maintenance for file system paths. Moreover, using a `DIRECTORY` type has security benefits. A directory object can be defined using a `DIRECTORY` type.

Additionally, the definition of an object of the `LIBRARY` type can now include a credential so that the designated external program can be run as a different operating system user than the owner of the Oracle installation.

These enhancements improve security and portability of an application that uses external procedures.

See Also:

Oracle Database Security Guide for details

2.1.4.3 Enhanced Oracle Native LEFT OUTER JOIN Syntax

In previous releases of Oracle Database, in a query that performed outer joins of more than two pairs of tables, a single table could be the null-generated table for only one other table. Beginning with Oracle Database 12*c*, a single table can be the null-generated table for multiple tables.

Prior to Oracle Database 12*c*, having multiple tables on the left hand side of an outer join was illegal and resulted in an `ORA-01417` error. The only way to execute such a query was to translate it into ANSI syntax. In Oracle Database 12*c*, the native syntax for a `LEFT OUTER JOIN` has been expanded to allow multiple tables on the left hand side. This expansion provides the following benefits:

- Merging of multiple table views on the left hand side of an outer join. Such views can originate from the user query or they may be generated during conversion from `LEFT OUTER JOIN` syntax.
- Merging of such views allows more join reordering and, therefore, more optimal execution plans. These views are merged in a heuristic manner without having to go through cost-based query transformation.
- It relieves the application developers from the burden of formulating their queries in terms of views or `LEFT OUTER JOIN` syntax.

See Also:

Oracle Database SQL Language Reference for details

2.1.4.4 JDBC Support for PL/SQL Data Types as Parameters

The ability for Java and JDBC applications to bind PL/SQL package types and boolean types as parameters is available in this release.

This feature improves ease-of-use, seamless mapping and exchange of PL/SQL types with Java types, and increases Java developer productivity.

2.1.4.5 Mechanism to Restrict the Ability to Reference a PL/SQL Unit to a White List of Database Objects

It is now possible to mark a schema-level function, procedure, package, or type specification with a white list of allowed callers. The allowed caller may be a trigger object type that can invoke a PL/SQL subprogram, but it must be in the same schema as the unit that has the white list. The white list is optional but, when used, only the listed objects may reference the unit in question. It is possible to specify the schema of a referenced object explicitly, thereby, allowing cross-schema calls.

This capability supports the robust implementation of a module, consisting of a main unit and helper units, by allowing the helper units to be inaccessible from anywhere except the unit they are intended to help.

See Also:

Oracle Database PL/SQL Language Reference for details

2.1.4.6 Native Client API Support for PL/SQL Package Types and Boolean Types as Parameters

This feature allows database client APIs (for example, OCI and JDBC) to natively describe and bind PL/SQL package types and boolean types. Java and C-based applications can now easily bind and execute PL/SQL functions or procedures with PL/SQL package types or boolean types as parameters.

This feature reduces the complexity of executing PL/SQL functions or procedures from client-side applications.

See Also:

Oracle Database Development Guide for details

2.1.4.7 New PL/SQL DBMS_UTILITY.EXPAND_SQL_TEXT Procedure

The `DBMS_UTILITY.EXPAND_SQL_TEXT` procedure accepts a subquery that references views and returns a subquery with the identical meaning that references only tables.

This functionality can help in the analysis of SQL which depends on views with the aim of fixing application logic or resolving performance issues.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

2.1.4.8 New PL/SQL Package UTL_CALL_STACK

The `UTL_CALL_STACK` package provides subprograms to return the current call stack for a PL/SQL program.

It is functionally similar to the existing DBMS_UTILITY.FORMAT_CALL_STACK procedure which returns information as a human-readable essay. This new package makes this information available in a structured representation amenable for programmatic analysis.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

2.1.4.9 New Predefined PL/SQL Inquiry Directives

The $$PLSQL_UNIT_OWNER and $$PLSQL_UNIT_TYPE predefined PL/SQL inquiry directives are now supported in this release.

Through Oracle Database 11*g* Release 2 (11.2), the predefined inquiry directives, $$PLSQL_LINE and $$PLSQL_UNIT, allowed diagnostic code to identify the current PL/SQL statement, but with a certain ambiguity. This ambiguity is now removed.

See Also:

Oracle Database PL/SQL Language Reference for details

2.1.4.10 New SCHEMA Parameter for DBMS_SQL.PARSE() Procedure

DBMS_SQL.PARSE() procedure has a new SCHEMA parameter. It specifies the schema in which to resolve unqualified object names.

This allows a definer's rights unit to control the name resolution for the dynamic SQL it issues.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

2.1.4.11 PL/SQL Functions Defined in the SQL WITH Clause

You can define a PL/SQL function in the WITH clause of a subquery and use it as an ordinary function beginning with this release.

The procedural logic needed to support a SQL statement is encapsulated with the SQL statement. This is particularly useful in a read-only database.

Using this construct results in better performance as compared with schema-level functions.

See Also:

Oracle Database SQL Language Reference for details

2.1.4.12 PL/SQL-Specific Data Types Allowed Across the PL/SQL-to-SQL Interface

Through Oracle Database 11*g* Release 2 (11.2), when PL/SQL invoked SQL, only values with data types supported by SQL could be bound. This restriction applied even when the called SQL was a PL/SQL anonymous block. This restriction is removed in Oracle Database 12*c* Release 1 (12.1). For example, a PL/SQL subprogram with a formal parameter whose data type is BOOLEAN can now be invoked dynamically using an anonymous block.

Other restrictions are also removed. The table operator can now be used in a PL/SQL program on a collection whose data type is declared in PL/SQL. This also allows the data type to be a PL/SQL associative array. (In prior releases, the collection's data type had to be declared at the schema level.)

The removal of these restrictions increases the power of expression and the usefulness of PL/SQL. In particular, the extended flexibility of the table operator allows code written to run other vendors' stored procedure languages to be easily migrated to PL/SQL.

See Also:

Oracle Database PL/SQL Language Reference for details

2.1.4.13 Precompilers Support for SQL Plan Management

There are new command-line options for the generation of plan baseline SQL statements providing control of the name and format of generated SQL files and log files.

This support avoids performance regression of SQL statement execution and provides easier upgrade of precompiler applications.

See Also:

*Pro*C/C++ Programmer's Guide* for details

2.1.4.14 SQLJ Support for SQL Plan Management

The following are new features in this release for SQLJ support for SQL plan management (SPM):

- Command line and property file options for the generation of plan baseline SQL statements.
- Generation of a SQL file containing the statements for creating SPM plans.
- Control in the naming of generated log files and Java files

This new support helps make the upgrade of SQLJ applications easier and helps to avoid performance regression of SQL statement execution.

See Also:

Oracle Database SQLJ Developer's Guide for details

2.1.4.15 Temporal Validity

With Temporal Validity, you can add one or more valid time dimensions to a table using existing columns, or using columns automatically created by the database.

Applications often indicate the validity of a fact recorded in the database with dates or time stamps that are relevant to the underlying business they manage. Examples of such dates include the hire date and termination date of an employee in a Human Resources application, the effective date range of coverage for an insurance policy, or the time duration for a stock price. Temporal Validity reduces the complexity of application code by providing a simple declarative interface to allow applications to manage the validity of rows.

See Also:

Oracle Database Development Guide for details

2.1.4.16 Temporal Validity Flashback Queries

Flashback Query has been extended to support queries on Temporal Validity dimensions. Users can now execute queries with the `AS OF` and `VERSIONS BETWEEN` clauses based on one or more valid time periods on the underlying tables. Flashback

Queries that combine Temporal Validity and Transaction Time Temporal (tracked using Flashback Data Archive) are called bi-temporal queries.

Users can now query data based on current values (that is, `CURRENT` in valid time and transaction time), what we know now (that is, `AS OF` in valid time; `CURRENT` in transaction time), or what we knew before (that is, `AS OF` in valid time and transaction time), giving declarative access to all possible views of data based on the two time dimensions. Bi-temporal queries in Oracle Database 12*c* Release 1 (12.1) provide functionality previously available only with extensive and complex application code.

See Also:

Oracle Database Development Guide for details

2.1.5 OCI/OCCI Enhancements

The following sections describe data access features and support for SQL queries performed through applications with a Web-based interface.

2.1.5.1 Auto-Tuning Oracle C Client Interface (OCI) Applications

Oracle Database 12*c* Release 1 (12.1) introduces a new client-side auto-tuning feature.

This feature provides automatic and transparent performance management.

See Also:

Oracle Call Interface Programmer's Guide for details

2.1.5.2 Oracle C/C++ Client (OCI/OCCI) Support for Row Count Per Iteration for Array DML

This feature provides support for C and C++ interfaces for retrieving the number of rows affected by each iteration of an array DML statement separately in an array buffer provided by the user.

This feature improves data access (for example, reliability, quality control, and ease of debugging) and support for SQL queries performed through applications with a Web-based interface.

See Also:

Oracle Call Interface Programmer's Guide for details

2.1.6 Reduced Cost and Complexities of Migrating to Oracle

The following sections describe features affecting the cost and complexities of migrating to Oracle.

2.1.6.1 Default Values for Columns Based on Oracle Sequences

Default values for columns can directly refer to Oracle sequences. Valid entries are `sequence.CURVAL` and `sequence.NEXTVAL`.

Providing the functionality to directly refer to a sequence as a default value expression simplifies code development.

See Also:

Oracle Database SQL Language Reference for details

2.1.6.2 DEFAULT Values for Columns on Explicit NULL Insertion

The `DEFAULT` definition of a column can be extended to have the `DEFAULT` being applied for explicit `NULL` insertion.

The `DEFAULT` clause has a new `ON NULL` clause, which instructs the database to assign a specified default column value when an `INSERT` statement attempts to assign a value that evaluates to `NULL`.

> **See Also:**
>
> *Oracle Database SQL Language Reference* for details

2.1.6.3 IDENTITY Columns

Table columns have been enhanced to support the American National Standards Institute (ANSI) SQL keyword `IDENTITY`.

This provides a standards based approach to the declaration of automatically incrementing columns simplifying application development and making the migration of DDL to Oracle simpler.

> **See Also:**
>
> *Oracle Database SQL Translation and Migration Guide* for details

2.1.6.4 Increased Size Limit for VARCHAR2, NVARCHAR2, and RAW Data Types

The maximum size of the `VARCHAR2, NVARCHAR2,` and `RAW` data types has been increased from 4,000 to 32,767 bytes.

Increasing the allotted size for these data types allows users to store more information in character data types before switching to large objects (LOBs). This is especially useful for brief textual data types and the capabilities to build indexes on these types of columns.

> **See Also:**
>
> *Oracle Database SQL Language Reference* for details

2.1.6.5 JDBC Support for Sybase Applications Migration

JDBC support for Sybase applications migration includes the following new APIs:

- `oracle.jdbc.sqlTranslationProfile`
- `oracle.jdbc.sqlErrorTranslationFile`
- `oracle.jdbc.OracleTranslatingConnection`

Also included is the configuration file `SQLErrorTranslation.xml`.

These new API's reduce the costs and complexities of migrating Sybase Java applications to Oracle.

> **See Also:**
>
> *Oracle Database SQL Translation and Migration Guide* for details

2.1.6.6 Implicit Result Sets

Before Oracle Database 12*c* Release 1 (12.1), a `SELECT` statement embedded as static SQL in a PL/SQL program, and run in the database, had to return its results into PL/SQL variables in that program using either an `INTO` clause, a `BULK COLLECT INTO` or `BULK FETCH INTO` clause, or a `CURSOR FOR LOOP` clause. The client then accessed these results using suitably defined scalar or composite bind arguments. Alternatively, the SELECT statement could be used to return a REFCURSOR to the client to allow it to manage the fetching of the results.

Now PL/SQL adds the equivalent capability as that provided in other vendors' environments to allow bare-bones SELECT statements to pass back their results to the client. When code is migrated to Oracle Database from other vendors' environments, the capability removes the need to rewrite code that takes advantage of implicit result set communication.

See Also:

Oracle Database SQL Translation and Migration Guide for details

2.1.6.7 Native SQL Support for Query Row Limits and Row Offsets

The FETCH FIRST and OFFSET clauses provides native SQL language support to limit the number of rows returned and to specify a starting row for the return set.

Many queries need to limit the number of rows returned or offset the starting row of the results. For example, top-N queries sort their result set and then return only the first n rows. FETCH FIRST and OFFSET simplify syntax and comply with the ANSI SQL standard.

See Also:

Oracle Database SQL Language Reference for details

2.1.6.8 Oracle Database Driver for MySQL Applications

The Oracle Database driver for MySQL applications is a drop-in replacement for the client library for MySQL 5.5. It enables applications and tools built using the MySQL C API to run against an Oracle Database using the new library that implements the MySQL C API.

The key benefits are the reuse of MySQL applications against both MySQL and Oracle and the reduction in the costs and complexities of migrating MySQL applications to Oracle.

See Also:

Oracle Database SQL Translation and Migration Guide for details

2.1.6.9 Precompiler Support for Prefetch By Memory

New command-line options allow Pro*C and Pro*COBOL to limit the amount of memory used for prefetching rows.

This feature provides resource control and reduces the costs and complexities of migrating DB2 applications to Oracle.

See Also:

*Pro*COBOL Programmer's Guide* for details

2.1.6.10 SQL CROSS APPLY, OUTER APPLY and LATERAL

The APPLY SQL syntax allows a table-valued function to be invoked for each row returned by a query's outer table expression. The table-valued function acts as the right input; the outer table expression acts as the left input. The right input is evaluated for each row from the left input and the rows produced are combined for the final output. Therefore, one can pass left-correlations to the table-valued functions.

There are two forms of `APPLY` - `CROSS APPLY` and `OUTER APPLY`. `CROSS APPLY` returns only rows from the outer table that produce a result set from the table-valued function. `OUTER APPLY` returns both rows that produce a result set, and rows that do not, with `NULL` values in the columns produced by the table-valued function.

`LATERAL`, part of the ANSI standard, is an extension of the inline view syntax that provides left-correlation scoping within the inline view. These new keywords provide easier and more flexible ways to evaluate and return SQL query results.

See Also:

Oracle Database SQL Language Reference for details

2.1.6.11 SQL Translation Framework

A new mechanism is provided to allow the text of a SQL statement, submitted from a client program using an open application programming interface (API) such as ODBC or JDBC, to be translated by user-supplied code before it is submitted to the Oracle Database SQL compiler. The translation code is named and installed in the database using a PL/SQL API. It can be implemented programmatically, by look-up, or by a suitable mixture of these. The name of the translator is specified at connect time. The mechanism also allows Oracle error codes and American National Standards Institute (ANSI) `SQLSTATES` to be translated by user-supplied code.

The motivating use case is to allow extant client-side application code, written for a different vendor's database (and therefore for a SQL dialect other than Oracle's), to run unchanged against an Oracle database by emulating the syntax and semantics of the other SQL dialect thereby greatly reducing the cost of migration. Additionally, this feature can satisfy any other use case where it is expedient to intervene between the SQL statement that the client submits and what is actually executed.

See Also:

Oracle Database SQL Translation and Migration Guide for details

2.1.7 Support .NET and Microsoft Development Community

The following sections describe new features for the .NET and Microsoft development community.

For more information about new features in Oracle Developer Tools for Visual Studio (ODT), see the ODT online help section titled "New Features for Oracle Developer Tools for Visual Studio". This online help is installed with the product.

2.1.7.1 Microsoft .NET Framework 4 and 4.5 Support

Oracle Data Provider for .NET supports .NET Framework 4 and 4.5, including Client Profile.

2.1.7.2 Oracle TimesTen In-Memory Database

Oracle Data Provider for .NET enables fast data access for any .NET application to Oracle TimesTen In-Memory Databases. ODP.NET support for TimesTen includes the classes, enumerations, interfaces, delegates, and structures of the `Oracle.DataAccess.Client` and `Oracle.DataAccess.Types` namespaces.

ODP.NET supports TimesTen Release 11.2.1.6.1 or later on Microsoft Windows 32-bit and 64-bit platforms. TimesTen can be used with .NET Framework 2.0, 3.0, 3.5, and 4 with Microsoft Visual Studio 2005 or later.

2.1.7.3 64-Bit ODP.NET XCopy for Windows x64

Now available for Microsoft Windows x64 systems, ODP.NET XCopy provides system administrators with a smaller client install size than the standard ODP.NET client, and is easier to configure.

ODP.NET XCopy simplifies embedding ODP.NET in customized deployment packages.

2.1.7.4 Entity Framework and LINQ

Entity Framework is a framework for providing object-relational mapping and services on data models. It tries to solve the impedance mismatch between the database format (relational) and the client's preferred format (object).

Language Integrated Query (LINQ) defines a set of operators that can be used to query, project and filter data in arrays, enumerable classes, XML, relational databases, and other data sources. One form of LINQ, LINQ to Entities, allows querying of Entity Framework data sources.

Oracle Data Provider for .NET (ODP.NET) supports Entity Framework so that Oracle Database can participate in object-relational modeling and LINQ to Entities queries.

Entity Framework and LINQ provide numerous productivity benefits for the .NET developer. It abstracts the database's data model from the application's data model. Working with object-relational data becomes easier with Entity Framework's tools. Oracle's integration with Entity Framework and LINQ allows Oracle .NET developers to take advantage of all these productivity benefits.

See Also:

Oracle Data Provider for .NET Developer's Guide for Microsoft Windows for details

2.1.7.5 Implicit REF CURSOR Parameter Binding

Oracle Data Provider for .NET (ODP.NET) can bind `REF CURSOR` parameters for stored procedures without binding them explicitly. To do so, the application must provide the `REF CURSOR` metadata as part of the .NET configuration file.

This feature allows Entity Framework function import to call Oracle stored procedures and return `REF CURSOR` result sets. ODP.NET can also update the database's data with a data set or data table obtained through a `REF CURSOR`.

In Entity Framework, result set parameters are generally not declared. By supporting the implicit `REF CURSOR` parameter, ODP.NET more closely integrates with typical Entity Framework usage scenarios.

See Also:

Oracle Data Provider for .NET Developer's Guide for Microsoft Windows for details

2.1.7.6 Oracle SQL Supports APPLY Keyword

Language Integrated Query (LINQ) is a .NET querying language. At runtime, LINQ is translated into native database SQL before it can query the database. In some circumstances, LINQ uses the non-standard `APPLY` keyword in its SQL translation for retrieving lateral views. Oracle Database and ODP.NET support the `APPLY` keyword in Oracle Database 12*c* Release 1 (12.1) to more fully support LINQ.

This feature allows the occasional LINQ query that uses SQL `APPLY` to work seamlessly with ODP.NET and Oracle Database for lateral views.

See Also:

Oracle Data Provider for .NET Developer's Guide for Microsoft Windows for details

2.1.7.7 Returning Row Counts

When using array binding to execute multiple DML statements, Oracle Data Provider for .NET (ODP.NET) now provides an array that lists the number of rows affected for each input value from the bound array, rather than just the total number of rows affected. This information provides more detailed feedback for the application developer. To retrieve the row count, ODP.NET can call the `OracleCommand.ArrayBindRowCount` property.

With more detailed feedback on the array bound DML execution, the developer can better evaluate the query's efficiency and whether the data changes were correctly applied.

See Also:

Oracle Data Provider for .NET Developer's Guide for Microsoft Windows for details

2.1.7.8 Windows Communication Foundation (WCF) Data Services and Open Data Protocol (OData)

WCF Data Services enables developers to create services that use OData to expose and consume data over the internet by using the semantics of representational state transfer (REST). OData exposes data as resources that are addressable by universal resource identifiers (URIs). OData uses Entity Data Model conventions to expose resources as sets of entities that are related by associations. Through its support of Entity Framework, ODP.NET can expose its data using OData and WCF Data Services.

WCF Data Services and OData facilitate creating flexible data services from any data source and naturally integrating them with the Web. It allows all types of data sources, including Oracle databases, to be used by the same data sharing standard making data exchange more interoperable.

See Also:

Oracle Data Provider for .NET Developer's Guide for Microsoft Windows for details

2.1.8 Support Java Development Community

The following sections describe new features for the Java development community.

For information on additional features that Java Database Connectivity (JDBC) supports, see the following topics:

- Section 2.1.4.4, "JDBC Support for PL/SQL Data Types as Parameters"
- Section 2.1.4.6, "Native Client API Support for PL/SQL Package Types and Boolean Types as Parameters"
- Section 2.1.6.4, "Increased Size Limit for VARCHAR2, NVARCHAR2, and RAW Data Types"
- Section 2.5.1.1, "Application Continuity for Java"

- Section 2.5.1.2, "Transaction Guard"
- Section 2.8.1.2, "Very Large Network Buffers"

2.1.8.1 JDBC Support for Monitoring Database Operations (DBOP)

Oracle Database 12*c* Release 1 (12.1) introduces the `DBOP` tag that can be associated with a thread in the application when the application does not have explicit access to a database. The `DBOP` tag is associated with a thread through the invocation of either `setClientInfo()` method or Oracle Dynamic Monitoring Services (DMS) APIs, without requiring an active database connection or a client/server round trip.

See Also:

Oracle Database JDBC Developer's Guide for details

2.1.8.2 Database Support for JDK Upgrade

New support in this release allows a customer to upgrade the embedded Java VM runtime to newer Java Development Kit (JDK) releases (for example, upgrade from JDK 1.6 to JDK 1.7 and conversely downgrade from JDK 1.7 to JDK 1.6). In addition, customers can choose a target JDK during database installation.

Using the latest Java standards can reduce costs by improving productivity and reusing Java classes or libraries. This feature allows compatibility with the latest Java standards.

See Also:

Oracle Database Java Developer's Guide for details

2.1.8.3 Database Support for Latest Java SE and Standard Utilities

Oracle Database supports JDK 1.6, JDK 1.7, Java Naming and Directory Interface (JNDI), Java Logging, and the Java SE Integration Libraries such as RMI-IIOP and scripting.

The JDK support reduces the cost and improves productivity through reuse of Java classes or libraries. Compatibility with the latest Java standards allows portability and use of client-side Java classes and libraries directly in the database.

See Also:

Oracle Database Java Developer's Guide for details

2.1.8.4 Enhanced Security for Java in the Database

Oracle Database includes enhanced permission and policy management for Java runtime. The Java policy can be reloaded by the system administrator after adding third-party encryption suites. In addition, the database administrator can change the algorithm search order.

These enhancements provide tighter permission and policy management, as well as flexible and advanced security support for third-party encryption libraries.

See Also:

Oracle Database Java Developer's Guide for details

2.1.8.5 JDBC Enhanced Security

Java Database Connectivity (JDBC) supports the security enhancements in Oracle Database including Kerberos authentication and Windows Authentication (NTS).

This feature provides advanced security for Java applications.

See Also:

Oracle Database JDBC Developer's Guide for details

2.1.8.6 JDBC Support for Database Resident Connection Pool

Database Resident Connection Pool (DRCP) is a pool of dedicated servers, enabled on the database server and shared across client applications, programming languages, and middle tiers. Once enabled on the database, new connection properties `oracle.jdbc.DRCP.name` and `oracle.jdbc.DRCP.purity` allow Java and JDBC applications to use it transparently using client-side connection pools (for example, Universal Connection Pool). New public methods under `oracle.jdbc.pool.OraclePooledConnection` are introduced to expose this feature to client pool developers.

This feature allows large scale deployment of Java applications (typically hundreds or thousands of middle tier connecting to the same database). Orders of magnitude reductions of database server processes and memory are seen with this new feature.

See Also:

Oracle Database JDBC Developer's Guide for details

2.1.8.7 JDBC Support for Latest Java Standards

Now there is support for row count on each iteration for array DML, JDBC 4.1 specification, `ParameterMetaData, getClientInfo,` and `setClientInfo` API.

Full compliance with Java standards ensures portability of foreign Java applications to Oracle or applications built with Oracle JDBC.

See Also:

Oracle Database JDBC Developer's Guide for details

2.2 Business Intelligence and Data Warehousing

The following sections describe the new business intelligence and data warehousing features for Oracle Database 12*c* Release 1 (12.1).

2.2.1 Oracle Advanced Analytics

The following sections describe new Oracle Advanced Analytics features.

2.2.1.1 Decision Tree Mining Text Data

The Decision Tree algorithm now supports nested data and can be used for text mining.

Decision Tree is popular due to its transparency and prevalence, therefore, it is important to enable the algorithm to handle unstructured data.

See Also:

Oracle Data Mining User's Guide for details

2.2.1.2 Expectation Maximization (EM) Clustering and Density Estimation

In Release 11*g*, Oracle Data Mining offered two clustering algorithms. However, these algorithms did not easily integrate data coming from different domains (for example, structured and unstructured data). Expectation Maximization (EM) is a probabilistic clustering algorithm that creates a density model of the data. The density model allows for an improved approach to combining data originating in different domains. Each domain can be modeled by distributions appropriate for the domain. The distribution parameters are optimized to provide the most likely joint distribution of the data. Given EM's probabilistic nature, its cluster assignment probabilities are more reliable than those produced by the current Oracle Data Mining algorithms. The EM algorithm also automatically determines the optimal number of clusters needed to model the data.

In bringing analytics to applications, Oracle Data Mining provides different types of clustering capabilities currently being used by multiple applications. While the current capabilities solve a range of problems, an additional method is needed that can effectively combine data from different domains, such as sales transactions and customer demographics, or structured and unstructured (for example, text) data, as well as help answer queries involving range and equality predicates. Expectation Maximization can address all of these requirements.

