Oracle® Database

New Features Guide

11*g* Release 1 (11.1)

B28279-04

August 2011

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®

Oracle Database New Features Guide, 11*g* Release 1 (11.1)

B28279-04

Contents

2   Oracle Database 11*g* 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 versions 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 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*

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 11*g* Database New Features

This chapter contains descriptions of all of the features that are new to Oracle 11*g* Database Release 1 (11.1). This chapter contains the following sections:

- Application Development
- Availability
- Database Overall
- Business Intelligence and Data Warehousing
- Clustering
- Content Management Services
- Fault Diagnosability
- Information Integration
- Location Services and Specialty Data
- Manageability
- Performance
- Security
- Windows

1.1 Application Development

The following sections describe new Application Development features for Oracle Database 11*g* Release 1 (11.1).

1.1.1 APIs and Precompilers

The following sections describe new Oracle Database 11*g* features in application programmatic interfaces. These features improve performance and scalability of applications and enable easy deployment of the Oracle client-side stack.

1.1.1.1 Client-Side Query Cache

This feature enables caching of query result sets in client memory. The cached result set data is transparently kept consistent with any changes done on the server side.

Applications leveraging this feature see improved performance for queries which have a cache hit. Additionally, a query serviced by the cache avoids round trips to the server for sending the query and fetching the results. It also reduces the server CPU that

would have been consumed for processing the query, thereby improving server scalability.

> **See Also:**
>
> *Oracle Call Interface Programmer's Guide* for details

1.1.1.2 OCI Client Session NLS Character Set Status Information

This new feature adds client session NLS and character set information to the V$SESSION_CONNECT_INFO and GV$SESSION_CONNECT_INFO views.

This enhancement provides DBAs with access to client-side session character set information for all clients. This information is especially useful in troubleshooting problems.

> **See Also:**
>
> *Oracle Database Globalization Support Guide* for details

1.1.1.3 Precompilers: Additional Array INSERT and SELECT Syntax Support by Pro*C/C++ and Pro*COBOL

In this Oracle Database 11*g* release, there is added support for IBM DB2 array INSERT and array SELECT syntax for Pro*C/C++ and Pro*COBOL.

Adding support for IBM DB2 array INSERT and array SELECT syntax helps with the migration of IBM DB2 installations to Oracle.

> **See Also:**
>
> *Pro*C/C++ Programmer's Guide* for details

1.1.1.4 Precompilers: Dynamic SQL Statement Caching in Pro*C/C++ and Pro*COBOL

Application developers can now precompile Pro*C/C++ and Pro*COBOL applications with the new statement caching option.

The potential benefit of this feature is an increase in performance by enabling statement caching in precompiled programs. The performance improvement is achieved by removing the overhead of parsing the dynamic statements on reuse.

> **See Also:**
>
> *Pro*C/C++ Programmer's Guide* for details

1.1.1.5 Precompilers: Fix Execution Plan in Pro*C/C++ and Pro*COBOL

This new feature focuses on fixing an execution plan for a Pro*C/C++ and Pro*COBOL application by using the outline feature.

By using the outline feature, you can ensure that the performance of the application is not affected when modules are integrated and deployed into different environments.

1.1.1.6 Precompilers: Flexible B Area Length for Pro*COBOL

The length of B Area for the Pro*COBOL application can now be from 8 to 253 columns.

The length of B Area for a Pro*COBOL program was limited to 72 when the format is set to ANSI. COBOL compilers now can support B Area length up to 253. This

provides a programmer with the flexibility to type a line that is longer than 72 columns.

1.1.1.7 Precompilers: Implicit Buffered Insert in Pro*C/C++ and Pro*COBOL

Some applications that require periodic insertion of new data use a single `INSERT` statement within a loop. When queried by a command-line argument, the precompiler buffers the rows being inserted, reducing network round trips.

The benefit is that an implicit buffered `INSERT` provides performance improvement that is comparable to using the precompiler array interface.

> **See Also:**
>
> *Pro*C/C++ Programmer's Guide* for details

1.1.1.8 Precompilers: Platform Endianness Support by Pro*COBOL

Oracle always stores Unicode data (UTF16) in Big Endian representation. Linux and Windows have Little Endian representation and Solaris has Big Endian representation. With this enhancement, Unicode data can be fetched in Little Endian format on Windows and Linux platforms using a Pro*COBOL application.

When a UTF16 (UCS2) string is fetched in `PIC N` variable in a Pro*COBOL application running on Little Endian platforms, the application can now be precompiled with the appropriate command-line option to have the correct platform endianness.

> **See Also:**
>
> *Pro*COBOL Programmer's Guide* for details

1.1.1.9 Precompilers: Scrollable Cursor Support in Pro*COBOL

Pro*COBOL now supports scrollable cursors to fetch data in a nonsequential manner.

A scrollable cursor enables you to access the rows of a database result set in a forward, backward, and random manner. This enables the program to fetch any row in the result set.