See Also:

Oracle Data Mining Concepts for details

2.2.1.3 Feature Extraction Using Singular Value Decomposition

Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) are powerful feature extraction methods that use orthogonal linear projections to capture the underlying variance of the data. This property is extremely useful for reducing the dimensionality of high-dimensional data and for supporting meaningful data visualization. Text mining is one of the domains where SVD projections have found wide application.

PCA can be viewed as a special scoring method under the SVD algorithm. It produces projections that are scaled with the data variance. Projections of this type are sometimes preferable in feature extraction to the standard non-scaled SVD projections.

In bringing analytics to applications, Oracle Data Mining provides powerful feature extraction capabilities that can be used in many contexts, special handling of unstructured data, and large numerical data sets such as those from sensors (for example, Radio Frequency Identification (RFID)) and time series.

While Oracle Data Mining already provides a basic feature extraction capability, additional feature extraction methods capable of scaling to large data sizes (both rows and attributes) and allowing greater compression of the data are necessary to support many applications.

See Also:

Oracle Data Mining Concepts for details

2.2.1.4 Feature Selection and Creation for Generalized Linear Models (GLM)

Feature selection is used to reduce the number of predictors used by a model. This allows for smaller, faster scoring, and more meaningful Generalized Linear Models (GLM).

Feature generation allows the creation of GLM models that use non-linear terms (up to cubic terms). This produces more powerful, transparent models.

In bringing analytics to applications, Oracle Data Mining continuously strives to address the competing goals of high accuracy and transparency (the ability to explain predictions).

Some of the techniques (for example, GLM) used most often by applications provide great transparency at the expense of lower accuracy than less transparent methods. There is a need for a transparent, highly accurate, and scalable method capable of handling thousands of attributes efficiently. This can be achieved by adding feature selection and creation to GLM.

See Also:

Oracle Data Mining Concepts for details

2.2.1.5 Native Double in Data Mining Functions

Support has been added for native double types (`BINARY_DOUBLE` and `BINARY_FLOAT`) in Oracle Data Mining functions.

Mining model deployment (scoring) performance is critical because this is run on the majority of production data, both in batch and real time. Through C performance analysis, Oracle has identified that the cost of scoring can be dominated by type coercion between Oracle number and double, rather than by the model itself. Removing this overhead leads to much faster scoring behavior.

See Also:

Oracle Data Mining User's Guide for details

2.2.1.6 Native SQL Support for Row Pattern Matching

The `MATCH_RECOGNIZE` clause enables native SQL queries to match specified patterns in sequences of rows.

Row pattern matching in native SQL improves application and development productivity and query efficiency for row sequence analysis. The syntax incorporates regular expressions and full conditional logic, enabling precise and flexible pattern definition. Whatever the domain (for example, financial market prices, internet clicks, or security sensor output), applications analyzing row sequences can benefit from `MATCH_RECOGNIZE`.

See Also:

Oracle Database Data Warehousing Guide for details

2.2.1.7 Native Text Support

Native support for text mining in Oracle Data Mining has been added in this release.

This change embeds some text processing in Oracle Data Mining, enabling simpler and more performant deployment.

See Also:

Oracle Data Mining User's Guide for details

2.2.1.8 On-the-Fly Models

On-the-fly models (called predictive queries in the Oracle Data Miner GUI workflow `SQLDEV` extension) are transient data mining models that are formed as part of analytic clauses. They represent a simpler form of mining which is tightly integrated with the

SQL language and engine. Moreover, they introduce the concept of partitioned models without the overhead of the persistence of many models.

Applications need to build models per partitioned segment, and this approach addresses that with a transient model.

See Also:

Oracle Data Mining User's Guide for details

2.2.1.9 Prediction Details and Cluster Functions

Prediction detail support for Decision Tree algorithms has been added in this release. Also added, cluster distance and details functions.

Applications require details explaining the reasons behind a prediction to be provided. In addition, certain applications need to find the record closest to the cluster center and the `CLUSTER_PROBABILITY` function is not capable of providing that information.

See Also:

Oracle Data Mining User's Guide for details

2.2.2 Oracle OLAP

The following sections describe new Oracle OLAP features.

2.2.2.1 Cube Query Performance Enhancements

Enhancements have been made in this release to improve query performance against OLAP cubes. These enhancements are designed to minimize CPU and memory consumption and reduce I/O for queries against cubes.

Oracle cubes allow SQL users to transparently access advanced analytic calculations. Recent hardware advances, in particular in Oracle Exadata machine, present numerous opportunities for cube query performance enhancements. This feature leverages those hardware improvements by fully and appropriately utilizing available hardware.

See Also:

Oracle OLAP User's Guide for details

2.2.2.2 Cube Statistics Support

This feature exposes statistics of the Oracle cube in Oracle statistics and workload repertories including Automatic Workload Repository (AWR), Active Session History (ASH), and Automatic Database Diagnostic Monitor (ADDM).

This feature simplifies the administration of Oracle instances that include Oracle cubes and OLAP dimensions.

See Also:

Oracle Database Reference for details

2.2.3 Partitioning Enhancements

The following sections describe partitioning features.

2.2.3.1 Asynchronous Global Index Maintenance for DROP and TRUNCATE Partition

Global index maintenance is decoupled from the DROP and TRUNCATE partition maintenance operation without rendering a global index unusable. Index maintenance is done asynchronously and can be delayed to a later point-in-time.

Delaying the global index maintenance to off-peak times without impacting the index availability makes DROP and TRUNCATE partition and subpartition maintenance operations faster and less resource intensive at the point-in-time of the partition maintenance operation.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.2.3.2 Cascade Functionality for TRUNCATE and EXCHANGE Partition

TRUNCATE and EXCHANGE partition operations provide cascading functionality for reference-partitioned tables, enabling the inheritance of the partition maintenance operation from the parent to the child tables.

Cascading data maintenance operations for TRUNCATE and EXCHANGE partition significantly simplifies application development and provides atomic enforcement of logical data consistency.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.2.3.3 Interval Reference Partitioning

A reference-partitioned table leverages interval partitioning as the top partitioning strategy.

Interval reference partitioning enhances Oracle's partitioning capabilities to model the database scheme according to real business needs.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.2.3.4 ONLINE Move Partition

ALTER TABLE ... MOVE PARTITION becomes non-blocking online DDL while DML operations continue to run uninterrupted on the partition that is being moved. Global indexes are maintained during the move partition, so a manual index rebuild is no longer required.

The online partitioning movement removes the read-only state for the actual MOVE PARTITION command.

See Also:

Oracle Database SQL Language Reference for details

2.2.3.5 Partial Indexes for Partitioned Tables

Local and global indexes can be created on a subset of the partitions of a table.

Partial indexes provide more flexibility in index creation for partitioned tables. For example, index segments can be omitted for the most recent partitions to ensure maximum data ingest rates without impacting the overall data model and access for the partitioned object.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.2.3.6 Partition Maintenance Operations on Multiple Partitions

Partition maintenance operations can be performed on multiple partitions as part of a single partition maintenance operation.

A single partition maintenance operation working on multiple partitions at the same time simplifies application development and leads to more efficient partition maintenance using less system resources.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.2.4 Performance With Zero Effort

The following sections describe performance features with zero effort.

2.2.4.1 Adaptive Query Optimization

It is possible for the optimizer to miscalculate some estimations during initial plan generation. Adaptive query optimization allows these miscalculations to be corrected in one of the following two ways:

- With adaptive plans, a plan can be stopped during execution and reoptimized based on information collected during the initial part of the execution. For example, if the initial plan choice was to do a `NESTED LOOP` with an estimated cardinality of 1, that plan is stopped after 1,000 records have been sent to the join and restarted using a `HASH JOIN` instead because the initial cardinality estimate was wrong.
- Automatic reoptimization does not affect the initial execution of a statement. Instead, the initial execution of a query is monitored and, if the actual execution statistics vary significantly from the original plan estimates, the execution statistics are recorded and used the next time the statement is executed to see if a new plan is chosen for subsequent execution.

Adaptive query optimization is a set of capabilities that enable the optimizer to make run-time adjustments to execution plans and discover additional information that can lead to better statistics.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.2 Adaptive SQL Plan Management

With adaptive SQL plan management, DBAs no longer have to manually run the verification or evolve process for non-accepted plans. When automatic SQL tuning is in `COMPREHENSIVE` mode, it runs a verification or evolve process for all SQL statements that have non-accepted plans during the nightly maintenance window. If the non-accepted plan performs better than the existing accepted plan (or plans) in the SQL plan baseline, then the plan is automatically accepted and becomes usable by the optimizer. After the verification is complete, a persistent report is generated detailing how the non-accepted plan performs compared to the accepted plan performance.

Because the evolve process is now an `AUTOTASK`, DBAs can also schedule their own evolve job at end time.

Unaccepted plans in a SQL plan baseline are automatically evolved during the nightly maintenance window and a persistent verification report is generated which means a DBA no longer has to manual evolve plans and they can go back days or weeks later and review what plans were evolved during each of the nightly maintenance windows.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.3 Automatic Column Group Detection

Extended statistics enables the gathering of statistics on a group of columns within a table as a whole providing the optimizer with more information about any correlation that may exist between the columns. With the introduction of automatic column group detection, DBAs no longer need to know which columns from each table are used together as a workload.

Oracle automatically determines which column groups are required for a table based on a given workload. By monitoring a workload, the necessary column groups are recorded and can be created by executing the `DBMS_STATS.CREATE_EXTENDED_STATS` procedure.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.4 Concurrent Execution of UNION and UNION ALL Branches

The `UNION` and `UNION ALL` statements can run several branches concurrently instead of working on them one-by-one.

Executing multiple branches of `UNION` and `UNION ALL` statements in parallel speeds up processing time and leads to better resource utilization.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.2.4.5 Concurrent Statistics Gathering

Concurrent statistics gathering enables users to gather statistics on multiple tables in a schema (or database) and multiple partitions (or subpartitions) within a table concurrently. Oracle employs Oracle Job Scheduler and Advanced Queuing components to create and manage multiple statistics gathering jobs concurrently.

If you call the `DBMS_STATS.GATHER_TABLE_STATS` procedure on a partitioned table when `CONCURRENT` is set to `true`, then Oracle creates a separate statistics gathering job for each partition (or subpartition) in the table. Oracle Job Scheduler decides how many of these jobs run concurrently and how many are queued based on available system resources. As the currently running jobs complete, more jobs are dequeued and executed until all partitions (or subpartitions) have had their statistics gathered.

Gathering statistics on multiple tables and partitions (or subpartitions) concurrently can reduce the overall time it takes to gather statistics by enabling Oracle to fully utilize multiprocessor environments.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.6 Database Smart Flash Cache Support for Multiple Flash Devices

This feature allows a database instance to access and combine multiple flash devices for Database Smart Flash Cache without the need for a volume manager.

You no longer need to incur the expense or management overhead of a logical volume manager in order to use multiple flash devices for Database Smart Flash Cache.

See Also:

Oracle Database Administrator's Guide for details

2.2.4.7 Dynamic Statistics

During the compilation of a SQL statement, the optimizer decides whether to use dynamic statistics by considering whether the available statistics are sufficient to generate a good execution plan. If the available statistics are not enough, then dynamic statistics are used. Dynamic statistics are persistent and may be used by other queries. One type of dynamic statistic is the information gathered by dynamic sampling. Traditionally, dynamic sampling would automatically occur only if one or more of the tables in the query did not have statistics. Dynamic sampling gathered basic statistics on these tables before optimizing the statement. Now, the optimizer automatically decides if dynamic statistics are useful for all SQL statements and if dynamic sampling is the right approach. If it is, the optimizer also determines what dynamic sampling level is used. The scope of the statistics gathered by dynamic sampling now includes the `JOIN` and `GROUP BY` clauses.

Dynamic statistics are automatically used when the optimizer deems it necessary and the resulting statistics are persistent in the statistics repository making them available to other queries.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.8 Enhanced Parallel Statement Queuing

Critical statements are enabled to bypass the parallel statement queue to reflect business criticality and to provide more flexibility. Parallel statement queuing provides more comprehensive monitoring information, including historical information.

Enhancing parallel statement queuing provides more flexibility to address business requirements for mission-critical environments.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.2.4.9 Enhancements to Incremental Statistics

Gathering statistics on partitioned tables consists of gathering statistics at both the table level and partition level. Incremental statistics allow Oracle to gather statistics only at the partition level and accurately calculate the global-level statistics from the partition-level statistics. Incremental statistics have been enhanced to support partition exchange loading. Data loaded into a non-partitioned table can be exchanged with a partition from the table and Oracle automatically and accurate computes the global statistics for the partition table using the statistics from the non-partitioned table and the existing partition-level statistics.

Previously, incremental statistics considered partition-level statistics stale if any DML occurred on the partition. Now, an incremental staleness threshold can be set to allow incremental statistics to use partition statistics even if some DML has occurred.

Incremental statistics gathering on partitioned tables greatly reduces the time and system resources necessary to gather accurate statistics. By supporting partition exchange operations, incremental statistics enables customers to not only load data into a partitioned table using a sub-second partition exchange operation but to have immediate accurate statistics.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.10 Enhancements to System Statistics

System statistics allow the optimizer to account for the hardware on which the database system is running. With the introduction of smart storage, such as Exadata storage, the optimizer needs additional system statistics in order to account for all of the smart storage capabilities.

The introduction of the new system statistics gathering method allows the optimizer to more accurately account for the performance characteristics of smart storage, such as Exadata storage.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.11 Improved Automatic Degree of Parallelism

Automatic degree of parallelism (Auto DOP) has been enhanced to take more database and statement characteristics into account when determining the degree of parallelism (DOP) for an individual statement.

Enhanced Auto DOP enables better overall system utilization and a more generic applicability of Auto DOP for any kind of mainstream application.

See Also:

Oracle Database Reference for details

2.2.4.12 New Types of Optimizer Statistics

Oracle creates histograms on columns that have a data skew to improve cardinality estimates. Two additional types of histograms have been introduced for columns which have more than 254 distinct values to improve the cardinality estimates generated using histograms. A top frequency histogram is created if a small number of distinct values occupy most of the data (greater than 99% of the data). The histogram is created using the small number of extremely popular distinct values. By ignoring the unpopular values, which are statistically insignificant, a better quality histogram for the highly popular values can be produced. Alternatively, a hybrid histogram can be created which combines a height-based histogram and frequency histogram. It is a height-based histogram where frequent values always become the endpoint values and a value never spans more than one bucket. By recording the frequency of each end value, we record the frequency of the frequent values.

A top frequency histogram provides more accurate cardinality estimates for columns that have more than 254 distinct values but contain a small number of extremely popular distinct values (greater than 99% of the data has one of those values).

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.13 Online Statistics Gathering for Bulk Loads

With online statistics gathering, statistics are automatically created as part of a bulk load operation such as a `CREATE TABLE AS SELECT` operation or an `INSERT INTO ... SELECT` operation on an empty table. Online statistic gathering eliminates the necessity to manually gather statistics after a bulk data load has occurred. It behaves in a similar manner to the statistics gathering done during a `CREATE INDEX` or `REBUILD INDEX` command.

Online statistics gathering improves both performance and manageability of bulk load operations by eliminating user intervention to gather statistics after the load and by removing an additional full table scan required for separate statistics gathering operations.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.14 Out-of-Place Materialized View Refresh

The materialized view (MV) refresh uses logically identical outside tables (and indexes) to perform the MV refresh operations and replaces the stale MVs with the up-to-date outside tables at the end of the refresh process. Out-of-place refresh supports all existing types of refresh methods including `COMPLETE, FAST,` and `PCT` refreshes under the non-atomic refresh mode.

Out-of-place refresh provides minimal refresh time and high availability for materialized views.

See Also:

Oracle Database Data Warehousing Guide for details

2.2.4.15 Session-Private Statistics for Global Temporary Tables

Traditionally, global temporary tables had only one set of statistics that were shared among all sessions even though the table could contain different data in different sessions. In Oracle Database 12*c* Release 1 (12.1), global temporary tables now have session-private statistics. That is a different set of statistics for each session. Queries issued against the global temporary table use the statistics from their own session.

Session-private statistics for global temporary tables improves the performance and manageability of temporary tables. Users no longer need to manually set statistics for the global temporary table on a per session basis or rely on dynamic sampling. This reduces the possibility of errors in the cardinality estimates for global temporary tables and ensures that the optimizer has the data to identify optimal execution plans.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.16 SQL Plan Directives

Besides the statistics gathered using the PL/SQL `DBMS_STATS` package, the optimizer can also gather statistics during compilation using dynamic sampling and during execution time using adaptive execution plans. In previous releases, the compilation and execution statistics were only stored in the cursor cache and were not persistent. With the introduction of SQL plan directives, the compilation and execution statistics are persisted on disk in the `SYSAUX` tablespace. SQL plan directives allow the optimizer access to a larger amount of information regarding the objects being accessed when it generates an execution plan. This information may be that dynamic sampling should be used if tables `t1` and `t2` are joined in a SQL statement or if a correlation is suspected between columns.

SQL plan directives improve execution plan accuracy by persisting both compilation (dynamic sampling results) and execution statistics (adaptive execution plan findings) in the `SYSAUX` tablespace, allowing them to be used by multiple SQL statements.

See Also:

Oracle Database SQL Tuning Guide for details

2.2.4.17 Synchronous Materialized View Refresh

Materialized views can be refreshed simultaneously with its base tables by leveraging partitioning and the logical dependencies between tables and the corresponding materialized views.

The amount of time a materialized view is stale (meaning its data is not up-to-date) is minimized, increasing its availability.

See Also:

Oracle Database Data Warehousing Guide for details

2.3 Compression and Archiving

The following sections describe the new compression and archiving features for Oracle Database 12*c* Release 1 (12.1).

2.3.1 Archiving

The following sections describe archiving features.

2.3.1.1 Database Hardening: Enable Flashback Data Archive (FDA) for Security-Related Application Tables

This features extends the Flashback Data Archive (FDA) feature to provide full history on security sensitive tables of an application. This feature provides a single command to enable FDA on all the designated tables for an application and addresses the need for strong auditing for these tables. This feature also allows the administrator to make all security tables in an application read-only with a single command.

Database hardening makes it easy to track history for all security-related tables in an application and to make those tables read-only as needed, without the need for writing scripts to loop through all the tables or other manual operations. Extending Flashback Data Archive support for tables grouped together by application makes it easy to track all the changes made to those tables and to access the history using Oracle Flashback Query.

See Also:

Oracle Database Development Guide for details

2.3.1.2 Flashback Data Archive Improvements

Several improvements have been made to Flashback Data Archive (FDA). They are:

- User-context tracking

  The metadata information for tracking transactions including the user context is now tracked making it easier to determine which user made which changes to a table.

- Hybrid Columnar Compression (HCC)

  FDA can now be fully utilized on HCC compressed tables on Exadata and other Oracle storage platforms.

- Import and export of history

  Support for importing user-generated history into FDA tables has been added. Customers who have been maintaining history using some other mechanism, such as triggers, can now import that history into FDA.

See Also:

Oracle Database Development Guide for details

2.3.2 General

The following section describes a new Flashback Data Archive feature.

2.3.2.1 Optimization for Flashback Data Archive History Tables

When using Flashback Data Archive to track changes on tables, you can now enable optimization of the corresponding history tables using the `OPTIMIZE DATA` clause when creating a Flashback Data Archive.

Optimization of Flashback Data Archive history tables provide better storage efficiency and better performance for flashback queries on the change history without additional intervention needed by the DBA.

2.3.3 Information Lifecycle Management

The following sections describe Information Lifecycle Management (ILM) features.

2.3.3.1 Automatic Data Optimization (ADO)

This feature provides declarative syntax for specifying Information Lifecycle Management (ILM) policies at the row, segment, and table level.

Database administrators can use this feature to automate the movement of data between different tiers of storage and between different levels of compression. This capability depends on the Heat Map feature which tracks access at the row level (aggregated to block-level statistics) and at the segment level.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.3.3.2 EXECUTE_ILM Procedure

This feature provides a PL/SQL procedure to enforce ADO policies immediately or after a short time delay.

It is sometimes necessary to move data as quickly as possible from one tier to another, or from one compression level to another. The `EXECUTE_ILM` procedure provides the ability to do so, regardless of any previously scheduled ADO policies.

> **See Also:**
>
> *Oracle Database VLDB and Partitioning Guide* for details

2.3.3.3 Heat Map

The Heat Map tracks modifications for individual rows (aggregated to the block level) and modifications and queries at the partition or table level.

Users can implement automated policy-driven data movement and data compression based on the information tracked in the Heat Map using the new ADO feature or using their own tools and scripts.

> **See Also:**
>
> *Oracle Database VLDB and Partitioning Guide* for details

2.3.3.4 PL/SQL Interface for Managing ADO Policies

This feature provides a PL/SQL interface for managing ADO policies, including functions such as scheduling, priority and resource management.

Some customers need to implement complex Information Lifecycle Management (ILM) scenarios, by controlling when their ADO policies are actively moving data, and how much system resources are consumed with data movement operations. This feature provides the ability to manage ILM activities so that they do not negatively impact important production workloads.

> **See Also:**
>
> *Oracle Database VLDB and Partitioning Guide* for details

2.3.3.5 Row-Level Compression Tiering

This feature provides the ability to specify ADO policies to implement compression at the row level within each table in a database. Compression is implemented when all rows in a database block qualify based on the policy being evaluated.

In combination with automatic segment-level compression tiering, this feature provides database administrators with fine-grained control over how the data in their database is stored and managed.

> **See Also:**
>
> *Oracle Database VLDB and Partitioning Guide* for details

2.3.3.6 Segment-Level Compression Tiering

This feature provides the ability to specify ADO policies to implement compression at the segment level within each table in a database.

In combination with automatic row-level compression tiering, this feature provides database administrators with fine-grained control over how the data in their database is stored and managed.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.3.3.7 In-Database Archiving

In-Database Archiving allows users and applications to set the archive state for individual rows. Rows that have been marked as archived are not visible unless the session is enabled to see archived data.

With In-Database Archiving, more data can be stored in production databases for a longer period of time without compromising application performance. In addition, archived data can be aggressively compressed to help improve query and backup performance. Updates to archived data can be deferred during application upgrades, greatly improving the performance of upgrades.

See Also:

Oracle Database VLDB and Partitioning Guide for details

2.3.4 SecureFiles Enhancements

The following sections describe SecureFiles enhancements.

2.3.4.1 Enable PDML Operations on SecureFiles

Limitations have been removed in this release with regard to Parallel DML (PDML) support for SecureFiles LOBs.

This feature allows SecureFiles to leverage the performance and scalability benefits of the PDML features of Oracle Database.

See Also:

Oracle Database SecureFiles and Large Objects Developer's Guide for details

2.3.4.2 Oracle Data Pump: Support SecureFiles LOB as Default

The `impdp` command line has a new parameter (and the PL/SQL `DBMS_DATAPUMP` package has a new option) that tells Oracle Data Pump to create all LOBs as SecureFiles LOBs. By default, beginning with Oracle Database 12*c* Release 1 (12.1), all LOB columns are created as SecureFiles LOBs. However, Oracle Data Pump re-creates tables exactly as they existed in the exported database, so if a LOB column was a BasicFile LOB in the exported database, Oracle Data Pump attempts to re-create it as a BasicFile LOB in the imported database.

This feature allows the user to force creation of LOBs as SecureFiles LOBs and to migrate to the latest more performant feature.

See Also:

Oracle Database Utilities for details

2.3.4.3 SecureFiles is the Default for LOB Storage

In this release, SecureFiles is now the default for LOB storage when the compatible initialization parameter is set to 12.1 or higher.

The SecureFiles feature provides optimal performance for storing unstructured data in the database. Making SecureFiles the default for unstructured data helps ensure that the database is delivering the best performance possible when managing unstructured data.

See Also:

Oracle Database SecureFiles and Large Objects Developer's Guide for details

2.4 Database Overall

The following sections describe the new database features for Oracle Database 12*c* Release 1 (12.1).

2.4.1 Database Consolidation

The following sections describe database consolidation features.

2.4.1.1 Integrate With Operating System Processor Groups

This feature allows the DBA to specify a parameter, `PROCESSOR_GROUP_NAME`, to bind the database instance to a named subset of the CPUs on the server. On Linux, the named subset of CPUs can be created using a Linux feature called control groups (cgroups). On Solaris, the named subset of CPUs can be created using a Solaris feature called resource pools.

This feature is primarily useful for consolidation. When consolidating on a large server, you may want to restrict the database to a specific subset of the CPU and memory. This feature makes it easy to enable CPU and memory restrictions for an Oracle Database instance.

See Also:

Oracle Database Reference for details

2.4.1.2 Oracle Data Pump Support for Database Consolidation: Full Transportable

Full transportable operations include:

- Full transportable support for multitenant container databases (CDBs):

  The new Oracle Data Pump full transportable feature lets you move an entire database from one Oracle Database occurrence to another. You can use this functionality to move a non-CDB (Oracle Database 11*g* Release 2 (11.2.0.3) and up) into a pluggable database (PDB).

  You can use this full transportable feature to move a PDB (Oracle Database 12*c* Release 1 (12.1) and up) into another PDB. You might want to do this if you are moving across versions or to another operating system or hardware platform.

- Full transportable support for non-CDBs:

  The new Oracle Data Pump full transportable feature lets you move an entire database from one Oracle Database instance to another.

  You can use this functionality to move a non-CDB (Oracle Database 11*g* Release 2 (11.2.0.3) and up) into another non-CDB. You can then transport a non-CDB into a CDB at a later date. You can also use the full transportable feature to move a PDB into a non-CDB.

Full transportable operations can reduce the export time and especially, the import time, because table data does not need to be unloaded and reloaded and index structures in user tablespaces do not need to be re-created. Full transportable is more automated than transportable tablespaces because it moves the metadata and user data that resides in non-transportable tablespaces than would previously have been moved in multiple operations. This makes the full transportable feature useful for efficiently moving a database to a new computer system or upgrading to a new release of Oracle Database.

See Also:

Oracle Database Administrator's Guide for details

2.4.1.3 Multitenant Architecture

The multitenant architecture is new in Oracle Database 12*c* Release 1 (12.1). You can have many PDBs inside a single Oracle Database occurrence. PDBs are fully backwards compatible with an ordinary pre-12.1 database.

The benefits of PDBs are:

- Fast provisioning of a new database or of a copy of an existing database.
- Fast redeployment, by unplug and plug, of an existing database to a new platform.
- Quickly patch or upgrade the Oracle Database version for many databases and for the cost of doing it once.
- Patch or upgrade by unplugging a PDB and plugging it into a different CDB in a later version.
- A machine can run more database instances in the form of PDBs than as individual, monolithic databases.

See Also:

Oracle Database Administrator's Guide for details

2.4.1.4 PDBs Backup and Recovery

A CDB consists of zero or more PDBs. Recovery Manager (RMAN) can backup the entire CDB and single or multiple PDBs to a consistent point-in-time. In addition, individual tablespaces or data files can be backed up from specific PDBs.

New syntax, `PLUGGABLE DATABASE`, is introduced to support individual pluggable database backup and recovery.

CDB users need backup and recovery facilities for the new pluggable database model.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.4.1.5 PDBs Point-in-Time Recovery

You can now recover a PDB to a specific point-in-time.

This feature is a high availability enhancement for consolidation and extends point-in-time recovery functionality to PDBs.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.4.1.6 PDBs Resource Plans

Oracle Resource Manager can manage resources on the CDB level and on the PDB level. You can create a CDB resource plan that allocates resources to the entire CDB and to individual PDBs. You can allocate more resources to some PDBs and less to others, or you can specify that all PDBs share resources equally.

With the advent of the multitenant architecture, allowing for the consolidation of multiple separate databases into a single database, there is a need for resource plan functionality to allow the CDB administrator to control the resources that each database in a container database consumes.

See Also:

Oracle Database Administrator's Guide for details

2.4.2 Grid Scheduler

The following sections describe the integration of the Database Scheduler with Oracle Enterprise Manager to create a new, enterprise class Grid Scheduler.

2.4.2.1 New Job Types

This feature provides out-of-the-box support for Oracle Recovery Manager (RMAN) scripts, shell scripts and SQL scripts. Currently there is a lot of setup required for these types of jobs and it can be error prone. With this feature, the user simply specifies the desired job type in the job definition.

This feature provides ease-of-use and reduces the complexity of creating jobs.

See Also:

Oracle Database Administrator's Guide for details

2.4.3 Overall

The following section describes cloning a database.

2.4.3.1 Cloning a Database

`CLONEDB` provides a way to easily and quickly create copies of databases on network attached storage (NAS), using a thin provisioning approach integrated with Oracle Database.

Cloning a production database is a common technique used to help develop and test changes to applications and their surrounding environments. Before a new operating system release, storage software, or application version is installed in a production environment, thorough testing is needed using production data. This is usually accomplished by copying the production database to a test environment. In addition, to the test environment, copies of the production database are also made to the development environments where application developers are creating or modifying applications and testing them. All of these copies require large amounts of storage to be allocated and managed. Using thin provisioning, `CLONEDB` greatly reduces the amount of storage needed for clones of production databases.

See Also:

Oracle Database Administrator's Guide for details

2.4.4 Utilities

The following sections describe new features for Oracle Database Utilities.

2.4.4.1 LOGTIME Parameter for Oracle Data Pump Command

The new `LOGTIME` command-line parameter available in Oracle Data Pump Export and Import allows you to request that messages displayed during export and import operations be timestamped. The valid values are:

- `NONE` - no timestamps on status or log file messages (same as the default)
- `STATUS` - timestamps on status messages only
- `LOGFILE` - timestamps on log file messages only
- `ALL` - timestamps on both status and log file messages

There is also a new option for the `DBMS_DATAPUMP.SET_PARAMETER` procedure called `LOGTIME` and the valid values are the same.

You can use the timestamps to know the elapsed time between different parts of an Oracle Data Pump operation, which can be helpful in diagnosing performance problems and in estimating the timing of similar operations in the future.

See Also:

Oracle Database Utilities for details

2.4.4.2 Oracle Data Pump Audit Commands

Oracle Data Pump commands can now be audited. This provides more complete auditing of operations performed against the database.

See Also:

Oracle Database Security Guide for details

2.4.4.3 Oracle Data Pump Change Table Compression at Import Time

There is a new `impdp` command-line option for Data Pump Import (as well as a new option for the PL/SQL `DBMS_DATAPUMP` package) that allows a user to change the compression options for a table.

This is useful when migrating to an Exadata machine where more compression options for tables are supported which provides better database performance.

See Also:

Oracle Database Utilities for details

2.4.4.4 Oracle Data Pump Enhanced Compression Options

There is a new `expdp` command-line option for Oracle Data Pump Export to control the degree of compression used for a Oracle Data Pump dump file. It also adds the same options to the PL/SQL `DBMS_DATAPUMP` package. This allows the DBA to trade off time spent compressing data against the size of the Oracle Data Pump dump file.

This feature allows the DBA to control the resources used during an export operation.

See Also:

Oracle Database Utilities for details

2.4.4.5 Oracle Data Pump Export View As a Table

There is a new expdp command-line option for Oracle Data Pump Export that allows the user to indicate that a view should be exported as a table. This means that, instead of exporting the view definition, Oracle Data Pump exports a table definition and then unloads all data from the view. At import time, Oracle Data Pump creates a table using the table definition in the dump file and then inserts the data unloaded from the view into the table. The PL/SQL `DBMS_DATAPUMP` package has a similar option.

This feature allows greater flexibility in what a DBA can export. A view gives the DBA greater capability than the current `WHERE` parameter to specify a subset of the database to be unloaded. In a network mode import, exporting the contents of a view can achieve much better performance than using the `impdp QUERY` option.

> **See Also:**
>
> *Oracle Database Utilities* for details

2.4.4.6 Oracle Data Pump No Logging Option for Import

The new `TRANSFORM` option, `DISABLE_ARCHIVE_LOGGING`, to the `impdp` command line causes Oracle Data Pump to disable redo logging when loading data into tables and when creating indexes. It also adds the same option as part of the PL/SQL `DBMS_DATAPUMP` package. With redo logging disabled, the disk space required for redo logs during an Oracle Data Pump import is smaller. However, to ensure recovery from media failure, the DBA should do an RMAN backup after the import completes.

Even with this parameter specified, there is still redo logging for other operations of Oracle Data Pump. This includes all `CREATE` and `ALTER` statements, except `CREATE INDEX`, and all operations against the master table used by Oracle Data Pump during the import.

This feature reduces the required maintenance of redo logs by DBAs.

> **See Also:**
>
> *Oracle Database Utilities* for details

2.4.4.7 Oracle Data Pump Security: No-Echo of Encryption Passwords on Export and Import Commands

This new option adds a parameter, `ENCRYPTION_PWD_PROMPT = [Y | N]`, to the expdp and impdp command line that allows the user to indicate whether the Oracle Data Pump client should prompt for passwords or whether it should retrieve the value from the command line.

This improves security by reducing the possibility of a password being exposed to operating system commands, and by making it unnecessary to include database passwords in operating system scripts.

> **See Also:**
>
> *Oracle Database Utilities* for details

2.4.4.8 SQL*Loader and External Tables: Use dNFS for Accessing Files on NFS Servers

Both external tables and SQL*Loader can be used to load files stored on Network File Storage (NFS) servers. Oracle Direct NFS (dNFS) is an internal I/O layer that provides faster access to large NFS files than traditional NFS clients. Both SQL*Loader and external tables automatically use the new package for large files. However, there is a command-line parameter for SQL*Loader and an access parameter for external tables that can be used to disable the use of dNFS.

See Also:

Oracle Database Utilities for details

2.4.4.9 SQL*Loader Audit Direct Path Loads

This new feature adds auditing capability for direct path loads to the database. This new capability provides complete auditing control for direct path load operations.

See Also:

Oracle Database Security Guide for details

2.4.4.10 SQL*Loader Express Mode

SQL*Loader has a new option that does not require the user to create a SQL*Loader control file. Instead, command-line parameters are used to specify how the data file is loaded, and SQL*Loader automatically chooses the best method with which to load the data. Data files formatted as comma-separated values (CSV) to both SQL*Loader and external tables are now supported.

In addition, there are new default options for SQL*Loader and external tables that help to minimize redundant specification of options in SQL*Loader control files and in external table access parameters. Instead of specifying the same option for every field, the user can specify that option once and have it apply to all fields.

Creating SQL*Loader control files can be complicated. SQL*Loader automatically generates the control file, and outputs a copy of the generated control file for reference or for future reuse. Eliminating the need for SQL*Loader control files for common data file formats such as CSV files makes it much easier and faster for customers to load data.

The new default options for SQL*Loader and external tables reduce the time and complexity of creating SQL*Loader control files or `ORACLE_LOADER` access parameter lists for those customers that need to do so.

See Also:

Oracle Database Utilities for details

2.5 High Availability

The following sections describe the new high availability features for Oracle Database 12*c* Release 1 (12.1).

2.5.1 Application Continuity

The following sections describe Application Continuity and Transaction Guard features.

2.5.1.1 Application Continuity for Java

Application developers were required to deal explicitly with outages of the underlying software, hardware, and communications layers if they wanted to mask outages from end users.

Since Oracle Database 10*g*, Fast Application Notification (FAN) delivered exception conditions to applications fast. However, neither FAN nor earlier Oracle technology reported the outcome of the last transaction to the application or recovered the in-progress request from an application perspective. As a result, outages were exposed leading to user inconvenience and lost revenue. Users could unintentionally make duplicate purchases and submit multiple payments for the same invoice. In the problematic cases, the administrator needed to reboot the mid-tier to deal with the incoming problems this caused.

Application Continuity is an application-independent feature that attempts to recover incomplete requests from an application perspective and masks many system, communication, hardware failures, and storage outages from the end user.

The protocol ensures that end user transactions are executed no more than once. When successful, the only time that an end user should see an interruption in service is when there is no point in continuing. When replayed, the execution appears to the application and client as if the request was slightly delayed. The effect is similar to a loaded system where the database runs the request slightly slower so that the response to the client is delayed.

Most failures should be masked. This results in fewer calls to the application's error handling logic. For example, less often, the application raises an error leaving the user not knowing what happened or forces the user to reenter data. Or, more problematic, the administrators must restart the mid-tier servers to cope with the failure.

Other benefits include:

- Improved end user experience.
- Higher application availability.
- Improved application developer productivity.

See Also:

Oracle Database Development Guide for details

2.5.1.2 Transaction Guard

Transaction Guard provides a generic tool for applications to use for at-most-once execution in case of planned and unplanned outages and repeated submissions. Applications use a new concept called the logical transaction ID (LTXID) to determine the outcome of the last transaction open in a database session following an outage. Without using Transaction Guard, applications that attempt to retry operations following outages can cause logical corruption by committing duplicate transactions.

One of the fundamental problems for recovering applications after an outage is that the commit message that is sent back to the client is not durable. If there is a break between the client and the server, the client sees an error message indicating that the communication failed. This error does not inform the application whether the submission executed any commit operations or if a procedural call ran to completion executing all expected commits and session state changes or if it failed part way through or, more problematic, is still running disconnected from the client.

Failing to recognize that the last submission has committed or will commit sometime soon or has not run to completion can lead applications, that attempt to replay, to cause duplicate transaction submissions because the software might try to reissue already persisted changes.

Without Transaction Guard, if a transaction has been started and commit has been issued, the commit message that is sent back to the client is not durable. The client is left not knowing whether the transaction committed or not. The transaction cannot be resubmitted if the non-transactional state is incorrect or if it already committed. In the absence of knowing the commit and completion information, resubmission can lead to transactions being applied more than once and in the incorrect state.

The benefits of Transaction Guard are:

- First RDBMS to preserve commit outcome.
- Known outcome for every transaction.
- A tool for at-most-once transaction execution.

See Also:

Oracle Database Development Guide for details

2.5.2 Logical Replication

The following sections describe creation of the replication solution based on Oracle GoldenGate and SQL Apply.

2.5.2.1 Support for Extended VARCHAR2 in XStream

XStream provides native support for the extended VARCHAR2 data type. XStream can capture or apply changes to tables that include the extended VARCHAR2 data type.

See Also:

Oracle Database XStream Guide for details

2.5.2.2 XStream New Apply Parameters

Additional parameters are available in this release for XStream inbound servers that control the behavior of the apply processes. New parameters include COMPUTE_LCR_ON_ARRIVAL and OPTIMIZE_PROGRESS_TABLE. The COMPUTE_LCR_ON_ARRIVAL parameter controls when scheduling dependencies are calculated for XStream apply processes. The OPTIMIZE_PROGRESS_TABLE parameter minimizes the apply progress table maintenance by using the local redo log to construct the apply progress table.

These new parameters give the database administrator more control for performance tuning of XStream inbound servers.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

2.5.2.3 XStream New Capture Parameters

In this release, additional parameters are available in XStream that control the behavior of the capture process. The EXCLUDETAG parameter is used in combination with the GETAPPLOPS and GETREPLICATES capture parameters to control the capture of changes from the redo log files with specific redo tag values.

XStream provides additional filtering control on changes that can be captured from the redo log files.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

2.5.2.4 XStream Optimized Management of Large Transactions

As a performance optimization, XStream inbound servers can process large transactions before the transaction COMMIT is received from the source. The EAGER_SIZE apply parameter controls the minimum size at which this optimization begins. Large transactions may require an additional server to apply the changes.

The MAX_PARALLELISM apply parameter controls the maximum number of apply servers that can be used for the apply process.

Large transactions can now begin to apply as soon as the source changes are received at the target database. This may reduce the replication latency of large transactions.

See Also:

Oracle Database XStream Guide for details

2.5.2.5 XStream Support for Advanced LOB Deduplication

XStream supports changes made to SecureFiles LOB columns stored using deduplication. Databases using logical replication can now take advantage of Advanced LOB Deduplication potentially improving performance and reducing storage space.

See Also:

Oracle Database XStream Guide for details

2.5.2.6 XStream Support for XML Object Relational and Binary

XStream data type support is extended to include XMLType data stored object relationally or as binary. Support is provided for both XStream outbound and inbound servers. XStream capture or apply of DML changes to tables is supported with any XMLType data storage type.

See Also:

Oracle Call Interface Programmer's Guide and *Oracle Database XStream Guide* for details

2.5.3 Global Data Services

The following sections describe features that provide load balancing similar to Oracle load balancing and failover for distributed environments of Oracle RAC and single-instance databases that are interconnected using Oracle Active Data Guard and Oracle GoldenGate.

2.5.3.1 Global Data Services (GDS)

Global Data Services (GDS) is a new capability of Oracle Database that extends the concept of services, which are only available in Oracle RAC, to a globally replicated configuration involving a combination of Oracle RAC, Active Data Guard, and Oracle GoldenGate. This allows services to be deployed anywhere within this globally replicated configuration, supporting load balancing, high availability, database affinity, and so on.

Customers who have utilized the concept of services for Oracle RAC can now extend the same benefits of automatic workload management to their Active Data Guard or Oracle GoldenGate configurations. Similarly, single-instance Active Data Guard or Oracle GoldenGate customers can now fully utilize the benefits of services and automatic workload management for their replicated configurations.

See Also:

Oracle Database Global Data Services Concepts and Administration Guide for details

2.5.3.2 Oracle C/C++ Application High Availability Enhancements

Enhancements to Oracle Call Interface (OCI) high availability infrastructure include:

- Transparent Application Failover (TAF) support for foreground process failure.
- Support for intelligent reconnect including restoration of connection objects (for example, module, action, execution context identifier (ECID), logical transaction identifier (LTXID), and client ID) from old session to recovered session.
- Document Global Data Services (GDS) support.

These features provide high availability services for C and C++ applications connecting to the Oracle 12*c* Database.

See Also:

Oracle Database Development Guide for details

2.5.4 Improved Resiliency

The following section describes improved resiliency for Oracle ASM.

2.5.4.1 Oracle ASM Disk Scrubbing

Oracle ASM disk scrubbing is a new feature that checks logical data corruptions and repairs them automatically in normal and high redundancy disk groups. This feature is designed so that it does not have any impact on normal I/O in production systems. The scrubbing process repairs logical corruptions using the mirror disks. Disk scrubbing leverages the Oracle ASM rebalancing to minimize I/O overhead.

Oracle ASM disk scrubbing improves availability and reliability by proactively reading data that would otherwise not be read. Latent errors or corruption can be discovered and fixed by Oracle ASM disk scrubbing while redundant data is available.

See Also:

Oracle Automatic Storage Management Administrator's Guide for details

2.5.5 Online Operations

The following sections describe new features for online operations.

2.5.5.1 Editioned and Noneditioned Objects

In Oracle Database 11*g* Release 2 (11.2), a noneditioned object could not depend upon an editioned object. This rule caused problems when the owner of objects that needed to become editioned owned an object of a noneditionable type that depended on an object of an editionable type. Either the `ALTER USER ENABLE EDITIONS` command failed or, if the `FORCE` keyword was used, violating dependants became invalid. The only workaround was to reestablish such violating dependency parents in a new schema and not to editions-enable its owner.

In Oracle Database 12*c* Release 1 (12.1), the editioned state of an object whose type is editionable is now controlled at the granular level of the individual object.

Also, for materialized views, indexes, and table columns (where violations were caused by references to an editioned view or an editioned PL/SQL function), the dependent phenomenon can now specify the edition in which the editioned object(s) are to be found. It is now easier to prepare an application to use edition-based redefinition.

See Also:

Oracle Database Development Guide for details

2.5.5.2 Enhanced Online DDL Capabilities

Several schema maintenance DDL operations no longer require blocking locks, making these operations non-intrusive and transparent for online use. The improved schema maintenance DDL operations are:

- `DROP INDEX ONLINE`
- `DROP CONSTRAINT ONLINE`
- `SET UNUSED COLUMN ONLINE`
- `ALTER INDEX UNUSABLE ONLINE`
- `ALTER INDEX [VISIBLE | INVISIBLE]`

Removing internal blocking locks enables simpler and more robust application development, especially for application migrations. It avoids application disruptions for many of the typical schema maintenance operations.

See Also:

Oracle Database SQL Language Reference for details

2.5.5.3 Invisible Columns

The property of whether a column is visible can be controlled by the user. Invisible columns are not seen unless specified explicitly in the `SELECT` list. Any generic access of a table (such as a `SELECT * FROM table` or a `DESCRIBE`) does not show invisible columns.

The notion of invisible columns enables easier online application migrations as provided by Oracle's edition-based redefinition.

See Also:

Oracle Database Administrator's Guide for details

2.5.5.4 Lock Timeout for FINISH_REDEF_TABLE

Now you can specify a lock timeout in number of seconds during which time `FINISH_REDEF_TABLE` attempts to acquire an exclusive lock for swapping the source and interim tables and, if timeout expires, the operation exits.

This feature increases the flexibility of `FINISH_REDEF_TABLE` to exit after waiting a user-specified number of seconds so that the user does not wait indefinitely or needs to force exit of the online redefinition session.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

2.5.5.5 Metadata-Only DEFAULT Column Values for NULL Columns

The default values of columns are maintained in the data dictionary for columns specified as `NULL`.

Adding new columns with `DEFAULT` values no longer requires the default value to be stored in all existing records. This not only enables a schema modification in sub-seconds and independent of the existing data volume, it also does not consume any space.

See Also:

Oracle Database Administrator's Guide for details

2.5.5.6 Move a Data File Online

In this release, a data file can now be moved online while it is open and being accessed.

Being able to move a data file online means that many maintenance operations, such as moving data to another storage device or moving databases into Oracle Automatic Storage Management (Oracle ASM), can be performed while users are accessing the system. This ensures that continuity of service and service-level agreements (SLA) on uptime can be met.

See Also:

Oracle Database Administrator's Guide for details

2.5.5.7 Multiple Indexes on Same Set of Columns

Multiple indexes can be created on the same set of columns as long as some characteristic is different. Qualifying characteristics are:

- B-tree versus bitmap
- Different partitioning strategies
- Unique versus non-unique

Providing the capability to create multiple indexes on the same set of columns enables transparent and seamless application migrations without the need to drop an existing index and re-create it with different attributes.

See Also:

Oracle Database Administrator's Guide for details

2.5.5.8 Online Redefinition of Multiple Partitions

Online redefinition supports redefinition of multiple partitions in a single redefinition session. This feature reduces the completion time to redefine multiple partitions while still providing access to the underlying table.

See Also:

Oracle Database Administrator's Guide for details

2.5.5.9 Single Command REDEF_TABLE to Redefine Table or Partition

`REDEF_TABLE` is a new procedure in the `DBMS_REDEFINITION` package which allows a one-step operation to easily redefine a table or partition under the following specific set of conditions:

- Tablespace changes for table or partition, index, and LOB columns.
- Compression type changes for table or partition, index key, and LOB columns.
- `STORE AS SECUREFILE` or `BASICFILE` for LOB columns.

See Also:

Oracle Database Administrator's Guide for details

2.5.5.10 Support Redefinition of Tables With VPD Policies

Online redefinition can redefine tables that have Virtual Private Database (VPD) policies defined on them. This feature eliminates downtime for redefining these tables.

See Also:

Oracle Database Administrator's Guide for details

2.5.6 Oracle Data Guard Enhancements

The following sections describe enhancements to Oracle Data Guard.

2.5.6.1 Advanced Data Guard Broker Manageability

Management of Data Guard configurations has been enhanced to include additional health check monitoring, error reporting, and problem diagnosis and resolution.

This feature provides simpler, more productive, and more reliable management of a Data Guard configuration to reduce management cost and further enhance high availability.

See Also:

Oracle Data Guard Broker for details

2.5.6.2 Oracle Data Guard Broker Support for Cascaded Standby Databases

Oracle Data Guard broker can now be used to manage configurations having cascaded standby databases. This feature provides improved management productivity in Data Guard configurations that include cascaded standby databases.

See Also:

Oracle Data Guard Broker for details

2.5.6.3 Fast Sync

Data Guard maximum availability supports the use of the `NOAFFIRM` redo transport attribute. A standby database returns receipt acknowledgment to its primary database as soon as redo is received in memory. The standby database does not wait for the Remote File Server (RFS) to write to a standby redo log file.

This feature provides increased primary database performance in Data Guard configurations using maximum availability and `SYNC` redo transport. Fast Sync isolates the primary database in a maximum availability configuration from any performance impact due to slow I/O at a standby database.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.6.4 Single Command Role Transitions

This feature implements single DDL commands to execute Data Guard role transitions (switchover and failover) to replace the multiple commands required in previous releases. This provides simpler and faster role transitions for higher availability.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.6.5 Real-Time Apply is Default Setting for Data Guard

In previous releases, when creating a Data Guard configuration using the SQL command line, the default configuration was to apply redo from archived log files on the standby database. In Oracle Database 12*c* Release 1 (12.1), the default configuration is to use real-time apply so that redo is applied directly from the standby redo log file.

Recovery time is shortened at failover given that there is no backlog of redo waiting to be applied to the standby database if a failover is required. An active Data Guard user also sees more current data. This enhancement eliminates additional manual configuration (and the requirement that the administrator be aware of the default setting) that was required in past releases. It also makes the default SQL*Plus configuration identical to the default configuration used by the Data Guard broker.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.6.6 Resumable Switchover Operations

In previous releases of Oracle Data Guard broker, if an issue was encountered during a switchover operation, there was no graceful way to resolve the issue and resume the switchover operation from where it left off. Oracle Data Guard broker introduces the capability for resumable switchover along with additional flexibility to facilitate switchover operations when things do not go as expected.

Higher availability during planned maintenance is now available. Issues that may be encountered during switchover operations can be resolved and switchover can resume from where it left off to minimize downtime during planned maintenance operations.

See Also:

Oracle Data Guard Broker for details

2.5.6.7 Active Data Guard Enhanced Security

A new view, `RO_USER_ACCOUNT`, has been added to track dynamic user information in an in-memory table of failed login attempts. This information is used to lock access to a user and allow the DBA to reenable the account on the standby database if desired. This feature enhances security of an Active Data Guard standby database.

See Also:

Oracle Database Reference for details

2.5.6.8 Active Data Guard Support for DML on Global Temporary Tables

This feature expands the number of read-only applications that can be off-loaded from production databases to an Active Data Guard standby database. Even though an Active Data Guard standby database is open in read-only mode, reporting applications are now able to write to global temporary tables at the standby database without any modification.

Offloading reports to an Active Data Guard standby improves primary database performance and increases production capacity by utilizing both primary and standby systems at all times. This increases the return on investment in disaster recovery systems.