1.1.1.10 Precompilers: SQL99 Syntax Support by Pro*C/C++

With this enhancement, the SQL99 syntax for `SELECT`, `INSERT`, `DELETE`, and `UPDATE` statements and the body of the cursor in a `DECLARE CURSOR` statement is supported by Pro*C/C++.

Oracle Database supports most of the SQL99 syntax. With this enhancement, Pro*C/C++ also supports the SQL99 syntax that is supported by Oracle Database.

> **See Also:**
>
> *Pro*C/C++ Programmer's Guide* for details

1.1.2 Application Express

The following sections describe new features for Oracle Application Express. Application Express provides an application development tool that is built into the database and is installed in the database by default which is different from Oracle Database 10*g* implementation.

1.1.2.1 Application and Schema Comparison

New application and schema comparison capabilities enable you to identify differences between two selected applications.

You can also identify differences between objects in two different schemas.

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

1.1.2.2 Drag and Drop Item Layout

The Drag and Drop Layout page enables you to interactively reorder items within a given region, change select item attributes, create new items, and delete existing items.

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

1.1.2.3 Flash Charts

Oracle Application Express includes support for 18 different types of Flash charts. You can create Flash charts when creating a new page or by defining a Chart region.

This release also includes a utility that enables you to convert existing Scalable Vector Graphics (SVG) charts to Flash.

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

1.1.2.4 Friendly URL Syntax to Facilitate Bookmarks

By specifying zero as the session ID, you can now make it easier for application users to bookmark pages within an application.

This features makes all public page links consistent, making them easier to bookmark.

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

1.1.2.5 Improved Web Services

Oracle Application Express now supports more loosely defined Web services types and document styles. Both of these enhancements enable support for Web services created with JDeveloper and Oracle BPEL synchronous processes. You can now interact with Web services over a Secure Sockets Layer (SSL) and work with Web services that require basic authentication.

Finally, this release also enables you to manually create a Web services reference even if the Application Express engine cannot correctly parse a WSDL document.

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

1.1.2.6 Improved Workspace Management

This release also includes a number of workspace management enhancements. For users requesting new workspaces or requesting additional storage, you can control what sizes display.

You can also request an e-mail of your workspace names. And, finally, you can view a log of login attempts.

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

1.1.2.7 Microsoft Access Migration

Application Migration enables you to migrate a Microsoft Access application and generate an Oracle Application Express application.

This feature provides an easy way to migrate applications from Microsoft Access to Oracle Express.

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

1.1.2.8 New Password and Account Controls

You can now specify rules for password expiration, mandate strong passwords (that is, specify the minimum number of and type of characters), require that users change their passwords on first use, and lock accounts.

With this new feature, you can configure account login controls and password policies across a development instance in Oracle Application Express Administration Services. You can also specify controls for a specific account under Account Controls on the Edit User page.

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

1.1.2.9 Page and Region Caching

You can improve the performance of your application by taking advantage of page and region caching. New Cache attributes have been added to the Page attributes and Region Definition pages.

Page caching works well for static pages and region caching is a good choice for regions such as lists that do not have any conditions or regions containing HTML text.

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

1.1.2.10 PDF Printing

You can now include the ability to print a report by exporting a report region to PDF. You can also define report queries and print attributes for report regions as Shared Components making them accessible to all pages within an application.

The advantage of this feature is that it provides production of high quality, easily printable reports.

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

1.1.3 Extensible Indexing

The following sections describe extensible indexing features for Oracle Database 11*g* Release 1 (11.1).

1.1.3.1 Create Rules and Actions for DML Events With Rules Manager

Rules Manager now adds the ability to treat DELETE and UPDATE operations on a table as events that drive the rule evaluation and action execution (INSERT operations are already supported). In addition to pre-commit evaluation of rules based on trigger, this release supports post-transaction evaluation through use of the Change Notification feature of Oracle Database.

Because rules can now be processed during UPDATE and DELETE statements, developers can now model the entire business process for OLTP workloads in addition to the current "append-only" workloads characteristic of data warehouse and sensor-based applications.

> **See Also:**
>
> *Oracle Database Rules Manager and Expression Filter Developer's Guide* for details

1.1.3.2 Enhanced Rules Manager

Rules Manager now includes a PL/SQL API to store reusable and shareable rule conditions in SQL WHERE clause format. Complex rule conditions can be assembled from these shared elements.

In addition, rules can now be enabled and disabled through standard SQL.

Application developers can reuse rule conditions to speed the rule definition process. Since these shared conditions are stored once, changes propagate to all the complex rules referring to them. For example, if a rule component defines a "suspicious bank transaction" as involving an international bank transfer over $10,000, a change to the amount, location, or other condition needs to be made only once and it is updated in all complex rules using "suspicious bank transaction" in their event scenario.

Rules can be turned on and off to facilitate rule testing and control introduction into the system as well as to reflect changes in business conditions.