See Also:

Oracle Database Administrator's Guide for details

2.5.6.9 Active Data Guard Support for Sequences

This feature provides enhanced support of sequences in an Active Data Guard environment, making it easier to offload read-only reporting to an Active Data Guard standby database. Global sequences created by the primary database can now be accessed from standby databases. The assigned sequence numbers are unique across the entire Data Guard configuration.

Additionally, a new type of special sequence, called a session sequence, is provided specifically for use with global temporary tables that have session visibility. A session sequence returns a unique range of sequence numbers only within a session, but not across sessions.

This new feature expands the range of reporting applications that can easily be off-loaded from a primary database to an Active Data Guard standby to include reports that require unique sequence numbers.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.6.10 Active Data Guard Real-Time Cascade

A standby database that cascades redo to other standby databases can transmit redo directly from its standby redo log file as soon as it is received from the primary database. Cascaded standby databases receive redo in real-time. They no longer have to wait for standby redo log files to be archived before redo is transmitted.

This feature ensures that cascaded standby databases are up-to-date with the primary production database. If used for disaster recovery, cascaded standby databases can deliver nearly the same recover point objective as any other standby database. Read-only queries and reports return up-to-date results.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.6.11 Active Data Guard Far Sync

Far Sync is used to extend zero data loss protection to a remote standby database and avoid the impact to primary database performance of WAN network latency. A primary database ships synchronously to a light-weight instance referred to as a far sync instance (a control file and log files, no data files and no media recovery). The far sync instance then forwards the redo asynchronously to a remote standby database that is the failover target. Additional Far Sync features include the ability to directly service up to 29 remote destinations, and the ability to utilize Oracle Advanced Compression to compress redo for efficient transmission across a WAN. Far Sync is transparent to the administrator with regards to Data Guard role transitions. The same switchover or failover command used for any Data Guard configuration transitions any remote standby databases served by a far sync instance to the primary production role.

Zero data loss protection can be achieved across long distances. The far sync instance is located within a distance of the primary database where synchronous transport does not impact application performance. Far Sync handles all communication with remote standby databases and is transparent when executing a zero data loss failover. Far Sync also offloads the production database of the overhead of servicing multiple remote destinations and redo transport compression.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.7 Data Guard Rolling Upgrade Enhancements

The following sections describe rolling upgrades in Oracle Database.

2.5.7.1 Data Guard Rolling Upgrade Support for Advanced Data Types

This release includes additional native redo-based replication for Data Guard SQL Apply to support database rolling upgrades (transient logical standby). Data types include XMLType stored as binary XML, XMLType stored in object-relational format, objects and collections, Database File System (DBFS), XDB, Oracle Spatial and Graph, Oracle Text, Oracle Multimedia, Label Security, and Oracle SecureFiles (deduplication and fragment operations).

Data Guard database rolling upgrades reduce planned downtime by enabling the upgrade to new database releases or patch sets in rolling fashion. Total database downtime for such an upgrade is limited to the small amount of time required to execute a Data Guard switchover.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.7.2 Data Guard Rolling Upgrade Support for XDB Repository

XML DB Repository supports Data Guard rolling upgrades.

XML DB Repository is no longer a restriction for Data Guard rolling upgrades. Oracle Database can be upgraded to new patch sets and database releases in a rolling fashion, minimizing planned downtime for a broader range of customer Oracle Database deployments.

See Also:

Oracle XML DB Developer's Guide for details

2.5.7.3 Disaster Protection During Database Rolling Upgrade

Data protection is maintained during the Oracle Data Guard database rolling upgrade process by enabling the standby database that is the target of the upgrade to continue receiving primary database redo while the standby database is open in upgrade mode.

This reduces management complexity by eliminating the requirement to create and maintain a separate archive log repository to provide the same level of data protection.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.7.4 Oracle Advanced Queuing (AQ) Support for Data Guard Database Rolling Upgrade

Databases that use Oracle Advanced Queuing (AQ) can be upgraded to new Oracle Database releases and patch sets in rolling fashion using Data Guard database rolling upgrades (transient logical standby database only). Rolling upgrades are supported beginning in Oracle Database 12*c* Release 1 (12.1).

Data Guard database rolling upgrades reduce planned downtime by enabling the upgrade to new database releases or patch sets in rolling fashion. Total database downtime for such an upgrade is limited to the small amount of time required to execute a Data Guard switchover.

> **See Also:**
>
> *Oracle Database Advanced Queuing User's Guide* for details

2.5.7.5 Oracle Data Guard Broker Support for Database Rolling Upgrades

Oracle Data Guard broker now supports database rolling upgrades. With this new feature, the Data Guard broker configuration can be preserved so that it does not have to be rebuilt after the upgrade is complete.

> **See Also:**
>
> *Oracle Data Guard Broker* for details

2.5.7.6 Oracle Scheduler Support for Data Guard Database Rolling Upgrade

Oracle Scheduler jobs that are created at a primary database are replicated to a transient logical standby database. This simplifies using Data Guard to upgrade to new Oracle Database releases and patch sets in rolling fashion (transient logical standby database only).

Data Guard database rolling upgrades reduce planned downtime by enabling the upgrade to new database releases or patch sets in rolling fashion. Total database downtime for such an upgrade is limited to the small amount of time required to execute a Data Guard switchover.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for details

2.5.7.7 Rolling Upgrade Using Active Data Guard

Active Data Guard provides several new PL/SQL packages and DDL commands to automate the previous manual steps of performing a database rolling upgrade to a new Oracle patch set, database release, or to perform other planned maintenance. The process starts with a primary and physical standby database at the previous version and ends with both primary and physical standby database at the new version. The automation includes handling the switchover of production to the new version. It also performs extensive validation at every step of the process. If problems are encountered users can choose to either correct the error and resume the upgrade or roll back to the original state of the configuration.

Rolling upgrade using Active Data Guard reduces management effort and improves the reliability of performing database rolling upgrades. Users benefit from lower administrative cost and higher availability by reducing downtime for planned maintenance.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.7.8 SQL Apply Extended Data Type Support (EDS)

Extended data type support (EDS) for SQL Apply enables replication for select data types by SQL Apply without requiring native redo-based replication. EDS supports `SDO_GEOMETRY`, XMLType stored in object-relational format, XMLType stored as binary XML, objects, and objects with varray columns.

EDS enables users to reduce planned downtime and increase availability by using database rolling upgrades even in cases where their database includes data types for which SQL Apply does not yet support redo-based replication. Users are able to seamlessly transition to redo-based replication as SQL Apply adds native support for these data types in subsequent Oracle Database releases.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.7.9 SQL Apply Support for Objects, Collections, and XMLType

SQL Apply supports Oracle objects and collections, XMLType stored as binary XML, and XMLType stored in an object-relational format. This feature provides greater flexibility for SQL Apply to support rolling upgrades for databases with nonscalar data types.

See Also:

Oracle Database Utilities for details

2.5.7.10 SQL Apply Support for XMLType

Replication of XMLType tables and columns for all XMLType storage models using SQL Apply is enabled in this release.

This feature allows XML content to be replicated between database instances in a safe and secure manner using the same proven technology that is already being used to replicate conventional relational data. XML content and relational content can now be replicated simultaneously without needing to perform complex and expensive conversions of the XML components. This feature enables the use of XMLType to manage XML content in situations that require the extreme levels of high availability that customers expect from Oracle Database.

See Also:

Oracle Data Guard Concepts and Administration for details

2.5.7.11 SQL Apply Support for SecureFiles LOBs

This feature provides SQL Apply support for deduplication of SecureFiles LOB columns. SQL Apply can be used for rolling upgrade of an Oracle Database that uses SecureFiles LOBs without any restrictions on their use.

See Also:

Oracle Database Utilities for details

2.5.8 Oracle Database Advanced Queuing Enhancements

The following sections describe Oracle Database Advanced Queuing (AQ) improvements.

2.5.8.1 JMS Bulk Message Purging

Oracle Streams Advanced Queuing (AQ) JMS uses partitioned tables beginning in Oracle Database 12*c* Release 1 (12.1). Oracle Java Message Service (JMS) messages are purged by truncating partitions instead of row-at-a-time deletes. This feature provides increased performance and reduced overhead.

See Also:

Oracle Database Advanced Queuing User's Guide for details

2.5.8.2 JMS Event-Driven Listener

AQ Java Message Service (AQ JMS) listener no longer requires open connections dedicated to listening for new JMS messages. Polling across multiple queues is not required. This feature improves AQ JMS performance and scalability and reduces overhead.

2.5.8.3 JMS Message Priorities, Exception Queues, and Expiration

AQ JMS supports priorities, exception queues, and message expiration differently than in prior releases. This feature improves AQ JMS performance, reduces overhead, and provides better standards compliance.

See Also:

Oracle Database Advanced Queuing User's Guide for details

2.5.8.4 JMS Transactional Nonpersistent Queues

In this release, AQ JMS supports transactional nonpersistent queues instead of emulating them with persistent queues.

This feature provides AQ JMS with better performance, scalability, reduced overhead, and better standards compliance.

See Also:

Oracle Database Advanced Queuing User's Guide for details

2.5.8.5 Message Forwarding on Oracle RAC

Advanced Queuing (AQ) on Oracle RAC is now sharded to avoid unnecessary exchange of blocks between instances. Tunable message forwarding is also supported on Oracle RAC. This feature improves AQ performance and scalability.

See Also:

Oracle Database Advanced Queuing User's Guide for details

2.5.8.6 AQ Rules Engine Fast Evaluation of SYS_CONTEXT and Other Predicates

AQ rules engine has been enhanced to provide faster evaluation of expressions such as `BITAND`, `CEIL`, `FLOOR`, `LENGTH`, `POWER`, `CONCAT`, `LOWER`, `UPPER`, `INSTR`, `SYS_CONTEXT`, and `UID`. This feature improves AQ performance and scalability.

See Also:

Oracle Database Advanced Queuing User's Guide for details

2.5.8.7 Rules Engine Result Cache

The rules engine introduces a result cache to improve the performance of many commonly used rules. The result cache bypasses the evaluation phase if an expression with the same attributes has already been evaluated. This feature provides performance improvement by caching the results of rule evaluations.

See Also:

Oracle Database Advanced Queuing User's Guide for details

2.5.8.8 Sharded Queues for Performance and Scalability

Oracle Streams Advanced Queuing (AQ) now has queue tables that are partitioned. Partitioned tables form part of the foundation to scale and increase performance of AQ, especially on Oracle RAC or Exadata.

See Also:

Oracle Database Advanced Queuing User's Guide for details

2.5.8.9 Simplified Metadata and Schema

AQ now has fewer tables and supports objects to support sharded queues. This feature provides improvement in performance, scalability, and manageability.

See Also:

Oracle Database Advanced Queuing User's Guide for details

2.5.9 RMAN Enhancements

The following sections describe improvements to Recovery Manager (RMAN).

2.5.9.1 Active Database Duplication Enhancements

Active `DUPLICATE` uses a network-enabled restore method that is run on the auxiliary database to clone the source database. This is in contrast to the image copy-based approach that was run on the source database in previous releases. Active `DUPLICATE` supports the `SECTION SIZE` option to divide data files into subsections that are restored in parallel across multiple channels on the auxiliary database. Active `DUPLICATE` supports compression during the restore phase.

The benefits of this new feature include:

- Reduce active duplicate time for databases with large data files by more evenly spreading out the restore workload across multiple channels on the auxiliary database.
- Reduce active duplicate time by more efficiently using network bandwidth using compression during data transfer operations.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.5.9.2 Cross-Platform Backup and Restore

`BACKUP` and `RESTORE` commands feature new options to create a cross-platform compatible backup and to restore the same on a different platform.

This feature reduces operational complexity using cross-platform transportable tablespace and cross-platform transportable database methods to migrate data between platforms.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.5.9.3 DUPLICATE Enhancements

This feature disables automatic opening of a recovered clone database so that you can perform any database setting changes before using the `NOOPEN` option. For example, you may want to modify block change tracking or Flashback Database settings before opening the clone database. This feature is also useful for upgrade scenarios where the database must not be open with `RESETLOGS` prior to running upgrade scripts.

These enhancements provide additional flexibility during `DUPLICATE` and expand its use for upgrade scenarios. For example, the `NOOPEN` option allows `DUPLICATE` to create a new database as part of an upgrade procedure and leaves the database in a state ready for opening in upgrade mode and subsequent execution of upgrade scripts.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.5.9.4 Multisection Image Copies

Image copies can be taken with the `SECTION SIZE` option to divide data files into subsections that can be backed up in parallel across multiple channels. This feature reduces image copy creation time for large data files which is especially beneficial in Exadata environments.

This can also reduce completion time for non-backup use cases. For example, copying a file as part of a transportable tablespace procedure or creating a clone with active duplicate.

See Also:

Oracle Database Backup and Recovery Reference for details

2.5.9.5 Multisection Incremental Backups

Incremental backups can be taken with the `SECTION SIZE` option to divide data files into subsections that can be backed up in parallel across multiple channels. This reduces the incremental backup time for large data files which is especially beneficial in Exadata and cloud environment backups.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.5.9.6 Network-Enabled RESTORE

`RESTORE` operations can copy data files from one database to another (for example, a physical standby database to a primary database) over the network. Network-enabled restore also supports compression and multisection options.

The RESTORE operation reduces data file copy time from one database to another by reducing data transfer sizes and by better parallelizing restore workload for large data files across multiple channels.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.5.9.7 RMAN Command-Line Interface Enhancements

Recovery Manager (RMAN) command-line interface has been enhanced to:

- Run SQL as-is at the command line, no longer requiring the SQL command.
- Support SELECT statements.
- Support the DESCRIBE command on tables and views.

These enhancements provide better ease-of-use when running SQL in an RMAN session.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.5.9.8 Storage Snapshot Optimization

Backup mode can induce additional system and I/O overhead due to writing whole block images into redo, in addition to increasing procedural complexity in large database environments. A third-party storage snapshot that meets the following requirements can be taken without requiring the database to be placed in backup mode:

- Database is crash-consistent at the point of the snapshot.
- Write ordering is preserved for each file within a snapshot.
- Snapshot stores the time at which a snapshot is completed.

A new RECOVER command keyword, SNAPSHOT TIME, is introduced to recover a snapshot to a consistent point, without any additional manual procedures for point-in-time recovery needs.

The overhead consists of logging additional redo and performing a complete database checkpoint.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.5.9.9 Table-Level Recovery From Backups

Recovery Manager (RMAN) can restore and recover a table or set of tables from existing backups on disk or tape with the new RECOVER TABLE option.

This option reduces time and disk space to recover a table or set of tables from an existing backup versus manually restoring and recovering the required tablespaces to a separate disk location, then exporting the desired tables and importing them to the original database.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.6 Manageability

The following sections describe the new manageability features for Oracle Database 12*c* Release 1 (12.1).

2.6.1 Database Performance Tuning

The following sections describe automatic performance management features.

2.6.1.1 Enterprise Manager Database Express

Enterprise Manager Database Express 12*c* is a web-based tool for managing Oracle databases. It is configured out-of-the-box and ships with every database, is extremely light weight, and does not require any special installation such as a JVM or an application server. Enterprise Manager Database Express provides an intuitive and interactive user interface for performing basic database administration tasks, such as database configuration and administration, space administration, users and roles management, and performance management.

Enterprise Manager Database Express greatly simplifies database performance diagnostics by consolidating the relevant database performance screens into a consolidated view called the database Performance Hub. DBAs get a single, consolidated view of the current real-time and historical view of the database performance across multiple dimensions such as database load, monitored SQL and PL/SQL, and Active Session History (ASH) in a single page for the selected time period.

See Also:

Oracle Database 2 Day DBA for details

2.6.1.2 PGA Size Limit

This feature limits the total amount of Program Global Area (PGA) that an instance can allocate, using a parameter called `PGA_AGGREGATE_LIMIT`. When the instance has allocated the `PGA_AGGREGATE_LIMIT` amount of PGA, sessions with the highest amount of allocated PGA are stopped until the limit is complied with.

This feature is important for consolidation because, without a hard limit, the instance can become unstable due to excessive paging. Excessive paging is one of the leading causes of instance eviction in an Oracle RAC database and can cause a multitude of performance and stability problems.

See Also:

Oracle Database Performance Tuning Guide for details

2.6.1.3 Real-Time Database Operations Monitoring

Real-time database operations monitoring allows database administrators to easily monitor and troubleshoot performance problems in long running jobs. This feature helps make long running database operations like a batch job, an ETL (extraction, loading, and transformation) operation, or a scheduler job, transparent so that administrators can see exactly what the operation is doing and at what time. It does this by tracking the SQL and PL/SQL commands that make up a database operation along with their time lines.

As a result, DBAs can easily trace any issues with the database operation to the problem SQL or PL/SQL for more in-depth analysis.

See Also:

Oracle Database SQL Tuning Guide for details

2.6.1.4 Resource Manager Runaway Query Management

Runaway queries are a persistent problem in databases and can adversely impact overall performance if not managed properly. Resource Manager now provides information about and proactively manages offending queries. There are new views that allow a DBA to see the most recent SQL commands that have hit limits in each consumer group. These views are persisted to the Automatic Workload Repository (AWR) to allow post-mortem analysis. Also, there is new functionality to allow the DBA to take preemptive action on offending SQL execution plans.

The net result is that DBAs can now proactively prevent runaway queries, before they do any damage, rather than being reactive to queries which have already consumed too many resources.

See Also:

Oracle Database Administrator's Guide for details

2.6.1.5 Spot ADDM

Oracle Database has many automatic performance diagnostics advisors such as Automatic Database Monitor (ADDM), real-time ADDM, and compare period ADDM to allow DBAs to diagnose and resolve performance problems. While ADDM reports problems found in the last hour (by default), there are times when a critical problem occurs and DBAs need to be notified right away. Spot ADDM is a new advisor that is automatically triggered when a database begins to encounter performance issues and tries to identify the root cause of the problem. Some of the types of problems that trigger spot ADDM include high CPU load or I/O bound scenarios. The results of spot ADDM are persisted in Automatic Workload Repository (AWR) as reports.

Getting visibility into these problems enables DBAs to respond rapidly and prevent cascading problems that could ultimately create catastrophic failures.

See Also:

Oracle Database Performance Tuning Guide for details

2.6.2 Database Testing

The following sections describe database testing features.

2.6.2.1 Mask At Source or Mask During Subset

To mask confidential data for non-production use, enterprises needed to make a copy of the production database and then mask the data before sharing it with non-production users such as testers or developers. Masking at the source or masking while subsetting at the source database allows enterprises to provision a secure and reduced size test system directly from the production database without the need for a full production database copy. Enterprises may choose to execute the masking or subsetting operations or both to provision the test database in a single workflow.

Masking at the source or masking while subsetting at the source ensures that sensitive production data never leaves the source database when provisioning test systems and, therefore, complies with data privacy policies.

See Also:

Oracle Database Testing Guide for details

2.6.2.2 Self Update for Oracle Applications Masking and Subsetting Templates

Enterprises can mask sensitive data using data masking templates for Oracle applications such as Fusion applications and E-Business Suite. Given the complexity of enterprise applications such as Fusion applications or E-Business Suite, the process of manually importing the data masking templates can be complex.

Using self update for Oracle applications masking and subsetting templates, enterprises can directly download these templates from Oracle and import them into their Oracle Enterprise Manager Cloud Control 12*c* environment with no manual intervention.

Enterprises can easily and seamlessly implement the best practices for protecting sensitive data and provisioning reduced sized databases in their Oracle applications non-production environments using the self update option for Oracle applications masking and subsetting templates.

See Also:

Oracle Database Testing Guide for details

2.6.2.3 Database Replay Support for Database Consolidation

Database Replay now supports simultaneous execution of multiple database captures on a single consolidated database. The consolidated database can be a CDB with multiple PDBs or a traditional database consolidated using schema consolidation methods. Consolidated database replay supports scheduling of the individual replays enabling investigations of various scenarios (for example, what if all my individual workloads hit their peak utilizations at the same time).

Consolidated replay supplies the ability to assure desired database performance for database consolidation projects, whether consolidating onto an Oracle database machine or other consolidated infrastructure.

See Also:

Oracle Database Testing Guide for details

2.6.2.4 Database Replay Workload Scale-Up and Characterization

Database Replay supports creation of a new workloads based on existing captured workloads. The new workloads can be used for capacity planning and validation of various what-if scenarios.

Workload manipulation techniques such as workload filtering by various dimensions (such as time, services, and module) and scheduling are used to compose a new workload. Additionally, a custom workload scenario, or scaled-up version of the original workload, can be easily created by using a combination of workload manipulation techniques such as filtering, sub-setting by time, and scheduling them to run at the same time.

A database workload captured using Database Replay can be characterized by various attributes such as request types, activity, access or transaction patterns, and application-defined attributes. This allows division of the captured workload into smaller, more manageable and autonomous units that can also be used to better understand captured workload.

Database workload scale-up and characterization capabilities allow businesses to perform capacity planning and system testing under various what-if scenarios.

See Also:

Oracle Database Testing Guide for details

2.6.2.5 Enhanced Database Replay Reporting

Database Replay reporting has been enhanced to provide insight into the causes of slow or fast replay. Utilizing the Active Session History (ASH) Analytics framework, Database Replay provides additional replay reports allowing you to quickly analyze and understand all of the performance characteristics of your database replay. In addition, you can use ASH Analytics to produce customized reports.

Enhanced Database Replay reporting reduces the time spent by users in analyzing replay performance.

See Also:

Oracle Database Testing Guide for details

2.6.3 General

The following section describes a new general feature.

2.6.3.1 Queryable Patch Inventory

Using `DBMS_QOPATCH`, Oracle Database 12*c* provides a PL/SQL or SQL interface to view the database patches that are installed. The interface provides all the patch information available as part of the OPatch `lsinventory -xml` command. The package accesses the Oracle Universal Installer (OUI) patch inventory in real time to provide patch and patch meta information.

Using this feature, users can:

- Query what patches are installed from SQL*Plus.
- Write wrapper programs to create reports and do validation checks across multiple environments.
- Check patches installed on Oracle RAC nodes from a single location instead of having to log onto each one in turn.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

2.7 Oracle RAC and Grid Infrastructure

The following sections describe the new Oracle RAC and Oracle Grid Infrastructure features with Oracle Database 12*c* Release 1 (12.1).

2.7.1 Oracle ASM Enhancements

The following sections describe Oracle Automatic Storage Management (Oracle ASM) improvements.

2.7.1.1 Oracle Flex ASM

Oracle Flex ASM decouples the Oracle ASM instance from the database servers. Oracle Flex ASM instances may be run on separate physical servers from Oracle Database 12*c* instances. Any number of Oracle Flex ASM servers can be clustered together to support a large set of databases.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.1.2 Oracle ASM Shared Password File in a Disk Group

This feature implements the infrastructure needed to address the bootstrapping issues of Oracle ASM shared password files in an Oracle ASM disk group. This approach simplifies the administration of password files by ensuring that only a single copy needs to be maintained.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.1.3 Oracle ASM Rebalance Enhancements

Oracle ASM rebalance enhancements improve scalability, performance, and reliability of the rebalance operation. This feature extends the rebalance to operate on multiple disk groups in a single instance. In addition, it improves support for thin provisioning, user data validation, and improved error handling.

With this new feature, you can distribute the rebalance load for higher performance, obtain better user validation, and control error handling and support for thin provisioning.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.1.4 Oracle ASM Disk Resync Enhancements

Oracle ASM disk resync allows multiple disks to be brought online simultaneously or to otherwise control the speed of the resync operation. Oracle ASM disk resync now has a Resync Power Limit to control resync parallelism and, therefore, improve performance. Disk Resync Checkpoint allows for faster recovery from instance failures by allowing the resync to resume from where it was interrupted or stopped instead of starting from the beginning.

This enhancement provides faster recovery from instance failure and faster resync performance overall.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.1.5 Oracle ASM chown, chgrp, chmod and Open Files Support

This feature allows the ASMCMD `chown, chgrp,` and `chmod` commands to run even if the affected files are open. In Oracle Database 11*g* Release 2 (11.2), this was not allowed. The ALTER DISKGROUP MODIFY OWNERSHIP SQL command is also similarly modified, as this SQL provides support for these ASMCMD commands.

This feature improves the manageability of Oracle ASM users and the files they own.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.1.6 Oracle ASM Support ALTER DISKGROUP REPLACE USER

This feature introduces a new SQL statement, ALTER DISKGROUP REPLACE USER, that allows the identity of an Oracle ASM user to be changed from one operating system user to another operating system user. This feature allows end users to change the identity of an Oracle ASM user without having to drop and re-create the user, which requires dropping all of the files a user may own.

This feature improves the manageability of Oracle ASM users and the files they own.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.1.7 Enterprise Manager Support for Oracle ASM Features

Enterprise Manager supports the following Oracle ASM features:

- Oracle Flex ASM server
- Disk resync improvements
- Oracle ASM rebalance improvements
- Enable access control for Oracle ASM files on Windows
- Oracle ASM corrupt media recovery (scrubbing)

Customers benefit from the easy-to-use interface to monitor and manage these new features like job scheduling, metrics collection, and so on.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.1.8 Oracle ASM File Access Control on Windows

Oracle ASM File Access Control restricts the access of files to specific Oracle ASM clients that connect as SYSDBA. An Oracle ASM client is typically a database which is identified as the user that owns the database instance home.

Beginning with Oracle Database 12*c* Release 1 (12.1), Oracle supports the use of low-privileged users on Windows instead of Local System Account to run Oracle Database services that let you use separate users for different Oracle databases. This release also supports Oracle ASM disk group file-level access control and privilege separation.

The Oracle ASM File Access Control feature helps to replace the current user with a new user and allows the user to change ownership, group membership, and permissions of a file while the file is open by one or more Oracle ASM clients. In this release, the low-privileged users for specific Oracle homes are restricted from directly accessing Oracle ASM storage devices and can be accessed through the Oracle Database services that have sufficient privileges to run that service.

Oracle ASM disk group users now manage ASM disk group user replacement with new ASMCMD commands and SQL statements.

See Also:

Oracle Database Platform Guide for Microsoft Windows for details

2.7.1.9 Oracle Grid Infrastructure Rolling Migration for One-Off Patches

Rolling migration framework has been enhanced to handle applying one-off patches released for Oracle ASM in a rolling manner. It also enables migration of a database (Oracle Database 12*c* Release 1 (12.1) and above) to another Oracle ASM instance to minimize downtime during the course of the rolling migration.

This feature improves database availability by migrating the database to another Oracle ASM instance prior to shut down and upgrade.

See Also:

Oracle Grid Infrastructure Installation Guide for Linux for details

2.7.2 Oracle ACFS Enhancements

The following sections describe improvements to Oracle Automatic Storage Management Cluster File System (Oracle ACFS).

2.7.2.1 Oracle ACFS Support for All Oracle Database Files

Oracle ACFS provides support for all database files. This feature takes advantage of Oracle ACFS data services for database files.

See Also:

Oracle Automatic Storage Management Administrator's Guide for details

2.7.2.2 Oracle ACFS and Highly Available NFS

For Grid homes, Oracle ACFS exports (using NFS) include Golden Images and patch updates applied to Oracle ACFS snapshots.

The Network File Storage (NFS) is deployed with Grid home servers in support of Grid home client systems. Application VIP (virtual internet protocol address) and NFS export resources are employed for Oracle ACFS and highly available NFS.

See Also:

Oracle Automatic Storage Management Administrator's Guide for details

2.7.2.3 Oracle ACFS Snapshots Enhancements

Oracle ACFS read/write snapshots have been enhanced to support snapshots created from an existing snapshot in the same Oracle ACFS file system (snaps of snaps) and snapshot conversions (read-only to read/write). These support creation of additional sparse patch set extensions (using snapshots) to Oracle ACFS snapshot management.

This enhances snapshot functionality by providing cascading snapshots and bi-directional conversion of read/write and read-only snapshots.

See Also:

Oracle Automatic Storage Management Administrator's Guide for details

2.7.2.4 Oracle ACFS Replication Integration with Oracle ACFS Security and Encryption

This feature enables integration of Oracle ACFS replication with Oracle ACFS security and encryption. Customers can use the combination of Oracle ACFS security and encryption with Oracle ACFS replication.

See Also:

Oracle Automatic Storage Management Administrator's Guide for details

2.7.2.5 Oracle Audit Vault and Database Firewall Support for Oracle ACFS Security and Encryption

This feature enables Oracle Audit Vault and Database Firewall support for Oracle ACFS security and encryption. This new support is for customers who want to use the combination of Oracle Audit Vault and Database Firewall and Oracle ACFS security and encryption.

See Also:

Oracle Automatic Storage Management Administrator's Guide for details

2.7.2.6 Oracle ACFS Security and Encryption Features

Oracle ACFS security feature provides the ability to create realms to specify security policies for users or groups for accessing file system objects. The Oracle ACFS security feature provides a finer-grained access control on top of the access control provided by the operating system.

Oracle ACFS encryption feature provides the ability to keep data in an Oracle ACFS file system in encrypted format to prevent unauthorized use of data in the case of data loss or theft.

See Also:

Oracle Automatic Storage Management Administrator's Guide for details

2.7.2.7 Oracle ACFS File Tags for Grid Homes

Oracle ACFS file tag operations are supported through a common operating system independent file tag API (C library). The patch software tools can use Oracle ACFS file tags to identify the specific patch files applied to a given Oracle ACFS snapshot image in support of sparse file Grid home patch set transfers. Tagged files can also be used in the display of file usage metrics.

This feature allows a programmatic interface to manage the file tag functionality in Oracle ACFS. The Grid home tools are examples of such use cases.

See Also:

Oracle Automatic Storage Management Administrator's Guide for details

2.7.2.8 Oracle ACFS Plug-in APIs

The Oracle ACFS plug-in functionality allows a user space application to collect just-in-time Oracle ACFS file and ASM Dynamic Volume Manager (ADVM) volume metrics from the operating system environment. Oracle and customer applications can leverage the Oracle ACFS plug-in infrastructure to create customized solutions that extend the general application file metric interfaces to include detailed Oracle ACFS file system and volume data.

This feature provides a programmatic interface to collect statistics data.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.2.9 Enterprise Manager Support for Oracle ACFS New Features

Enterprise Manager support for Oracle ACFS features includes:

- Oracle ACFS enhancements for Grid homes
- Oracle ACFS tagging
- Oracle ACFS snapshot enhancements

This support allows all Oracle ACFS file system functionality to be managed by an easy-to-use GUI management interface.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.2.10 Oracle ACFS Replication and Tagging on AIX

Oracle can now port Oracle ACFS replication and tagging on AIX providing broader operating system platform support.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.2.11 Oracle ACFS Replication and Tagging on Solaris

Oracle can now port Oracle ACFS replication and tagging on Solaris providing broader operating system platform support.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* for details

2.7.3 Oracle Clusterware Enhancements

The following sections describe new Oracle Clusterware features.

2.7.3.1 Oracle Flex Cluster

Oracle Flex Cluster is a new Oracle Clusterware based topology utilizing two types of cluster nodes: Hub Nodes and Leaf Nodes. Hub Nodes represent traditional nodes, tightly coupled using network and storage. Leaf Nodes are a new type of node that runs a lighter weight stack and does not require direct shared storage connectivity.

Combining tightly coupled Hub Nodes and lightweight Leaf Nodes in one cluster allows the running of a variety of workload and applications across hundreds of nodes without additional overhead, while maintaining the ability to create dependencies between them. Centralized, cluster-wide management and standardized resource allocation policies further facilitate workload consolidation on Oracle Flex Cluster.

2.7.3.2 Policy-Based Cluster Management and Administration

Oracle Grid Infrastructure allows running multiple applications in one cluster. Using a policy-based approach, the workload introduced by these applications can be allocated across the cluster using a policy set. In addition, a policy set enables different policies to be applied to the cluster over time as required. Policy sets can be defined using a web-based interface or a command-line interface.

Hosting various workloads in the same cluster helps to consolidate the workloads into a shared infrastructure that provides high availability and scalability. Using a centralized policy-based approach allows for dynamic resource reallocation and prioritization as the demand changes.

See Also:

Oracle Clusterware Administration and Deployment Guide for details

2.7.3.3 What-If Command Evaluation

Oracle Clusterware now provides a set of evaluation commands to determine the impact of a certain operation before the respective operation is actually executed.

Clusterware administrators are constantly challenged to react to changes in the cluster. The What-If command evaluation helps clusterware administrators to determine the impact of certain operations before they are executed in order for them to make the most appropriate decision and, thereby, ensure smooth cluster operation.

See Also:

Oracle Clusterware Administration and Deployment Guide for details

2.7.3.4 Oracle Cluster Registry Backup in ASM Disk Group Support

The Oracle Cluster Registry (OCR) backup mechanism enables storing the OCR backup in an Oracle ASM disk group.

Storing the OCR backup in an Oracle ASM disk group simplifies OCR management by permitting access to the OCR backup from any node in the cluster should an OCR recovery become necessary.

See Also:

Oracle Clusterware Administration and Deployment Guide for details

2.7.3.5 Shared Grid Naming Service (GNS)

In previous releases, the Grid Naming Service (GNS) was dedicated to one Oracle Grid Infrastructure-based cluster, providing a name resolution for only those nodes that were part of this cluster. This restriction has been lifted and one GNS can now manage either the nodes of one cluster only or act as the naming resolution for all nodes across all clusters in the data centers.

Using only one GNS for all nodes that are part of an Oracle Grid Infrastructure-based cluster in the data center not only streamlines the naming convention, but also enables a data center cloud minimizing day-to-day administration efforts.

2.7.3.6 SRVCTL Support for Oracle Flex Cluster Implementations

Oracle Flex Cluster is a new concept, which joins together a traditional closely coupled cluster with a modest node count with a large number of loosely coupled nodes.

In order to support various configurations that can be established using this new concept, SRVCTL provides new commands and command options to ease the installation and configuration.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for details

2.7.3.7 Online Resource Attribute Modification

Oracle Clusterware manages hardware and software components for high availability using a resource model. For example, the listener or VIPs used by an Oracle RAC database are managed using respective listener and VIP resources. Resource attributes are used to define how Oracle Clusterware manages those resources. For example, a resource attribute is used to define the subnet of a VIP resource or the name of a listener. Using online resource attribute modification, certain attributes of a resource can be modified without the need to restart the resource for the change to take effect. Online resource attribute modification is available using certain SRVCTL and CRSCTL commands.

Being able to modify certain resource attributes without having to restart the resource for the change to take effect increases resource availability, reduces the likeliness for failures, and overall simplifies the resource management in high availability environments.

See Also:

Oracle Clusterware Administration and Deployment Guide for details

2.7.4 Other Grid Infrastructure Enhancements

The following sections describe new general Oracle Grid Infrastructure features.

2.7.4.1 Grid Infrastructure Script Automation for Installation and Upgrade

Using script automation for installation and upgrade eliminates the need to run scripts manually on each node during the final steps of an Oracle Grid Infrastructure installation or upgrade. This feature minimizes the likelihood of errors and simplifies the installation process.

See Also:

Oracle Grid Infrastructure Installation Guide for Linux for details

2.7.4.2 Multipurpose Cluster Installation Support

Earlier Oracle releases required post-installation steps to cover certain configuration scenarios. The new Oracle Universal Installer (OUI) allows for the most commonly used configuration to be installed during the initial installation flow.

Integrating the most common configuration scenarios into the default installation flow of OUI for the Oracle Grid Infrastructure makes the installation easier, less error prone, and overall more efficient.

> **See Also:**
>
> *Oracle Grid Infrastructure Installation Guide for Linux* for details

2.7.5 Oracle RAC Enhancements

The following sections describe new features for Oracle RAC.

2.7.5.1 Support for IPv6 Based IP Addresses for Oracle RAC Client Connectivity

Cluster nodes can be configured to use either IPv4 or IPv6 based IP addresses for the Virtual IPs (VIP) on the public network, while more than one public network can be defined for the cluster. Database clients and applications can connect to either IPv4 or IPv6 VIP addresses. The Single Client Access Name (SCAN) listener automatically redirects client connects to the appropriate database listener within a given subnet considering the IP protocol requested by the client. SCAN listeners can be defined for each subnet in the cluster.

IPv6 based IP addresses have become the latest standard for the information technology infrastructure in today's data centers. With this release, Oracle RAC and Oracle Grid Infrastructure support this standard for client connectivity.

> **See Also:**
>
> *Oracle Clusterware Administration and Deployment Guide* for details

2.8 Performance

The following sections describe the new performance features for Oracle Database 12*c* Release 1 (12.1).

2.8.1 Database Performance Enhancements

The following sections describe improvements in database performance.

2.8.1.1 Advanced Network Compression

New parameters, `SQLNET_COMPRESSION` and `SQLNET.COMPRESSION_SCHEME_LIST`, allow the compression of data transitioning over Oracle Net Services between client and server. Compression can be enabled at the:

- Connection level (connect string, URL)
- Service level (`tnsnames.ora, ldap.ora`)
- Database level (`sqlnet.ora`)

Network acceleration reduces network latency in local area network (LAN) and wide area network (WAN) environments and increases performance.

> **See Also:**
>
> *Oracle Database Net Services Reference* for details

2.8.1.2 Very Large Network Buffers

Support for larger packets (session data unit (SDU) sizes) in the Oracle Net layer has been added in this release. SDU defines the size of internal buffers. In previous releases, the default was 8K with a maximum of 64K. This release pushes the limit beyond the current values.

The benefits of very large network buffers are increased application throughput through optimization and better utilization of available network bandwidth.

See Also:

Oracle Database Net Services Administrator's Guide for details

2.8.2 General

The following sections describe new general features.

2.8.2.1 Asynchronous I/O Control for Direct NFS Client

The `DNFS_BATCH_SIZE` parameter controls the number of asynchronous I/O requests that can be queued by an Oracle process when Direct NFS Client is enabled. In environments where the Network File Storage (NFS) server cannot handle a large number of outstanding asynchronous I/O requests, this parameter can be used to limit the number of I/O requests issued by an Oracle foreground process. The recommended setting for this parameter is to start at 128K and increase or decrease it based on NFS server performance.

Some NFS servers do not perform well if too many asynchronous I/O requests are delivered within a short period of time. The default Direct NFS Client setting for this behavior is higher than some NFS servers can handle. The `DNFS_BATCH_SIZE` parameter allows customers to tune the behavior of Direct NFS Client to match their specific NFS server requirements, maximizing performance and system stability.

See Also:

Oracle Database Performance Tuning Guide for details

2.8.2.2 Tracking I/O Outliers

Oracle Database maintains a `V$IO_OUTLIER` view that captures I/O operations that take a long time to complete. Long delays in I/O completions can cause serious performance degradation. DBA's can monitor the `V$IO_OUTLIER` view to see the health of the I/O subsystem. Although the `V$IO_OUTLIER` view provides clues for overall I/O latencies, DBA's often need to drill down further to know where in the operating system high latencies are seen. Oracle Database now uses the dynamic tracing feature in Solaris to gather information about the amount of time an I/O spends in each of the key operating system subsystems it passes through on its way to the physical device. The timing information for each component is recorded in the new `V$KERNEL_IO_OUTLIER` view. This provides an operating system-based latency breakdown of I/O taking an excessive amount of time to complete.

This is an essential tuning and debugging tool that enables faster identification and resolution of I/O related issues.

See Also:

Oracle Database Reference for details

2.8.3 Hardware Optimizations

The following sections describe hardware configuration and technology features.

2.8.3.1 Multi-Process Multi-Threaded Oracle

Multi-process multi-threaded Oracle uses multiple processes and multiple threads within each process to provide a new execution model for Oracle Database. Support for multi-process multi-threaded Oracle provides improved performance and manageability through more efficient sharing of system and processor resources.

See Also:

Oracle Database Concepts for details

2.8.4 Improve Ease of Performance Out-of-the-Box

The following section describes improvements to out-of-the-box performance.

2.8.4.1 Ability to Specify NFS Version in Direct NFS Client

The `NFS_VERSION` parameter allows the user to specify the NFS protocol to be used by Direct NFS Client. Possible values are `nfsv3, nfsv4,` and `nfsv4.1`. If no version is specified, `nfsv3` is used by default.

NFS Version 4 provides performance advantages over NFS Version 3, and the `NFS_VERSION` parameter allows the user to take advantage of those capabilities when available.

See Also:

Oracle Database Installation Guide for Linux for details

2.9 Security

The following sections describe the new database security features for Oracle Database 12*c* Release 1 (12.1).

2.9.1 Data Encryption, Hashing and Redaction

The following sections describe the encryption, hashing and redaction features.

2.9.1.1 Oracle Data Redaction

This new database security feature is part of Oracle Advanced Security and prevents data columns (such as credit card numbers, U.S. Social Security numbers, and other sensitive or regulated data) from being displayed. It is driven by declarative policies that can take into account database session factors and information passed by applications. Sensitive display data can be redacted at runtime on live production systems with minimal disruption to running applications and without altering the actual stored data. Different types of redaction are supported including full, partial, random, and regular expression redaction. You can conceal entire data values or redact only part of the value. The functionality is implemented inside of the database, therefore separate installation is not required.

See Also:

Oracle Database Advanced Security Guide for details

2.9.1.2 Support for Secure Hash Algorithm SHA-2 in Oracle Database

Oracle Database 12*c* support for the SHA-2 algorithm builds upon the existing support for SHA-2 in Oracle Database 11.2.0.3. The expanded support for the SHA-2 algorithm includes the PL/SQL `DBMS_CRYPTO` package.

Support for the SHA-2 algorithm provides increased security assurance for Oracle Database. In addition, it provides compliance with regulations that may now or in the future require use of the SHA-2 algorithm for hashing of sensitive data.

See Also:

Oracle Database Security Guide for details

2.9.2 Database Security Enhancements

The following sections describe database security enhancements.

2.9.2.1 Auditing Enabled By Default

The new unified auditing architecture can be used in Oracle Database with no changes required to the database initialization parameters. This feature enables audit policies to be created and enabled in the database with no production database downtime, providing flexibility and ease of administration for database auditing.

See Also:

Oracle Database Security Guide for details

2.9.2.2 Code-Based Security

Code-based security enables roles to be associated with PL/SQL packages, functions, and procedures. Associating roles with packages, functions, and procedures provides finer granularity for privileged grants, eliminating the need to grant these roles directly to the runtime users.

Code-based security provides increased security for applications by enabling roles only for the execution scope of the PL/SQL program units without granting them directly to the user. Scoping the grants of roles reduces the database privilege grants to users and helps enforce the security concept of least privilege.

See Also:

Oracle Database Security Guide for details

2.9.2.3 Data Guard Support for Separation of Duty (SoD)

This feature makes it possible to administer a Data Guard configuration without requiring `SYSDBA` privileges. Administration of Data Guard configurations can be delegated to a class of users who would not be granted `SYSDBA` privileges.

See Also:

Oracle Data Guard Concepts and Administration for details

2.9.2.4 Enhanced Security of Audit Data

The new unified audit policy-based database audit architecture stores audit records in read-only tables. This new audit table is created as part of the Oracle database infrastructure. The maintenance of the audit trail records are implemented using the audit trail cleanup package that can only be used by users with the new `AUDIT_ADMIN` administrator role.

Security and compliance regulations require accurate monitoring and reporting of Oracle database activity. The new read-only table provides increased assurance that audit records are not modified or deleted after they have been written to the audit trail. Maintenance of the new audit trail is limited to users who have been granted the new `AUDIT_ADMIN` role. Only users with the new `AUDIT_ADMIN` role can manage the retention policy of the audit data.

See Also:

Oracle Database Security Guide for details

2.9.2.5 Increased Security When Using SELECT ANY DICTIONARY

The `SELECT ANY DICTIONARY` privilege no longer permits access to security sensitive data dictionary tables `DEFAULT_PWD$, ENC$, LINK$, USER$, USER_HISTORY$,` and `XS$VERIFIERS`.

This change increases the default security of the database by not allowing access to a subset of data dictionary tables through the SELECT ANY DICTIONARY privilege.

2.9.2.6 Last Login Time Information

The last login time for database users is recorded in the `USER$` table and displayed when connecting to the database using Oracle SQL*Plus.

Recording the last login time for database users increases database security by providing security administrators the ability to determine when an account was last used in the database. Displaying the last login time in the Oracle SQL*Plus connection banner provides the SQL*Plus user information on their last account usage.

See Also:

*SQL*Plus User's Guide and Reference* for details

2.9.2.7 Oracle Database Vault Mandatory Realms

Oracle Database Vault mandatory realms block both DBA privileges and direct object privilege grants, including the object owner. Traditional Oracle Database Vault realms protect against the common `DBA ANY` system privileges, preventing privileged users from accessing realm-protected objects using their `SELECT ANY` privilege. With the mandatory realm, users with direct object privileges, including the object owner, are blocked from accessing realm protected objects as well. As with traditional realms, users who need access are authorized using the realm authorization capability of Oracle Database Vault.

Oracle Database Vault mandatory realms provide increased protection for sensitive application tables that exist within a larger application. Using this feature, application tables that contain highly sensitive information can be placed in a mandatory realm and users with direct object grants are blocked from accessing data contained in those tables. Mandatory realms can also be used in situations where database administrators, support analysts, or developers need temporary access to an application schema but access to specific application tables needs to be blocked.

See Also:

Oracle Database Vault Administrator's Guide for details

2.9.2.8 Oracle Label Security Metadata Export and Import

Oracle Label Security metadata in the `LBACSYS` schema can be included when doing a full database export and import operation. The source database can be Oracle

Database 11*g* Release 2 (11.2.0.3), or higher. The target database must be Oracle Database 12*c* Release 1 (12.1) or higher.

Oracle Label Security metadata export and import provides the ability to move Oracle Label Security policies and protected tables between databases.

See Also:

Oracle Label Security Administrator's Guide for details

2.9.2.9 Password Complexity Check

New databases created using the Oracle Database Configuration Assistant (DBCA) can optionally have a default password complexity check enabled. Password complexity checks increase the security of Oracle databases and the overall enterprise by reducing the potential for new databases to be created without a strong password check enabled.

See Also:

Oracle Database Security Guide for details

2.9.2.10 Privilege Analysis

Privilege analysis, which is available with Oracle Database Vault, enables customers to create a profile for a database user and capture the list of system and object privileges that are being used by this user. The customer can then compare the user's list of used privileges with the list of granted privileges and reduce the list of granted privileges to match the used privileges.

Privilege analysis helps improve the security of applications and operations by identifying unused or excessive privileges. Privileges required by database administrators can easily be identified by analyzing the privileges used while performing common administration activities. Privileges required by applications can be easily identified by running privilege analysis during an application connection to the database.

See Also:

Oracle Database Vault Administrator's Guide for details

2.9.2.11 Resource Role Default Privileges

The `UNLIMITED TABLESPACE` privilege is no longer the default `RESOURCE` role starting in Oracle Database 12*c*. This change increases the default security of the database by eliminating the potential for database users and applications that have been granted the `RESOURCE` role to exceed their intended resource quotas for tablespaces.

See Also:

Oracle Database Security Guide for details

2.9.2.12 Separation of Duty for Audit Administration

The new unified context-based database audit configuration provides two new roles for managing database auditing. The new `AUDIT_ADMIN` role provides the ability to create and enable new audit policies and specify the audit record retention policy. The new `AUDIT_VIEWER` role provides auditors and security administrators the ability to view audit data in the new unified audit trail.

Separation of duty in the new unified context-based database audit architecture provides the ability to selectively assign the users that may create, enable, and delete audit policies while still allowing security team members and managers to review the audit data that has been generated. Database administration can be separated from audit administration, increasing the security of day-to-day operations.

See Also:

Oracle Database Security Guide for details

2.9.2.13 Separation of Duty for Database Administration

Oracle Database provides new roles for database administrative activities such as backup and recovery, high availability, and key management. Providing new roles for common database administration tasks increases the security of the Oracle database by eliminating the need to grant the highly privileged `SYSDBA` role for common day-to-day operations.

See Also:

Oracle Database Security Guide for details

2.9.2.14 SYSBACKUP Administration Privilege

A new administration privilege, `SYSBACKUP`, allows Recovery Manager (RMAN) users to connect to the target database and run RMAN commands, no longer requiring `SYSDBA`.

This feature enforces the separation of duty security model, whereby backup operators only need `SYSBACKUP` privilege to run RMAN commands and have separate responsibilities from database administrators who need real `SYSDBA` privileges.

See Also:

Oracle Database Backup and Recovery User's Guide for details

2.9.3 Encryption Key Management Enhancements

The following sections describe encryption key management enhancements.

2.9.3.1 Updated Key Management Framework

This feature updates the Oracle Advanced Security Transparent Database Encryption (TDE) key management capabilities with a range of new functionality including:

- A common layer for keystore management that enables consistent administration of Oracle keystores for TDE (called wallets in previous releases) and third-party Hardware Security Modules (HSMs).
- New key management SQL statements (`ADMINISTER KEY MANAGEMENT`) that consolidate functionality previously provided in separate Oracle utilities.
- New metadata for tracking important attributes of master encryption keys.
- New built-in database views for examining keys and their attributes.
- A `SYSKM` database administrative privilege for managing keystores and master encryption keys.
- Support for exporting or importing individual keys from the keystore to move them between Oracle databases.
- Support for storing TDE keystores directly on Oracle ASM managed disk groups, with no requirement for an additional file system.

The updated key management framework provides a more flexible, secure, and user-friendly key management interface.

See Also:

Oracle Database Advanced Security Guide for details

2.9.4 Improve Security Manageability, Administration and Integration

The following sections describe improved security manageability, administration and integration features.

2.9.4.1 Oracle Database Vault Persistent Protections

The security controls provided by Oracle Database Vault no longer depend on the Oracle executable. The same Oracle software home can be used for databases with or without Oracle Database Vault enabled.

Oracle Database Vault can be enabled much faster and easier across the enterprise. The controls enforced by Oracle Database Vault remain enforced when the databases are moved to a new Oracle home or to a new server, simplifying administration and increasing security.

See Also:

Oracle Database Vault Administrator's Guide for details

2.9.4.2 Simplified Oracle Database Vault and Oracle Label Security Installation

The installation of Oracle Database Vault and Oracle Label Security has been simplified. In previous releases, the software for these solutions was not installed by default. In this release, the software is installed by default but not enabled. Enabling Oracle Database Vault or Oracle Label Security now requires a simple command line configuration step.

The default installation of Oracle Database Vault and Oracle Label Security simplifies the use of these powerful security solutions, enabling additional security controls to be configured quickly and easily on production systems, helping protect against security threats.

See Also:

Oracle Database Installation Guide for Linux for details

2.9.4.3 Transparent Sensitive Data Protection

Transparent sensitive data protection enables you to protect sensitive data consistently in the database based on a classification type (for example, credit card numbers whose columns use a specific data type). This feature makes it easier to manage database enforced protections around sensitive data as well as enforce additional protections. In addition, you can easily export transparent sensitive data protection policies to other databases. You can use transparent sensitive data protection with Oracle Data Redaction policies.

Transparent sensitive data protection provides the ability to apply protection policies across data classifications inside the Oracle database, reducing the cost and complexity of protecting sensitive data. By applying policies across classification types, the need to apply polices on a column-by-column basis is eliminated.

See Also:

Oracle Database Security Guide for details

2.9.4.4 VPD Fine-Grained Context-Sensitive Policies

Fine-grained context-sensitive policies provide the ability to associate one or more (`context,attribute`) pairs with a virtual private database (VPD) policy. The VPD policy function gets evaluated only when one of the (`context,attribute`) pair changes its value.

Fine-grained context sensitive policies provide improved performance for applications using virtual private database.

See Also:

Oracle Database Security Guide for details

2.9.5 Protect the Database Server From Outside

The following section describes protecting the database server from outside.

2.9.5.1 Restricted Service Registration for Oracle RAC

Listeners managed by Oracle Grid Infrastructure can be configured to restrict clients from accessing a database registered with this listener using various conditions, for example, the subnet from which these clients are connecting. Restricting client access to a database makes Oracle RAC even more secure and less vulnerable to security threats and attacks.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for details

2.9.6 Real Application Security

A security infrastructure is needed in the database for application security that understands application users and roles natively along with their access rights and ACLs so that they can be enforced in the database securely and efficiently. With declarative and extensible security policies, customers can build secure applications quickly.

The following sections describe improvements in Real Application Security.

2.9.6.1 Real Application Security

Real Application Security provides an Oracle database authorization solution for end-to-end application security. It specifies, provisions, and enforces application-level security policies at the database layer, eliminating the task of building custom application logic to handle application users, their authorizations, and security policies on data. A wider range of data-centric security policies and constraints on application users' authorization can be defined inside the Oracle database, providing a consistent and uniform authorization model across applications.

Real Application Security strengthens overall application and data security and ultimately reduces application development time by moving security controls from the application layer to where the data resides in the database. Application users, privileges, roles, grants, and security policies can be defined, provisioned, and enforced at the database layer, enhancing security of the data and application. It reduces custom development of application security by providing security features, such as privilege delegation, role-based constraints, time-based access control, code-based security, multi-level authorization, negative grants, authorization on user interface artifacts, access constraints on relational data, and application users auditing. Enforcement of application security at the database layer increases security for data by enforcing application security logic regardless of the access path to the database.

See Also:

Oracle Database Real Application Security Administrator's and Developer's Guide for details

2.9.7 Security Optimizations

The following sections describe security optimization features.

2.9.7.1 Unified Context-Based Database Audit Architecture

The Oracle database now supports a single unified audit trail and a new policy syntax that enables named audit policies to be created inside the Oracle database. This powerful new audit implementation supports context-based conditions, limiting when an audit record should be created. In addition, auditing can be specified for specific database roles and a set of users can be listed as exempt from auditing.

Auditing is playing an increasingly important role in security. The new unified audit trail and policy syntax simplifies management of database auditing and provides highly granular controls over when to audit, optimized performance, and flexibility for security and compliance.

See Also:

Oracle Database Security Guide for details

2.10 Spatial and Graph

This section describes the new Spatial and Graph features available with Oracle Database 12*c*.

2.10.1 Oracle Spatial and Graph Enhancements

Oracle Spatial has been renamed Oracle Spatial and Graph. While Oracle Database has supported native graphs as a feature since the release of Oracle Spatial 10*g*, this change has been made to highlight the existing graph capabilities in Oracle Spatial and in recognition of the increasing market demand for graph database capabilities.

Oracle Spatial and Graph includes Network Data Model (NDM) graphs used in traditional network applications in major transportation, telecommunications, utilities, and energy organizations. It also includes support for Resource Description Framework (RDF) Semantic Graphs used in social networks and social interactions to address requirements from the media, finance, research, life sciences, pharmaceutical, and intelligence communities. These are proven, robust graph database technologies.

The following sections describe enhancements to these Oracle Spatial and Graph capabilities.

2.10.1.1 Vector Performance Acceleration

Vector operations can be substantially improved by invoking new vector performance acceleration capabilities in Oracle Spatial and Graph. These result in improved index performance, enhanced geometry engine performance, optimized secondary filter optimizations for Spatial operators, and improved CPU and memory utilization for many advanced vector functions. Vector performance acceleration is especially beneficial when using Oracle Exadata Database Machine and other large-scale systems.

Oracle Spatial and Graph vector performance acceleration builds on general improvements available to all `SDO_GEOMETRY` operations in the following areas:

- Caching of index metadata
- Concurrent update mechanisms
- Optimized spatial predicate selectivity and cost functions

These optimizations enable more efficient use of CPU, memory, and partitioning, resulting in substantial query performance improvements. For example, internal test results show up to 100 times faster query performance than with the previous release for non-geodetic point data and a polygon query window.

See Also:

Oracle Spatial and Graph Developer's Guide for details

2.10.1.2 Spatial Routing Engine Enhancements

The Oracle Spatial and Graph routing engine supports truck-specific routing and logical turn restrictions. It computes drive times based on truck speed limits, which often differ from car speed limits. It also provides information on truck services such as weigh stations and truck stops along a route. Finally, it can handle logical turn restrictions involving more than two edges in the route geometry.

These enhancements yield more accurate results for logistics and truck routing applications.

See Also:

Oracle Spatial and Graph Developer's Guide for details

2.10.1.3 Spatial GeoRaster - Raster Algebra and Analytics

Oracle Spatial and Graph GeoRaster includes a new raster algebra language providing local algebraic function types and related raster operations. Coupled with PL/SQL, it enables expressing complex pixel query, value-based raster editing, mathematical operations, classification, and cartographic modeling over large numbers of rasters and images of unlimited size. These operations can be parallel-enabled to significantly improve performance.

You can develop cartographic modeling applications with the server-based raster analytic capabilities and a set of efficient I/O and cell manipulation utilities and programming interfaces. With the growing availability of raster data, you can realize performance and scalability benefits from moving the analytic processing to the database, as opposed to trying to perform analysis on client tools.

See Also:

Oracle Spatial and Graph GeoRaster Developer's Guide for details

2.10.1.4 Spatial GeoRaster - Enhanced Image Processing

Oracle Spatial and Graph GeoRaster provides new image processing capabilities. These include image rectification, orthorectification, image stretching, image segmentation, image update and appending, advanced mosaicking, large scale virtual mosaic, and on-the-fly spatial queries.

More image processing can now be handled in the server instead of the client, and some of them are parallelized. This enables improved performance of image processing on a much larger scale, with larger data sets, which are increasingly being used in government and commercial applications as raster data becomes more available.

See Also:

Oracle Spatial and Graph GeoRaster Developer's Guide for details

2.10.1.5 Spatial GeoRaster - Java API Enhancements

The Oracle Spatial and Graph GeoRaster Java API has been enhanced to support features such as ground control point (GCP) storage and manipulation, GCP georeferencing, reprojection, grid interpolations, and `getCellValue`. These features were previously supported only by the PL/SQL API.

With these enhancements, Java developers have more access to Oracle Spatial and Graph GeoRaster data management features.

See Also:

Oracle Spatial and Graph Java API Reference for details

2.10.1.6 Spatial GeoRaster - New Metadata Content

Oracle Spatial and Graph GeoRaster is enhanced to support relational raster data tables (RDTs) and to allow users to specify default alpha channel and pyramid level in its metadata. GeoRaster also adds a new resampling algorithm, supports resolution unit specification and parallel processing in many operations, and adds some new loading and exporting capabilities.

These new features improve GeoRaster data manageability, usability, security, and performance.

See Also:

Oracle Spatial and Graph GeoRaster Developer's Guide for details

2.10.2 Network Data Model and RDF Semantic Graph Enhancements

The following sections describe enhancements to the graph features in Oracle Spatial and Graph.

2.10.2.1 Network Data Model Graph - Feature Modeling and Analysis

Feature modeling bridges the gap between abstract network elements and concrete objects of interest in real world applications. Oracle Spatial and Graph network data model now includes feature modeling and analysis. The previous release only included interfaces at the network element level for editing and analysis. With feature modeling, the application can make one call to a feature analysis function to get the resulting feature representation.

Feature modeling offloads the burden for application developers of maintaining the mapping of application objects to network elements. For example, when a utility network application needs to find affected households when a substation experiences a power failure, the application would have to do the mapping between the application features (substations, power lines and transformers) and network elements (links and nodes). With feature modeling, this relationship is maintained through feature metadata, removing the need for the development and maintenance of application code.

See Also:

Oracle Spatial and Graph Topology Data Model and Network Data Model Graph Developer's Guide for details

2.10.2.2 Network Data Model Graph - Temporal Modeling and Analysis

Oracle Spatial and Graph network data model now supports modeling of networks with the dimension of time. Users may associate time attributes with nodes and links, and specify temporal inputs in their network analysis queries.

Most real-world networks have a dependence on time. Travel times on road segments vary with the time of day. Utility networks experience different demand loads based on seasonal demand and the time of day. Analytic and planning applications can benefit from more accurate representation of real-world conditions. Oracle Spatial and Graph network data model supports queries such as finding the fastest travel route for a specified time of day, modeling and analysis of multimodal transportation networks, and computing the fastest paths on multimodal transportation networks.

See Also:

Oracle Spatial and Graph Topology Data Model and Network Data Model Graph Developer's Guide for details

2.10.2.3 RDF Views on Relational Tables

Resource Description Framework (RDF) views can be created on a set of relational tables or views. Semantic graph queries on RDF views can integrate relational data and RDF Semantic Graph triple data stored in Oracle Database. Oracle supports semantic queries on these views written in the SPARQL 1.1 query language. It does this using Oracle's Jena/Joseki SPARQL endpoint. In addition, it supports the use of the Oracle SQL `SEM_MATCH` table function with embedded SPARQL 1.1 graph patterns.

Semantic graph queries provide a powerful capability to discover relationships through pattern matching on RDF graph data. RDF views extend semantic discovery to relational tables without requiring the conversion of relational tables to RDF triples. This removes the need to duplicate data and the associated storage previously required to perform RDF graph queries on relational data sets.

Semantic graph queries and RDF views can also be used to enable data integration and discovery within and across relational schema and RDF graphs in Oracle Database. This simplifies semantic discovery work flows.

See Also:

Oracle Spatial and Graph RDF Semantic Graph Developer's Guide for details

2.10.2.4 RDF Semantic Graph Named Graph Support

Oracle Spatial and Graph supports Resource Description Framework (RDF) named graphs as defined by the World Wide Web Consortium (W3C). RDF triples can be loaded into one or more named graphs using SQL `INSERT` and bulk load. In addition, the Sesame and Jena Adapters for Oracle Database now support named graph-extended inference, query, and loading APIs. Global inferencing over a group of named graphs, along with local inferencing over a single named graph, with or without a common ontology, is also supported.

This simplifies application development by providing a scalable mechanism for meaningful compartmentalization of associated triples in the graph. Loading, querying, and inferencing can now be performed at the named graph level, reducing the cost and improving the performance of these operations compared to similar operations across the overall semantic network.

See Also:

Oracle Spatial and Graph RDF Semantic Graph Developer's Guide for details

2.10.2.5 Support for Analytic Operations and Tools

Oracle Spatial and Graph RDF Semantic Graph now 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.

In addition, user-provided inference extensions can implement aggregation, arithmetic and advanced analytical functions.

These enhancements allow data represented in RDF graphs and triple stores to be delivered to powerful analysis tools, enabling analysts to derive more information, more quickly.

The built-in graph analysis functions and support for user-provided inference extensions simplify application development and consolidate analytics on the server instead of the client system to address extremely large data sets common to semantic graph applications. The user-provided inference extensions also give users more control over the functions and optimization of the inference process and the results that are generated.

See Also:

Oracle Spatial and Graph RDF Semantic Graph Developer's Guide for details

2.10.2.6 RDF Semantic Graph Support for XML Schema, Text and Spatial Data Types

Oracle Spatial and Graph RDF Semantic Graph now has support for Oracle Database XML schema, text, and spatial data types. This includes an API to add, drop, and alter data type indexes.

Semantic queries written in SPARQL or SQL can now be filtered using XML schema, text, and spatial attributes. This improves the performance and selectivity of queries based on keywords, geography and distance, and document structure.

See Also:

Oracle Spatial and Graph RDF Semantic Graph Developer's Guide for details

2.10.2.7 RDF Semantic Graph Document Indexing Enhancements

The semantic indexing API for documents loads the output from third-party entity extraction services into an RDF graph. This graph provides an index of the entities in the document collection.

Enhancements include:

- Batch indexing of documents.
- Flexible framework for managing entity extraction engines and associated rules.
- Local partitioned indexing.

It also provides a new operator to calculate the relevance of found documents. Finally, documents can be inferenced individually or as a group with a domain-specific ontology.

These enhancements to the RDF Semantic Graph feature in Oracle Spatial and Graph provide the following benefits:

- Enable more efficient processing of large document workloads by supporting batch indexing.
- Simplify the configuration and use of multiple entity extraction tools against the same documents.
- Improve the performance of indexing operations by accessing relevant partitions when processing documents.
- Speed the loading of triples from entity extraction engines by enabling parallel operations.

> **See Also:**
>
> *Oracle Spatial and Graph RDF Semantic Graph Developer's Guide* for details

2.10.2.8 RDF Semantic Graph Support for W3C and OGC Standards, Open Source, and Third-Party Technologies

The RDF Semantic Graph in Oracle Spatial and Graph supports new standards, new versions of standards, and popular open source and third-party semantic tools.

It now conforms to W3C SPARQL 1.1 and the Open Geospatial Consortium (OGC) GeoSPARQL 1.1 language standards. The native inference engine supports the latest W3C OWL 2 EL profile and the extensibility framework supports open source specialty reasoners, such as TrOWL and Pellet. Enhancements to the Oracle Spatial and Graph Jena Adapter feature include distributed querying of SPARQL endpoints and a SPARQL Gateway that enables popular analytical tools capable of accessing XML data sources to process SPARQL query results. The Jena Adapter also has several unique extensions for query execution control and management, including query timeout and abort, query optimizer hints in SPARQL syntax, property paths, results and metadata caching, and user-defined functions.

Oracle RDF Semantic Graph conforms to the latest W3C and OGC standards, as well as open source frameworks. It allows users to take advantage of proven relational and XML-based tools, including Oracle Business Intelligence Enterprise Edition (OBIEE) and Oracle Advanced Analytics (Oracle Data Mining and Oracle R Enterprise) to analyze and visualize the results of semantic queries. The Jena Adapter now provides up to a 45% improvement in query performance with mid-tier caching as well as scalable querying of distributed SPARQL endpoints.

See Also:

Oracle Spatial and Graph RDF Semantic Graph Developer's Guide for details

2.11 Unstructured Data

The following sections describe the unstructured data features for Oracle Database 12*c* Release 1 (12.1).

2.11.1 Oracle Multimedia Enhancements

The following sections describe enhancements to Oracle Multimedia.

2.11.1.1 DICOM Protocol Support for Oracle Database

Oracle Multimedia has supported the management of Digital Imaging and Communications in Medicine (DICOM) format data types since the introduction of Oracle Database 11*g*. Oracle Multimedia now has support for the DICOM protocol, the universally accepted standard for communicating DICOM images over computer networks. This allows users to use the DICOM protocol to store and access DICOM content in Oracle Database.

DICOM applications and devices can now easily access DICOM data in Oracle Database, enabling Oracle Database to store and manage DICOM content as part of a clinical workflow. Large repositories of DICOM content can be managed and secured using Oracle Database tools, reducing management costs. The inclusion of images in electronic health care record management systems and other applications is simplified.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

2.11.1.2 Integrating Oracle Multimedia DICOM with Oracle WebCenter Content

Oracle Multimedia now enables Oracle WebCenter Content to store and access DICOM (Digital Imaging and Communications in Medicine) content in Oracle Database. It includes a DICOM protocol adapter that supports access to Oracle WebCenter Content DICOM data sources and DICOM viewers. It also has an Oracle WebCenter Content component that extracts DICOM metadata into the Oracle WebCenter Content store as well as performing thumbnail generation and image format conversion to web-friendly image formats. It includes support for both access to the DICOM content in its original repository and transfer of the DICOM content into Oracle WebCenter Content.

Support for DICOM content in Oracle WebCenter Content simplifies the development and management of image-enabled patient portals, referring physician portals, electronic medical records (EMRs), and life sciences research applications.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

2.11.1.3 Full Mode Database Import and Export for Oracle Multimedia

Oracle Multimedia now supports full mode database export and import using Oracle Data Pump. This simplifies Oracle Data Pump export and import operations of databases using Oracle Multimedia because special handling of the Oracle Multimedia DICOM data model is no longer required.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

2.11.2 Oracle Text Enhancements

The following sections describe enhancements to Oracle Text.

2.11.2.1 Near Real-Time Indexing

Near real-time indexing allows for frequent synchronization of indexes with heavy DML by maintaining recently changed index information in a new staging index which is designed to remain in memory. Data can be periodically moved from the staging index to the main index by means of a new `MERGE` mode for index optimization. The new option is turned on using the `STAGE_ITAB` storage option.

The new staging index table is relatively small and easy to cache in memory. When resident in memory, there is virtually no cost to this part of the index being fragmented. By separating the fragmented index from the unfragmented main index, performance improves and users are allowed to synchronize their indexes frequently without slowing down query performance. When used with the `TRANSACTIONAL` and `SYNC(ON COMMIT)` index parameters, the index is effectively synchronous.

See Also:

Oracle Text Application Developer's Guide for details

2.11.2.2 Automatic Management of Near Real-Time Indexes

In conjunction with near real-time indexes, automatic management allows for a background task that avoids the need for running optimize merge to move data from the small (normally in memory) `$G` table to the larger (normally on disk) `$I` table. The automatic management process runs in the background when the system is not in heavy use. Indexes must be registered with the management system if they are to be automatically optimized.

This feature simplifies management and improves performance for near real-time indexes and avoids the risk of a manual optimize merge slowing down the system.

See Also:

Oracle Text Reference for details

2.11.2.3 BIG_IO Large TOKEN_INFO Option

A new storage preference, `BIG_IO`, specifies that `TOKEN_INFO` should be stored, where possible, in a single large SecureFiles database field rather than using in-line `BLOBS` limited to 4,000 bytes. This avoids the need to do many seeks when loading large `TOKEN_INFO` data items from disk. Sequential I/O is generally much faster than random I/O, thus improving performance.

See Also:

Oracle Text Reference for details

2.11.2.4 Separate Offsets

`DOCID` identifies the documents which contain indexed terms and `OFFSET` identifies the location of those terms within each document.

A new storage preference, SEPARATE_OFFSETS, used in conjunction with BIG_IO, causes the DOCID and OFFSET to be stored in separate locations within the index.

The DOCID list is much shorter than the previous combined TOKEN_INFO data. It thus reduces the I/O necessary to perform single term queries, AND queries, and other queries where offset (that is, word position) information is not needed. Performance is improved for such queries. Queries that do require offset information (for example, phrase or near searches) may be slightly slower.

See Also:

Oracle Text Application Developer's Guide for details

2.11.2.5 Updatable SDATA Sections

SDATA sections may be updated using a new PL/SQL package, CTX_DDL.UPDATE_SDATA. This package updates the value of an SDATA item without requiring reindexing of all the data in that row. Additionally, the maximum number of SDATA sections has been increased from 32 to 99.

This feature provides better performance for rapidly mutating metadata. For example, if you want to include stock level in text queries, make it an SDATA section. The associated row might include a long data sheet of information that you do not want to reindex every time the stock level changes. With this new feature you can update only the SDATA part of the index.

See Also:

Oracle Text Reference for details

2.11.2.6 Add SDATA Section to Existing Index

SDATA sections can be added to an existing index without needing to completely rebuild the index. The new SDATA sections are indexed in all documents added or updated after this time. Previously indexed documents are not affected. Application flexibility and uptime is improved, as indexes can be modified to reflect new business requirements without having to rebuild the index from scratch.

See Also:

Oracle Text Reference for details

2.11.2.7 Ordering by SDATA Sections

Query templates now support ordering by one or more SDATA sections. This allows for more flexible application development and faster queries compared to a standard database sort.

See Also:

Oracle Text Application Developer's Guide for details

2.11.2.8 Increased Number of Field Sections

Previously, you were allowed only 64 field sections. You can now create an almost unlimited number (10,000+) of field sections. Field sections are more efficient than zone sections. Previously, some applications had to use zone sections since there were not enough field sections available. This feature improves the performance of such applications.

See Also:

Oracle Text Reference for details

2.11.2.9 Document-Level Lexer

The document-level Lexer allows users to define different Lexer and stoplist preferences to different documents in an index. This is an extension of the `MULTI_LEXER` and `MULTI_STOPLIST` features, but now the Lexer choice can be independent of language. This feature allows applications to be more flexible. Different types documents from different sources may have Lexer or stopword requirements which differ.

See Also:

Oracle Text Reference for details

2.11.2.10 Increased Number of MDATA Sections

The number of MDATA sections allowed is now effectively unlimited. The previous maximum was 100. This feature provides increased application flexibility. There is no longer a need to combine multiple MDATA fields into a single one.

See Also:

Oracle Text Reference for details

2.11.2.11 Language Identification

A new procedure called `POLICY_LANGUAGES` has been added to the `CTX_DOC` package. The procedure allows for the identification of the language of a section of text. Applications can identify the language of a document in order to process it in an appropriate manner (for example, to set a `LANGUAGE` metadata column).

See Also:

Oracle Text Reference for details

2.11.2.12 BIGRAM Mode for the Japanese VGRAM Lexer

Currently, with Japanese VGRAM lexer, certain Japanese queries require wildcard expansion which can be expensive. Oracle now provides a switch to Japanese VGRAM lexer to generate BIGRAM mode only and, therefore, eliminate the need for wildcard queries. The benefit of this feature is faster query performance on text indexed with the Japanese VGRAM lexer.

See Also:

Oracle Text Reference for details

2.11.2.13 Mild Not (MNOT) Operator

Mild Not (`MNOT`) is a new operator designed to find words which are not part of a phrase. For example, if you want to find references to the city of `York`, you might want to avoid finding it as part of the phrase `New York`. Excluding all documents containing `New York` does not solve this problem, since some documents might reference `York` and `New York`. The new `MNOT` operator makes such semantics possible. The `MNOT` operator improves precision and recall of searches by allowing searches for words, but excluding unwanted phrases containing those words.

See Also:

Oracle Text Reference for details

2.11.2.14 Forward Index

The forward index feature stores a tokenized and compressed version of the document in the Oracle Text index. This means that features such as highlighting and snippet generation no longer need to access, filter and tokenize the original document which is often an expensive process.

See Also:

Oracle Text Application Developer's Guide for details

2.11.2.15 NEAR Operator Enhancements

The `NEAR` operator has been improved to allow for nested `NEAR` operators, and to allow for `OR` constructs within the `NEAR` operator. These enhancements improve the flexibility for application creation.

See Also:

Oracle Text Reference for details

2.11.2.16 Pattern Stopclass

The pattern stopclass allows users to specify regular expressions, and any tokens matching those regular expressions are considered as stopwords. In other words, they are not indexed and are not considered significant in queries.

Unwanted strings, for example hexadecimal numbers or identifying codes, can be removed from the index to save space and improve performance.

See Also:

Oracle Text Reference for details

2.11.2.17 Session-Duration SQEs

Stored Query Expressions (SQE) are a way of saving frequently used query expressions. Session-duration SQEs are not permanently saved but exist only for the current session. This allows faster performance than permanent SQEs as they are stored in session memory which avoids the clutter that might occur when SQEs are frequently created for short-term use within an application.

See Also:

Oracle Text Reference for details

2.11.2.18 Query Filter Cache

A common scenario in text searching is that a particular set of criteria are used in many queries. For example, you might want to apply a security filter that restricts the results to only those appropriate to a particular user. The query filter cache feature allows you to cache the results of a particular query, or part of a query, then use those results to filter future searches. Conceptually, this is similar to using a Stored Query Expression (SQE) but it provides better performance.

This feature provides better performance for queries that have components shared with other queries.

See Also:

Oracle Text Reference for details

2.11.2.19 Snippet Support in Result Set Interface

The result set interface in Oracle Database 11*g* was able to produce the various kinds of data needed for a page of search results all at once, improving performance by sharing overhead. The result set interface could also return data views which were difficult to express in SQL, such as top *n* by category queries.

To present snippet information along with search results to the end user, multiple iterations were required in Oracle Database 11*g*. With the approach in Oracle Database 11*g*, it was necessary to do a search query and iterate through the result to retrieve snippet information on each row.

In Oracle Database 12*c* Release 1 (12.1), native support of snippet information from the result set interface resolves the Oracle Database 11*g* issues. The result set descriptor only needs `SNIPPET` defined if it is required. If defined, the user retrieves the snippet in the result set along with other search results.

This support provides faster, more flexible applications based on the result set interface.

See Also:

Oracle Text Application Developer's Guide for details

2.11.3 Oracle XML Enhancements

The following sections describe enhancements to Oracle XML.

2.11.3.1 Enhanced ANYDATA Support

Restrictions have been removed from the `ANYDATA` implementation that prevented its use with Abstract Data Types (ADT) that contained attributes whose data type was `LOB` or `XMLType`. This enhancement increases the flexibility of the Oracle `ANYDATA` implementation and it can now be used with database editions.

See Also:

Oracle Database SQL Language Reference for details

2.11.3.2 Integrated Oracle XQuery Implementation

This feature unifies the Oracle and BEA XQuery engines creating a single Java-based XQuery engine, supporting the XQuery 1.0 recommendation that can be used to leverage the XQuery language outside of Oracle Database. It also adds support for XQuery API for Java (XQJ) as an API which is the Java Specification Request (JSR) for executing XQuery statements from Java programs.

This feature allows customers to leverage the benefits of the World Wide Web Consortium (W3C) XQuery language by consolidating the existing Oracle and WebLogic XQuery engines into a single engine that combines developer productivity with highly scalable and performant XQuery processing.

The new XQuery engine delivers support for the latest XQuery standard modules and XQuery update. It improves developer productivity by providing consistency with the XQuery engine used in Oracle Database, support for XQuery debugging, and support for XQJ.

The new engine is able to leverage other Oracle XML technology, including scalable Document Object Model (DOM) processing and the new binary XML formats used by Oracle Database and Oracle XML Developer's Kit. The XQuery engine is capable of processing very large documents and very large numbers of concurrent operations.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

2.11.3.3 Oracle XDK/J DOM Improvements

This feature adds support for W3C DOM Level 3 Core API's and reduces the memory footprint associated with using XML schemas.

These improvements allow developer's to leverage the benefits of the latest API's used for XML processing, as defined by the W3C, including those defined as part of the DOM Level 3.0 Core specification. This results in improved performance and scalability of the Oracle XDK/J DOM implementation by reducing the memory footprint of the DOM and improved support for Oracle's Scalable DOM (SDOM).

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

2.11.3.4 Domain Index Support of Hash Partitioned Tables

Applications that use domain indexes can now use the hash partitioning method. Oracle XML DB now has support for hash partitioning. You can create a locally partitioned `XMLindex` index on `XMLType` tables and columns that you have partitioned using hash partitioning (in addition to range and list partitioning). Hash partitioning is an effective approach to balancing I/O evenly over a series of partitions.

See Also:

Oracle Database Data Cartridge Developer's Guide and *Oracle XML DB Developer's Guide* for details

2.11.3.5 Oracle XSLT or XPath Engine Interoperability

This feature enables the use of non-XDK-based data models with the Oracle XDK/J, XSLT, or XPath engine, which supports interoperability between these Oracle engines and third-party XML processors.

Interoperability between the Oracle XDK/J, XSLT, and XPath engines and third-party XML processors is enabled.

2.11.3.6 Programmatically Create and Manipulate Scalable DOM

In previous releases, a scalable Document Object Model (DOM) could only be created by Oracle's XML parser. This meant that a scalable DOM could only be created from an existing XML document. This feature removes this limitation by allowing developers to programmatically create and manipulate a new XML document, based on scalable DOM techniques, using standard DOM API's.

This feature allows developers to create and manipulate very large XML documents programmatically by creating an instance of Oracle's scalable DOM, rather than a traditional in-memory DOM. The scalable DOM can then be manipulated using the standard DOM API's provided by Oracle's XDK/J DOM implementation.

2.11.3.7 Standalone XQuery Virtual Machine

The full capabilities of the Oracle XQuery Virtual Machine can be accessed using a standalone application. This allows XQuery expressions to be performed directly from the command line without interacting with Oracle Database.

It also enhances the Oracle XQuery Virtual Machine to add support for the XQuery Update standard as well as the emerging XQuery scripting language. The Oracle XQuery Virtual Machine and database can also share the same native XML format, allowing the Oracle XQuery Virtual Machine to process XML from the database without having to incur the overhead of serializing and parsing the XML in question.

The Oracle XQuery Virtual Machine is a powerful XQuery processor currently only available as part of Oracle Database. Enabling a standalone command-line mode allows the Oracle XQuery Virtual Machine to be used to execute XQuery operations in situations when running XQuery inside the database is not appropriate.

2.11.3.8 Support XQuery Full-Text Specification

This feature extends Oracle's support for the W3C XQuery specification by adding support for the XQuery full text extension. This enables customers to perform XML-aware full text searches on XML content stored in the database.

> **See Also:**
>
> *Oracle XML DB Developer's Guide* for details

2.11.3.9 XDK/J Support for Fast Infoset

This feature adds support for the Fast Infoset to XDK/J model, enabling developers to use Fast Infoset techniques while working with XML content in Java.

Fast Infoset provides the following benefits in comparison with other formats:

- It is more compact, parses faster, and serializes better than XML documents.
- It parses five times faster than the Xerces parser, is three times faster at serializing, and Fast Infoset documents are generally 20 to 60 percent smaller than the corresponding XML documents.
- It leads other binary XML formats in performance and ration of compression, and handles small to large documents in a more balanced manner.

> **See Also:**
>
> *Oracle XML Developer's Kit Programmer's Guide* for details

2.11.3.10 XmlDiff Support for XDK Java

This feature adds support for a Java-based XmlDiff that is format compatible with the existing C and PL/SQL XmlDiff capabilities introduced in Oracle Database 11*g* Release 1 (11.1).

This feature enables mid-tier programs written in pure Java to exchange XmlDiff output with programs written in C or programs which use Oracle Database to perform XmlDiff operations.

> **See Also:**
>
> *Oracle XML Developer's Kit Programmer's Guide* for details

2.11.3.11 Support for XQuery Update

Support has been added for the XQuery update recommendation, allowing users to perform fragment and node-level updates using the W3C standard query language.

This support allows users to perform fragment-level updates on XML content managed by Oracle XML DB in a performant and standards-based manner. This support also enables XML-based applications, that have been written using XQuery update syntax, to be ported to Oracle Database.

This feature improves developer productivity by replacing Oracle's XPath 1.0 based DML operators with a simpler standards-based approach that leverages the full benefits of XQuery.

> **See Also:**
>
> *Oracle XML DB Developer's Guide* for details

2.11.4 Oracle XML Repository Enhancements

The following sections describe the Oracle XML repository enhancements.

2.11.4.1 Enable Digest Authentication with Oracle Database HTTP Listener

Digest authentication uses an industry-standard mechanism that lets the client and server exchange authentication tokens without passwords being transmitted in plain text. This mechanism reduces the likelihood of passwords being compromised during transmission.

Digest authentication support of Oracle XML DB ensures that the Oracle XML DB HTTP server remains compatible with Microsoft Web Folders WebDAV client.

This feature enhances database security by adding support for digest authentication. Digest authentication is an industry-standard protocol commonly used with the HTTP protocol, and is supported by most HTTP clients. Digest authentication ensures that passwords are always transmitted in a secure manner, even when an encrypted (HTTPS) connection is not in use. Support for digest authentication allows organizations to deploy applications that leverage the Oracle XML DB HTTP without having to worry about passwords being compromised.

> **See Also:**
>
> *Oracle XML DB Developer's Guide* for details

2.11.4.2 WebDAV, HTTP, and FTP Access for DBFS

This feature provides WebDAV, HTTP, and FTP access to Database File System (DBFS) by extending Oracle XML DB support to DBFS. Files stored in a DBFS file system can now be edited and managed collaboratively over the World Wide Web, extending file system-like access to DBFS file systems on non-Linux platforms.

> **See Also:**
>
> *Oracle Database SecureFiles and Large Objects Developer's Guide* for details

2.12 Upgrades

The following sections describe the upgrade enhancements for Oracle Database 12*c* Release 1 (12.1).

2.12.1 General

The following sections describe general upgrade features and enhancements.

2.12.1.1 Enhanced Upgrade Automation

Database upgrade has been enhanced for better ease-of-use by improving the amount of automation applied to the upgrade process. Additional validation steps have been added to the pre-upgrade phase in both the command-line pre-upgrade script and the Database Upgrade Assistant (DBUA). In addition, the pre-upgrade validation steps have been enhanced with the ability to generate a fix-up script to resolve most issues that may be identified before the upgrade.

Post-upgrade steps have also been enhanced to reduce the amount of manual work required for a database upgrade. The post-upgrade status script gives more explicit guidance about the success of the upgrade on a component-by-component basis. Post-upgrade fix-up scripts are also generated to automate tasks that must be performed after the upgrade.

See Also:

Oracle Database Upgrade Guide for details

2.12.1.2 Parallel Upgrade

The database upgrade scripts can now take advantage of multiple CPU cores by using parallel processing to speed up the upgrade process. This results in less downtime due to a database upgrade, and thus improved database availability.

See Also:

Oracle Database Upgrade Guide for details

2.13 Windows

The following sections describe the new Windows features for Oracle Database 12*c* Release 1 (12.1).

2.13.1 Windows Security Enhancements

The following sections describe Windows security integration for Oracle Database on Windows.

2.13.1.1 Support for Oracle Home User on Windows

Starting with Oracle Database 12*c* Release 1 (12.1), Oracle Database supports the use of Oracle home user, specified at the time of installation. Oracle home user is the owner of Oracle services that run from Oracle home and cannot be changed post installation. On a system, different Oracle homes can share the same Oracle home user or use different Oracle home user names.

Oracle home user can be a Windows built-in account or a standard Windows user account (not an Administrator account). This account is used for running the Windows services for the Oracle home. For a database server installation, Oracle recommends that you use a standard Windows user account (instead of a Windows built-in account) as the Oracle home user for enhanced security.

For Oracle RAC Database, the Oracle home user must be a Windows domain user account and must be an existing Windows account.

See Also:

Oracle Database Platform Guide for Microsoft Windows for details

2.13.1.2 Support of Oracle Home User for Oracle Net Services

Oracle Database 12*c* supports Oracle Net Services such as Oracle Net Listener, Oracle Connection Manager Administration (CMADMIN), and Oracle Connection Manager (CMAN) proxy listener to run under an Oracle home user account specified during Oracle Database installation. In earlier releases, Oracle Net Services ran under the highly privileged, built-in Local System account (LSA). This feature provides more control over security.

See Also:

Oracle Database Net Services Administrator's Guide for details

2.13.1.3 Named User Support for Oracle RAC Services on Windows

Windows service can be run using a built-in account or a named user on Windows operating systems. Oracle RAC supports running various Oracle RAC services as different users, while sharing the same Oracle Grid Infrastructure environment.

Using a named user for services allows for more flexibility when creating Oracle RAC environments. It also enables Oracle RAC based consolidation ensuring a separation of duty as required.

See Also:

Oracle Clusterware Administration and Deployment Guide for details

3

Oracle Database 12*c* Release 1 (12.1) Documentation

This chapter lists the books available online that document Oracle® Database 12*c* Release 1 (12.1).

3.1 Documentation Titles and Online Documentation Structure

All books are available in electronic format.

The following categories of documentation are available:

- Data Mining Documentation
- Gateways Documentation
- Installation Documentation
- Java Development Documentation
- Linux x86 Documentation
- Microsoft Windows Documentation
- Oracle® Multimedia 12c Documentation
- OLAP Documentation
- Release Notes Documentation
- Server Documentation

  Subcategories include:

  - Administration, Concepts, and Reference
  - Application Development
  - Application Express
  - Backup and Recovery
  - Data Warehousing
  - Language and Interface
  - Networking and Security
  - Oracle Real Application Clusters
  - Oracle Text
  - R Enterprise

  - Streams and Replication
- Spatial Documentation
- TimesTen In-Memory Database Documentation

3.1.1 Data Mining Documentation

Table 3–1 Data Mining Documentation

Documentation	Part Number	Description
Oracle Data Mining Concepts	E17692	Discusses the basic concepts underlying Oracle Data Mining. Includes an overview of the features of Oracle Data Mining and information about mining functions and algorithms.
Oracle Data Mining User's Guide	E53115	Explains how to use the SQL interface to Oracle Data Mining to create models and score data. This Guide also addresses administrative issues such as security, import/export, and upgrade for Oracle Data Mining.

3.1.2 Gateways Documentation

Table 3–2 Gateways Documentation

Documentation	Part Number	Description
Oracle Database Gateway Installation and Configuration Guide for Microsoft Windows	E17926	Describes how to install Oracle Database Gateway for Sybase, Informix, Teradata, SQL Server, ODBC, and DRDA on Microsoft Windows (32-bit) and Microsoft Windows (64-bit) platform.
Oracle Database Gateway Installation and Configuration Guide for IBM AIX on POWER Systems (64-Bit), Linux x86-64, Oracle Solaris on SPARC (64-Bit), Oracle Solaris on x86-64 (64-Bit), and HP-UX Itanium	E17927	Describes how to install and configure Oracle Database Gateway for Sybase, Informix, Teradata, SQL Server, ODBC, and DRDA on UNIX-based platforms.
Oracle Database Gateway for APPC Installation and Configuration Guide for Microsoft Windows	E17928	Provides Oracle applications with seamless access to IBM mainframe data and services through Remote Procedure Call (RPC) processing.
Oracle Database Gateway for APPC Installation and Configuration Guide for IBM AIX on POWER Systems (64-Bit), Linux x86-64, Oracle Solaris on SPARC (64-Bit), and HP-UX Itanium	E17929	Provides Oracle applications with seamless access to IBM mainframe data and services through Remote Procedure Call (RPC) processing.
Oracle Database Gateway for WebSphere MQ Installation and User's Guide	E17930	Provides access to WebSphere MQ services. This gateway requires a system that is capable of running 64-bit applications.

Table 3–2 (Cont.) Gateways Documentation

Documentation	Part Number	Description
Oracle Database Gateway for APPC User's Guide	E17931	Provides Oracle applications seamless access to virtually any APPC-enabled system, including IBM mainframe data and services through Remote Procedure Call (RPC) processing.
Oracle Database Gateway for Sybase User's Guide	E17932	Describes the Oracle Database Gateway for Sybase, which enables Oracle client applications to access Sybase data through Structured Query Language (SQL).
Oracle Database Gateway for Informix User's Guide	E17933	Describes the Oracle Database Gateway for Informix, which enables Oracle client applications to access Informix data through Structured Query Language (SQL).
Oracle Database Gateway for Teradata User's Guide	E17934	Describes the Oracle Database Gateway for Teradata, which enables Oracle client applications to access Teradata data through Structured Query Language (SQL).
Oracle Database Gateway for SQL Server User's Guide	E17935	Describes the Oracle Database Gateway for SQL Server, which enables Oracle client applications to access SQL Server data through Structured Query Language (SQL).
Oracle Database Gateway for ODBC User's Guide	E17936	Describes the Oracle Database Gateway for ODBC, which enables Oracle client applications to access non-Oracle systems data through Structured Query Language (SQL).
Oracle Database Gateway for DRDA User's Guide	E17937	Provides users with transparent access to DB2.

3.1.3 Installation Documentation

Table 3–3 Installation Documentation

Documentation	Part Number	Description
Oracle Database SQL Translation and Migration Guide	E22508	Describes the installation, configuration, and administration tasks for all activities related to migrating applications to Oracle Database from other databases, including DB2, MySQL, and Sybase.
Oracle Database Installation Guide for Linux	E41491	Provides configuration information for network and system administrators, and database installation information for database administrators (DBAs) who install and configure Oracle Database and Oracle Grid Infrastructure for a standalone server on Linux.
Oracle Database Migration Assistant for Unicode Guide	E48475	Introduces you to the Oracle Database Migration Assistant for Unicode, discusses the database character set migration process, and verifies that there are no character set data problems in a current Unicode database.

Table 3–3 (Cont.) Installation Documentation

Documentation	Part Number	Description
Oracle Grid Infrastructure Installation Guide for Linux	E48914	Describes how to install Grid Infrastructure for a cluster software, which includes Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM), on systems running Linux.
Oracle Database Client Installation Guide for Microsoft Windows	E49462	Describes how to install and configure Oracle Database Client on Windows.
Oracle Database Installation Guide for Oracle Solaris	E50818	Provides configuration information for network and system administrators, and database installation information for database administrators (DBAs) who install and configure Oracle Database, and Oracle Grid Infrastructure for a standalone server on Oracle Solaris.
Oracle Grid Infrastructure Installation Guide for Oracle Solaris	E51120	Describes how to install Oracle Grid Infrastructure for a cluster software, which includes Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM), on systems running Oracle Solaris.
Oracle Database Quick Installation Guide for Linux x86-64	E54543	Describes how to quickly install Oracle Database on Linux x86-64 systems.
Oracle Database Client Installation Guide for Linux	E55097	Describes how to install and configure Oracle Database client on Linux.
Oracle Database Client Quick Installation Guide for Linux x86-64	E55098	Describes how to quickly install Oracle Database Client on Linux x86-64 systems.
Oracle Database Client Installation Guide for Oracle Solaris	E55099	Describes how to install and configure Oracle Database Client on Oracle Solaris.
Oracle Database Quick Installation Guide for Oracle Solaris on SPARC (64-Bit)	E55100	Describes how to quickly install Oracle Database on Oracle Solaris on SPARC (64-Bit).
Oracle Database Client Quick Installation Guide for Oracle Solaris on SPARC (64-Bit)	E55101	Describes how to quickly install Oracle Database Client on Oracle Solaris on SPARC (64-Bit).
Oracle Database Quick Installation Guide for Oracle Solaris on x86-64 (64-Bit)	E55102	Describes how to quickly install Oracle Database on Oracle Solaris on x86-64 (64-Bit).
Oracle Database Client Quick Installation Guide for Oracle Solaris on x86-64 (64-Bit)	E55103	Describes how to quickly install Oracle Database on Oracle Solaris on x86-64 (64-Bit).
Oracle Database Examples Installation Guide	E55464	Describes how to install and configure the products available on the Oracle Database Examples installation media.
Oracle Database Client Quick Installation Guide for IBM AIX on POWER Systems (64-Bit)	E56114	Describes how to quickly install Oracle Database Client on IBM AIX on POWER Systems (64-Bit).

Table 3–3 (Cont.) Installation Documentation

Documentation	Part Number	Description
Oracle Database Client Installation Guide for IBM AIX on POWER Systems (64-Bit)	E56492	Describes how to install and configure Oracle Database Client on IBM AIX on POWER Systems (64-Bit).
Oracle Database Quick Installation Guide for Microsoft Windows x64 (64-Bit)	E56576	Describes how to quickly install Oracle Database on Windows x64 systems.
Oracle Database Client Quick Installation Guide for Microsoft Windows x64 (64-Bit)	E56585	Describes how to quickly install Oracle Database Client on Windows x64 systems.
Oracle Database Quick Installation Guide for IBM AIX on POWER Systems (64-Bit)	E56891	Describes how to quickly install Oracle Database on IBM AIX on POWER Systems (64-Bit).
Oracle Database Client Installation Guide for HP-UX Itanium	E56892	Provides configuration information for network and system administrators, and database installation information for database administrators (DBAs) who install and configure Oracle Database, and Oracle Grid Infrastructure for a standalone server on HP-UX systems.
Oracle Database Quick Installation Guide for HP-UX Itanium	E56893	Describes how to quickly install Oracle Database on HP-UX Itanium systems.
Oracle Database Client Quick Installation Guide for HP-UX Itanium	E56894	Describes how to quickly install Oracle Database Client on HP-UX Itanium.
Oracle Database Quick Installation Guide for IBM: Linux on System z	E56895	Describes how to quickly install Oracle Database on Z/Linux systems.
Oracle Database Client Quick Installation Guide for IBM: Linux on System z	E56896	Describes how to quickly install Oracle Database Client on Z/Linux systems.
Oracle Database Client Quick Installation Guide for Microsoft Windows (32-Bit)	E56927	Describes how to quickly install Oracle Database Client on Windows 32-bit systems.

3.1.4 Java Development Documentation

Table 3–4 Java Development Documentation

Documentation	Part Number	Description
Oracle Database Advanced Queuing Java API Reference	E15980	Describes the Java APIs for application development and integration using the Oracle messaging system, Advanced Queuing.
Oracle Database XML Java API Reference	E15981	Describes Java APIs for XML.
Oracle Universal Connection Pool for JDBC Java API Reference	E17662	Provides Java API reference for Universal Connection Pool for JDBC.
Oracle Database JDBC Java API Reference	E17663	Provides Java API reference for Java Database connectivity.
Oracle Database RAC FAN Events Java API Reference	E17664	Provides easy access to Oracle Fast Application Notification (FAN) events.
Oracle Database XStream Java API Reference	E17774	Describes the Java APIs for XStream.
Oracle Database 2 Day + Java Developer's Guide	E17944	Guides application developers on how to use Java to access and modify data in Oracle Database.
Oracle Globalization Development Kit Java API Reference	E37513	Describes APIs for the Oracle Globalization Development Kit.
Oracle Database JDBC Developer's Guide	E49300	Describes how to use Oracle JDBC drivers to develop powerful Java database applications.
Oracle Universal Connection Pool for JDBC Developer's Guide	E49541	Provides instructions for using Oracle Universal Connection Pooling API. The API is JDBC driver agnostic.
Oracle Database Java Developer's Guide	E50793	Describes how to develop, load, and run Java applications in Oracle Database.

3.1.5 Linux x86 Documentation

Table 3–5 Linux x86 Documentation

Documentation	Part Number	Description
Oracle Database Provider for DRDA User's Guide for Linux x86-64	E55482	Describes how customers who have a large investment in DB2-based applications can migrate to Oracle Database while keeping these applications largely unchanged.

3.1.6 Microsoft Windows Documentation

Table 3–6 Microsoft Windows Documentation

Documentation	Part Number	Description
Oracle Database Platform Guide for Microsoft Windows	E10714	Provides platform-specific information for configuring and using Oracle Database software on the Windows platform. Topics discussed include: architecture, services, tools, configuration, administration, tuning, backup, accounts, passwords, user authentication, registry parameters, and application development.
Oracle Data Provider for .NET Developer's Guide for Microsoft Windows	E17732	Describes provider-specific features and properties of ODP.NET. Oracle Data Provider for .NET is an implementation of Microsoft's ADO.NET interface and provides access to Oracle databases.
Oracle Providers for ASP.NET Developer's Guide for Microsoft Windows	E17733	Provides introductory, installation, postinstallation configuration, and usage information for using these providers with ASP.NET services and controls.
Oracle Services for Microsoft Transaction Server Developer's Guide for Microsoft Windows	E18595	Describes Oracle Services for Microsoft Transaction Server, which allows Oracle databases to be used as resource managers in distributed transactions. The guide describes its usage and configuration in a Microsoft Distributed Transaction Coordinator environment.
Oracle Database Extensions for .NET Developer's Guide for Microsoft Windows	E18596	Provides introductory, installation, post-installation configuration, and usage information for Oracle Database Extensions for .NET, which makes it possible to build and run .NET stored procedures or functions with Oracle Database for Microsoft Windows.

3.1.7 Oracle® Multimedia 12c Documentation

Table 3–7 Oracle® Multimedia Documentation

Documentation	Part Number	Description
Oracle Multimedia Reference	E17696	Presents reference information about using Oracle Database to store, manage, and retrieve images, audio, video, DICOM format medical images and other objects, or other heterogeneous media data in an integrated fashion with other enterprise information. Oracle Multimedia extends Oracle Database reliability, availability, and data management to multimedia content in Internet, electronic commerce, medical, financial, and other media-rich applications.
Oracle Multimedia User's Guide	E17697	Presents information about using Oracle Database to store, manage, and retrieve images, audio, video, DICOM format medical images and other objects, or other heterogeneous media data in an integrated fashion with other enterprise information. Oracle Multimedia extends Oracle Database reliability, availability, and data management to multimedia content in traditional, Internet, electronic commerce, medical, financial, and other media-rich applications.
Oracle Multimedia DICOM Developer's Guide	E17698	Presents reference and other information about using Oracle Database to store, manage, and retrieve DICOM content such as single-frame and multiframe images, waveforms, slices of 3-D volumes, video segments, and structured reports in an integrated fashion with other enterprise information. Oracle Multimedia DICOM extends Oracle Database reliability, availability, and data management to media objects in medical applications. Oracle Multimedia DICOM supports Digital Imaging and Communications in Medicine, the standard for medical images.
Oracle Multimedia Java API Reference	E17699	Presents reference information about writing Java applications using Oracle Multimedia objects to manage multimedia content within Oracle Database.

Table 3–7 (Cont.) Oracle® Multimedia Documentation

Documentation	Part Number	Description
Oracle Multimedia DICOM Java API Reference	E17700	Presents reference information about writing Java applications using the Oracle Multimedia object designed to store Digital Imaging and Communications in Medicine (DICOM) data within Oracle Database.
Oracle Multimedia Mid-Tier Java API Reference	E17701	Presents reference information about writing Java applications using Oracle Multimedia to extract Digital Imaging and Communications in Medicine (DICOM) metadata outside of the database, either on the client or in the middle tier.
Oracle Multimedia Servlets and JSP Java API Reference	E28184	Presents reference information about writing Java applications using Java servlets and JavaServer Pages (JSP) to retrieve and upload multimedia data from and to Oracle Database.

3.1.8 OLAP Documentation

Table 3–8 OLAP Documentation

Documentation	Part Number	Description
Oracle OLAP DML Reference	E17673	Contains a complete description of the OLAP Data Manipulation Language (OLAP DML) used to define and manipulate analytic workspace objects.
Oracle OLAP Java API Developer's Guide	E17707	Provides guidance for developing client applications using the Oracle OLAP Java API.
Oracle OLAP Java API Reference	E17708	Provides reference information for the Oracle OLAP Java API.
Oracle OLAP Customizing Analytic Workspace Manager	E17709	Describes the Analytic Workspace Manager XML and Java plug-in interfaces and provides simple examples of implementations of them.
Oracle OLAP Expression Syntax Reference	E23382	Describes the Oracle OLAP expression syntax used to generate information-rich content from the data stored in dimensional database objects such as cubes and cube dimensions.
Oracle OLAP User's Guide	E53237	Explains how SQL applications can extend their analytic processing capabilities and manage summary data by using the OLAP option of Oracle Database.

3.1.9 Release Notes Documentation

Table 3–9 Release Notes Documentation

Documentation	Part Number	Description
*Pro*COBOL Release Notes*	E18407	Contains important information not included in the Pro*COBOL Programmer's Guide.
*Pro*C/C++ Release Notes*	E18409	Contains important information not included in the Pro*C/C++ Programmer's Guide.
Oracle Database Release Notes for Linux	E51263	Contains important information not included in the Oracle Database documentation for Linux.
Oracle Database Migration Assistant for Unicode Release Notes	E51606	Contains late-breaking information for Oracle Database Migration Assistant for Unicode.
Oracle Database Release Notes for Oracle Solaris	E51863	Contains important information not included in Oracle Database documentation for Solaris.
Oracle Database ODBC Driver Release Notes	E53668	Describes new features, functionality, platforms supported, software required, server requirements, hardware required, testing matrix, UNIX platform certification matrix, post-installation, known issues, and software problems fixed.

3.1.10 Server Documentation

Table 3–10 Server Documentation

Documentation	Part Number	Description
Administration, Concepts, and Reference		
Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems	E10638	Explains how to configure and manage Oracle Database on Linux and UNIX-based operating systems.
Oracle Database XML C API Reference	E15885	Describes Oracle XML Developer's Kits (XDK) and Oracle XML DB APIs for the C programming language. It primarily lists the syntax of functions, methods, and procedures associated with these APIs.
Oracle Database XML C++ API Reference	E15977	Describes Oracle XML Developer's Kits (XDK) and Oracle XML DB APIs for the C++ programming language. This manual primarily lists the syntax of functions, methods, and procedures associated with these APIs.
Oracle Database Sample Schemas	E15979	Describes the sample database schemas available with Oracle Database and used in some Oracle documentation and curriculum examples.

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Database Heterogeneous Connectivity User's Guide	E17907	Describes Oracle's approach to information integration in a heterogeneous environment. Describes Oracle Database Gateways and is a user's guide for these Oracle products.
Oracle Database Advanced Queuing User's Guide	E17945	Describes features of application development and integration using Oracle Database Advanced Queuing (AQ). This information applies to versions of the Oracle Database server that run on all platforms, unless otherwise specified.
*SQL*Plus Release Notes*	E18402	Contains important information not included in the SQL*Plus User's Guide and Reference.
*SQL*Plus Quick Reference*	E18403	Contains a complete list of SQL*Plus command syntax.
*SQL*Plus User's Guide and Reference*	E18404	Contains a complete description of the SQL*Plus command-line interface to Oracle Database, and a complete reference to SQL*Plus commands and syntax.
Oracle Database Global Data Services Concepts and Administration Guide	E22100	Provides dynamic load balancing, failover, and centralized service management for a set of replicated databases that offer common services. The set of databases can include Oracle Real Application Clusters (Oracle RAC) and noncluster Oracle databases interconnected through Oracle Data Guard, or any other replication technology.
Oracle Database High Availability Best Practices	E40019	Provides high availability best practices for configuring and maintaining your Oracle Database system and network components.
Oracle Database VLDB and Partitioning Guide	E41057	Contains an overview of very large database (VLDB) topics, with emphasis on partitioning and parallel execution as a key component of the VLDB strategy.
Oracle Automatic Storage Management Administrator's Guide	E41058	Describes how to administer Oracle Automatic Storage Management (Oracle ASM) for Oracle Database.
Oracle Database SQL Language Reference	E41329	Contains a complete description of Oracle SQL, the language used to manage information in Oracle Database.
Oracle Database Concepts	E41396	Describes Oracle Database architecture and essential topics for database administrators and developers.
Oracle Database Upgrade Guide	E41397	Guides you through the process of planning and performing upgrades for Oracle Database. This book also provides information about compatibility, upgrading applications, and changes in the new release that affect upgrading Oracle Database.

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Database Administrator's Guide	E41484	Describes how to create, configure, and administer an Oracle database. Includes sections on Oracle Scheduler, Oracle Database Resource Manager, Oracle Restart, Pluggable Databases, and distributed computing.
Oracle Database Reference	E41527	Provides reference information about database initialization parameters, static data dictionary views, and dynamic performance views that are part of Oracle Database.
Oracle Database Utilities	E41528	Describes how to use Oracle Database utilities to load data into a database, transfer data between databases, and maintain data. The topics discussed include Data Pump Export, Data Pump Import, SQL*Loader, external tables and associated access drivers, the Automatic Diagnostic Repository Command Interpreter (ADRCI), DBVERIFY, DBNEWID, LogMiner, the Metadata API, original Export, and original Import.
Oracle Database Globalization Support Guide	E41669	Describes Oracle globalization support for the database. Explains how to set up a globalization support environment, choose and migrate a character set, customize locale data, do linguistic sorting, program in a global environment, and program with Unicode.
Oracle Database PL/SQL Packages and Types Reference	E41829	Describes the PL/SQL packages provided with the Oracle database server that extend database functionality and provide PL/SQL access to SQL features.
Oracle Database Quality of Service Management User's Guide	E49047	Describes the concepts of Oracle Database Quality of Service Management (Oracle Database QoS Management), and explains how to configure, administer, and troubleshoot Oracle Database QoS Management.
Oracle Database Performance Tuning Guide	E49058	Provides information about how to tune the performance of Oracle Database using Oracle performance tools.
Oracle Database High Availability Overview	E49097	Introduces Oracle MAA solutions for deploying a highly available environment, helps you determine your availability requirements, and describes the database architectures that can help your business achieve high availability.
Oracle Database Vault Administrator's Guide	E49109	Explains how to configure an Oracle database to use Oracle Database Vault, which enables users to restrict and monitor administrative user access.
Oracle Database SQL Language Quick Reference	E49206	Contains a high-level text description of the SQL syntax used to manage information in Oracle Database.

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Database Licensing Information	E49208	Describes Oracle Database licensing. If you have a question about your licensing needs, contact your Oracle sales representative, or refer to the resources listed in "Related Documents" for more information.
Oracle Database Readme	E49321	Describes important last-minute features and changes not included in the Oracle Database Documentation Library for 12c Release 1 (12.1)
Oracle Database New Features Guide	E49322	Oracle Database New Features Guide is addressed to those of you familiar with previous releases of Oracle Database and who would like to become familiar with features, options, and enhancements that are new in this release of the database.
Oracle Database Error Messages Reference	E49325	Describes error messages that may appear while using products that are part of Oracle Database.
Oracle Database JPublisher User's Guide	E49873	Describes the JPublisher utility useful for Java programmers who want classes in their applications to correspond to SQL or PL/SQL entities or server-side Java classes.
Oracle Database PL/SQL Language Reference	E50727	Presents PL/SQL, the Oracle procedural extension of SQL, an advanced fourth-generation programming language. Explains the concepts behind PL/SQL and illustrates every facet of the language.
Oracle Database 2 Day DBA	E51671	Provides a single source reference for administering Oracle Database in a small to midsize environment, using the graphical user interface of Oracle Enterprise Manager Database Express. If you are familiar with computers, but unfamiliar with administering Oracle Database, and your computer has four CPUs or fewer, and you do not expect to support more than 500 users, this book is appropriate for you.
Oracle Database Testing Guide	E55028	Provides information about how to assure the integrity of database changes and manage test data using Oracle Real Application Testing, data masking, and data subsetting.
Oracle Database 2 Day + Performance Tuning Guide	E55047	Explains how to perform day-to-day database performance tuning tasks using features provided by Oracle Diagnostics Pack, Oracle Tuning Pack, and Oracle Enterprise Manager Cloud Control.
Application Development		
Oracle Database Data Cartridge Developer's Guide	E15882	Describes how to implement custom indexing and query optimization services and how to package and use these as a server extension called a data cartridge.

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Database SecureFiles and Large Objects Developer's Guide	E17605	Describes use of SecureFiles and large object data types BLOB, CLOB, NCLOB, and BFILE in application development. Presents APIs for working with LOBs in supported programmatic environments. Describes database file systems and the Oracle Database File System Hierarchical Store package.
Oracle Database SQLJ Developer's Guide	E17660	Provides guidance to developers on SQLJ programming. This document can be read by anyone with an interest in SQLJ programming, but assumes at least some prior knowledge of Java, SQLJ, PL/SQL, and Oracle database.
Oracle Database 2 Day + PHP Developer's Guide	E18554	Provides a tutorial on how to develop PHP scripts that use the database languages SQL and PL/SQL to access and manipulate Oracle data.
Oracle SQL Developer User's Guide	E35117	Provides conceptual and usage information about Oracle SQL Developer, a graphical tool that enables you to browse, create, edit, and delete (drop) database objects; run SQL statements and scripts; edit and debug PL/SQL code; manipulate and export data; migrate third-party databases to Oracle; view metadata and data in third-party databases; and view and create reports.Note: This book is for an old release of SQL Developer. You are encouraged to use the latest available release. See http://www.oracle.com/technetwork/developer-tools/sql-developer/ for information about downloads and documentation.
Oracle XML DB Developer's Guide	E41152	This manual describes Oracle XML DB. It includes guidelines and examples for storing, generating, accessing, searching, validating, transforming, evolving, and indexing XML data in Oracle Database.
Oracle Database Development Guide	E41452	Provides usage information for developers (all experience levels) of databases and database applications.
Oracle Database SQL Tuning Guide	E49106	Describes how to perform SQL tuning, which is the iterative process of improving SQL statement performance to meet specific, measurable, and achievable goals.
Oracle Database Workspace Manager Developer's Guide	E49170	Provides usage and reference information about Oracle Workspace Manager, which enables applications to create workspaces and group different versions of table row values in different workspaces.

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Database 2 Day Developer's Guide	E53088	Explains general concepts behind development with Oracle Database, introduces basic features of SQL and PL/SQL, provides references to in-depth information elsewhere in Oracle Database library, and shows how to create a simple application.
Oracle Database Object-Relational Developer's Guide	E53277	Explains how to use the object-relational features of Oracle Database.
Oracle XML Developer's Kit Programmer's Guide	E54405	Provides information to application developers who need to use components of the Oracle XML Developer's Kit (XDK) to generate and store XML data in a database or in a document outside the database.
Application Express		
Oracle Application Express Release Notes	E17957	Contains important information not included in the documentation for Oracle Application Express.
Oracle Application Express Installation Guide	E17958	Provides an overview of installing Oracle Application Express and describes issues to consider before installing.
Oracle Database 2 Day + Application Express Developer's Guide	E17960	Introduces you to application development using Oracle Application Express. This manual shows you how to set up a development environment, or access a hosted demonstration environment. It then walks you through building an initial application, modifying it, and previewing it.
Oracle Application Express Application Builder User's Guide	E17961	Describes how to use the Oracle Application Express development environment to build and deploy database-centric Web applications.
Oracle Application Express Application Migration Guide	E17962	Describes how to convert applications from either Microsoft Access or Oracle Forms to Oracle Application Express.
Oracle Application Express API Reference	E17963	Describes the Application Programming Interfaces, referred to as APIs, available when programming in the Oracle Application Express environment.
Oracle Application Express SQL Workshop Guide	E17964	Describes how to use Oracle Application Express SQL Workshop and utilities to load and unload data from an Oracle database, generate DDL, view object reports, and restore dropped database objects.
Oracle Application Express Administration Guide	E17965	Describes how to perform administration tasks for an Oracle Application Express workspace, application, or an entire development instance.
Oracle Application Express End User Guide	E29541	Explains common Oracle Application Express end user tasks.
Backup and Recovery		

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Database Backup and Recovery User's Guide	E50658	Explains how to use Recovery Manager (RMAN) to back up, recover, and duplicate a database, and how to use Oracle Flashback Technology to restore all or a portion of your data to a previous state.
Oracle Database Backup and Recovery Reference	E50791	Provides detailed descriptions of the Recovery Manager (RMAN) commands and recovery catalog views.
Data Warehousing		
Oracle Database Data Warehousing Guide	E41670	Provides conceptual, reference, and implementation material for using Oracle Database in data warehousing. It covers the full range of data warehousing activities, from physical database design to advanced calculation techniques.
Oracle Data Guard Broker	E48241	Describes the Oracle Data Guard broker, a management and monitoring interface that automates many of the tasks involved in configuring and monitoring standby databases in an Oracle Data Guard configuration.
Oracle Data Guard Concepts and Administration	E48552	Provides a comprehensive overview of Oracle Data Guard concepts and describes how to configure and implement standby databases that can take over production operations if your production database becomes unusable. This guide includes several database scenarios such as creating, recovering, failing over, switching over, configuring, and backing up standby and primary databases.
Language and Interface		
*Pro*FORTRAN Supplement to the Oracle Precompilers Guide*	E18411	Supplements the Oracle Database Programmer's Guide to the Oracle precompilers to provide additional information on how to develop FORTRAN programs that use the database languages SQL and PL/SQL to access and manipulate Oracle data.
*Oracle SQL*Module for Ada Programmer's Guide*	E18412	Describes how to use SQL*Module to develop module programs and Ada code to access and manipulate Oracle data.
Oracle C++ Call Interface Programmer's Guide	E48221	Presents the Oracle C++ Call Interface, an application program interface that enables applications written in C++ to interact with one or more Oracle databases. To extend the functionality of code generated by the Object Type Translator, programmers may want to add code in the OTT-generated file. The OTT can distinguish between OTT-generated code and user-generated code by looking for some predefined markers (tags).

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Call Interface Programmer's Guide	E49886	Describes the Oracle Call Interface (OCI) which is an application programming interface (API) that allows applications written in C or C++ to interact with Oracle Database.
*Pro*COBOL Programmer's Guide*	E53282	Describes how to develop COBOL programs that use the database languages SQL and PL/SQL to access and manipulate Oracle data.
Oracle Database Programmer's Guide to the Oracle Precompilers	E53283	Describes how to develop applications using the Oracle precompilers and the SQL and PL/SQL database languages to access and manipulate Oracle data.
*Pro*C/C++ Programmer's Guide*	E53432	Describes how to develop C++ programs that use the SQL and PL/SQL database languages to access and manipulate Oracle data.
Networking and Security		
Oracle Database 2 Day + Security Guide	E17609	Provides an introduction to securing an Oracle database using the default security features, such as how to secure the database installation and configuration or manage encryption. It includes beginning level tutorials for creating secure application roles, Oracle Database Vault policies, Oracle Virtual Private Database policies, Oracle Data Redaction policies, Oracle Label Security policies, and unified auditing policies.
Oracle Database Net Services Administrator's Guide	E17610	Explains how to plan, configure, and manage enterprise-wide connectivity with Oracle Net Services.
Oracle Database Net Services Reference	E17611	Contains complete listing with descriptions of the control utility commands and configuration file parameters available for managing components of Oracle Net Services.
Oracle Database Enterprise User Security Administrator's Guide	E17731	Explains how to configure and use enterprise users, which are users defined in a directory. The guide also covers administrative tasks related to enterprise user security.
Oracle Database Security Guide	E48135	Explains how to configure an Oracle database to use the default security features.

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Database Real Application Security Administrator's and Developer's Guide	E48189	Describes how to implement Real Application Security on the database. Real Application Security is a new feature in Oracle Database 12c. Real Application Security is a database authorization model that enables end-to-end security for multitier applications. It provides an integrated solution to securing the database and application user communities. Also, it advances the security architecture of Oracle Database to meet existing and emerging demands of applications developed for the Internet.
Oracle Database Real Application Security Java API Reference	E48191	Documents two Java packages: - oracle.security.xs contains the APIs for using lightweight user sessions - oracle.security.xs.admin contains the APIs for administering lightweight user session
Oracle Label Security Administrator's Guide	E48437	Explains how to configure and use Oracle Label Security, which provides row-level access control based on the virtual private database technology of Oracle Database Enterprise Edition.
Oracle Database Advanced Security Guide	E50333	Explains how to configure and use Oracle Database Advanced Security Transparent Data Encryption (TDE) and Oracle Data Redaction.
Oracle Database Real Application Security Session Service Java API Reference	E51179	Documents one Java package: - oracle.security.xs.ee.session contains the APIs for using an application session service.
Oracle Database Real Application Security Administration	E61899	Provides a graphical user interface (RASADM) that security administrators and developers using Real Application Security can use to develop end-to-end security for mid-tier applications. The RASADM application allows users to develop declarative data security policies to enforce access control requirements at the database layer to protect data at both the row and column levels. This approach allows developers to control application user access to data in Oracle database throughout all components of an Oracle enterprise in a common manner.
Oracle Real Application Clusters		
Oracle Database 2 Day + Real Application Clusters Guide	E17616	Describes how to install, configure, and administer Oracle Clusterware and Oracle Real Application Clusters (Oracle RAC) on a two-node system using the Oracle Linux operating system.
Oracle Clusterware Administration and Deployment Guide	E48819	Describes the Oracle Clusterware architecture and provides an overview of this product. This book also describes administrative and deployment topics for Oracle Clusterware.

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Real Application Clusters Administration and Deployment Guide	E48838	Describes the Oracle Real Application Clusters (Oracle RAC) architecture and provides an overview of this product. This book also describes administrative and deployment topics for Oracle RAC.
Oracle Real Application Clusters Installation Guide for Linux and UNIX	E48917	Describes how to install and configure Oracle Real Application Clusters (Oracle RAC) on systems running Linux and UNIX.
Oracle Text		
Oracle Text Application Developer's Guide	E41398	Provides information for building applications with Oracle Text.
Oracle Text Reference	E41399	Provides reference information for building applications with Oracle Text.
R Enterprise		
Oracle R Enterprise User's Guide	E35158	Describes how to install and use Oracle R Enterprise which provides an environment for statistical computing.
Oracle R Enterprise Release Notes	E35159	Contains important information that was not included in the documentation for Oracle R Enterprise 1.2.
Streams and Replication		
Oracle Streams Extended Examples	E17912	Contains extended examples that configure various types of Oracle Streams environments.
Oracle Database Advanced Replication Management API Reference	E18896	Contains reference information about and step-by-step instructions for using the replication management API, which is a set of PL/SQL packages for setting up and managing an Advanced Replication environment. It also includes reference information about data dictionary views that are important for Advanced Replication and security considerations for an Advanced Replication environment.
Oracle Streams Concepts and Administration	E53116	Contains conceptual information about Oracle Streams and information about administering, monitoring, and troubleshooting an Oracle Streams environment. It also contains instructions for using Oracle Streams to record table changes. It also contains conceptual information about information provisioning and instructions for using Oracle Streams for information provisioning. It also includes instructions for using Oracle Streams for database upgrades and database maintenance operations.

Table 3–10 (Cont.) Server Documentation

Documentation	Part Number	Description
Oracle Database Advanced Replication	E53117	Contains conceptual information about Oracle Database Advanced Replication. It also includes information about planning an Advanced Replication environment and information about troubleshooting Advanced Replication problems.
Oracle Streams Replication Administrator's Guide	E53118	Contains instructions for configuring and managing an Oracle Streams replication environment. It also includes best practices for Oracle Streams replication environments and instructions for migrating from Advanced Replication to Oracle Streams replication.
Oracle Database XStream Guide	E53119	Contains conceptual information about XStream. Also contains instructions for configuring and using XStream.

3.1.11 Spatial Documentation

Table 3–11 Spatial Documentation

Documentation	Part Number	Description
Oracle Spatial and Graph GeoRaster Developer's Guide	E49118	Provides usage and reference information for the GeoRaster feature of Oracle Spatial and Graph, which lets you store, index, query, analyze, and deliver raster data (raster image and gridded data and its associated metadata).
Oracle Spatial and Graph Developer's Guide	E49172	Provides usage and reference information for indexing and storing spatial data and for developing spatial applications using Oracle Spatial and Graph and Oracle Locator.
Oracle Spatial and Graph Java API Reference	E50631	Provides Javadoc-generated information about interfaces and classes that support features available with Oracle Spatial and Graph, including GeoRaster and the Topology Data Model and Network Data Model Graph features, and with MapViewer, which is packaged with Oracle Fusion Middleware.
Oracle Spatial and Graph RDF Semantic Graph Developer's Guide	E51611	Provides usage and reference information about Oracle Database support for semantic technologies, including storage, inference, and query capabilities for data and ontologies based on Resource Description Framework (RDF), RDF Schema (RDFS), and Web Ontology Language (OWL).
Oracle Spatial and Graph Topology Data Model and Network Data Model Graph Developer's Guide	E53120	Provides usage and reference information about the Topology Data Model and Network Data Model Graph features of Oracle Spatial and Graph.

3.1.12 TimesTen In-Memory Database Documentation

Table 3–12 TimesTen In-Memory Database Documentation

Documentation	Part Number	Description
Oracle TimesTen In-Memory Database Release Notes	E21630	Contains important information not included in the documentation for Oracle TimesTen In-Memory Database.
Oracle TimesTen Application-Tier Database Cache Introduction	E21631	Describes the features of TimesTen Application-Tier Database Cache and provides information to help developers plan a TimesTen Cache application.
Oracle TimesTen In-Memory Database Installation Guide	E21632	Provides information about installing and upgrading the Oracle TimesTen In-Memory Database.
Oracle TimesTen In-Memory Database Operations Guide	E21633	Provides information about how Oracle TimesTen In-Memory Database works. It includes step-by-step instructions and examples that show how to perform commonly needed tasks.
Oracle TimesTen Application-Tier Database Cache User's Guide	E21634	Provides information about designing and configuring the TimesTen Application-Tier Database Cache. It includes information about setting up a cache grid.
Oracle TimesTen In-Memory Database Replication Guide	E21635	Provides information about designing and configuring TimesTen to TimesTen replication, including information about TimesTen integration with Oracle Clusterware.
Oracle TimesTen In-Memory Database Troubleshooting Guide	E21636	Provides information about utilities and procedures for troubleshooting problems that may be encountered when using the Oracle TimesTen In-Memory Database.
Oracle TimesTen In-Memory Database C Developer's Guide	E21637	Provides information about developing C applications to access the Oracle TimesTen In-Memory Database through ODBC. It also includes information about TimesTen support for XLA, OCI, and the Pro*C/C++ Precompiler.
Oracle TimesTen In-Memory Database Java Developer's Guide	E21638	Provides information about developing Java applications to access the Oracle TimesTen In-Memory Database through JDBC. It also includes information about support for TimesTen JMS/XLA.
Oracle TimesTen In-Memory Database PL/SQL Developer's Guide	E21639	Provides information about using PL/SQL with the Oracle TimesTen In-Memory Database.
Oracle TimesTen In-Memory Database TTClasses Guide	E21640	Describes how to use the TimesTen C++ Interface Classes (TTClasses) library.
Oracle TimesTen In-Memory Database SQL Reference	E21642	Provides a reference for SQL statements, expressions, and functions used with the Oracle TimesTen In-Memory Database, including TimesTen SQL extensions. It also describes data types.

Table 3–12 (Cont.) TimesTen In-Memory Database Documentation

Documentation	Part Number	Description
Oracle TimesTen In-Memory Database Reference	E21643	Provides a reference for attributes, built-in procedures, utilities, and system limits for the Oracle TimesTen In-Memory Database.
Oracle TimesTen In-Memory Database System Tables and Views Reference	E21644	Describes TimesTen system tables and replication tables.
Oracle TimesTen In-Memory Database PL/SQL Packages Reference	E21645	Provides reference information for packages provided with TimesTen PL/SQL.
Oracle TimesTen In-Memory Database Error Messages and SNMP Traps	E21646	Provides a reference for Oracle TimesTen In-Memory Database error messages and SNMP traps.
Oracle TimesTen In-Memory Database JDBC Extensions Java API Reference	E21647	Provides reference information for TimesTen JDBC classes and interfaces.
Oracle TimesTen In-Memory Database JMS/XLA Java API Reference	E21648	Provides reference information for TimesTen JMS/XLA classes and interfaces.
Oracle Enterprise Manager System Monitoring Plug-in for Oracle TimesTen In-Memory Database User's Guide	E28645	Describes how to install the Oracle Enterprise Manager Plug-in for Oracle TimesTen In-Memory Database.
Oracle Enterprise Manager System Monitoring Plug-in for Oracle TimesTen In-Memory Database Release Notes	E28646	Provides late-breaking information about the Oracle Enterprise Manager Plug-in for Oracle TimesTen In-Memory Database.
Oracle Data Provider for .NET Oracle TimesTen In-Memory Database Support User's Guide for Windows	E38358	Enables ADO.NET data access from .NET client applications to TimesTen databases. This document covers features specific to ODP.NET support of and use with TimesTen.