> **See Also:**
>
> *Oracle Database Rules Manager and Expression Filter Developer's Guide* for details

1.1.3.3 New Capabilities for Domain Index Operations

A new domain index callout updates table and column name metadata for a domain index during an ALTER TABLE ... RENAME or ALTER TABLE ... RENAME COLUMN command to ensure domain index metadata is kept up-to-date.

A partition-specific parameter string provides control over placement of domain index data during partition maintenance operations.

The SQL MERGE statement can now be used with a domain index.

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

1.1.3.4 Oracle Text Predicates and Aggregate Operators are Included in Rule Conditions

The Rules Manager feature of Oracle Database now supports the use of the Oracle Text `CONTAINS` operator with the SQL `WHERE` clause rule conditions.

Rules Manager can now compute aggregate values (`SUM`, `AVG`, `MIN`, `MAX`, and `COUNT`) using moving windows semantics. The window size can be based on time or number of events. The aggregate values for a group of events can be computed and tested against predefined or relative thresholds in rule conditions.

Rules can respond to real-world trends identified from a collection of events occurring over a particular length of time or a sliding window of time, such as for Business Activity Monitoring and Business Intelligence applications. For example, you can offer a promotion to a customer if the sum of purchases in the last three days is over $1000.00.

> **See Also:**
>
> *Oracle Database Rules Manager and Expression Filter Developer's Guide* for details

1.1.3.5 Oracle Text Predicates Included in Stored Expressions

The Expression Filter feature of Oracle Database now supports the use of the Oracle Text `CONTAINS` operator with the SQL `WHERE` clause stored expressions.

Stored expressions can evaluate text and XML data.

> **See Also:**
>
> *Oracle Database Rules Manager and Expression Filter Developer's Guide* for details

1.1.3.6 System-Managed Partitioning of Extensible Indexes

System-managed domain indexes (both non-partitioned and local partitioned) use Oracle Database to track and maintain the domain index storage tables during DDLs and partition maintenance operations on the base table. System-partitioned tables can be used as index storage tables for local domain indexes. System-managed domain indexes with range partitioning are supported in this release.

System-managed partitioning of domain indexes enhances local domain index manageability and performance.

> **See Also:**
>
> *Oracle Database Data Cartridge Developer's Guide* for details

1.1.4 Globalization and Unicode

This section lists improvements to globalization features and Unicode support of Oracle Database.

1.1.4.1 Unicode 5.0 Support

The NLS data files for AL32UTF8 and AL16UTF16 character sets have been updated to match version 5.0 of the Unicode Standard character database.

With this enhancement, Oracle Database conforms to the newest version of the standard.

1.1.5 Java in the Database

The major enhancements to the Java runtime in this release include performance with a new JIT compiler, ease-of-use with a JDK-like command-line interface, and tooling.

1.1.5.1 OracleJVM Java/JDK 5.0 Compatibility

OracleJVM supports Sun's JDK 1.5.

This feature provides portability of J2SE applications (stand-alone JDBC or middle-tier) into Oracle Database and the ability to take advantage of the key benefits of J2SE 5.0 such as the "declarative" programming style enabled by Java Annotations.

This feature allows sharing metadata of Java classes loaded by user class loader among different database user sessions.

1.1.5.2 OracleJVM JIT Configuration

The JIT is enabled by default (out-of-the-box) and does not need further configuration. However, it can be disabled or reenabled by setting `JAVA_JIT_ENABLED` to `TRUE` or `FALSE` in the parameter file or dynamically by altering the instance or the session, for example:

```
ALTER SYSTEM SET JAVA_JIT_ENABLED=true
or
ALTER SESSION SET JAVA_JIT_ENABLED=true
```

In addition, the `OJVM_JIT_PROPERTIES` are used to specify the list of properties, for example:

```
OJVM_JIT_PROPERTIES="key0-value0","key1=value1","keyword0"
```

Currently, supported properties are:

■   `DISABLE_OPTIM_*` which are various switches to disable optimization passes.

■   New dictionary table and Oracle Runtime methods for checking the compilation status of classes or methods.

■   New column `IS_COMPILED` in `USER_JAVA_METHODS`, `ALL_JAVA_METHODS`, and `DBA_JAVA_METHODS` views set to `YES` or `NO`.

This feature allows fine-grained control on the behavior of the JIT. In addition, you have the ability to query the status of compiled classes and methods.

1.1.5.3 OracleJVM JIT (Transparent Native Java Compilation Without a C Compiler)

Enabled by default and similar to the JDK JIT, this feature compiles Java in the database natively and transparently without the need of a C compiler.

The JIT runs as an independent session in a dedicated Oracle server process. There is at most one compiler session per database instance and it is Oracle RAC-aware and amortized over all Java sessions.

The compilation session uses profiling feedback data gathered by all Java sessions to compile the most often used methods, so as to get the best overall performance. The executable binary persists in the database.

Java sessions just use the executable binaries of already compiled methods or discover newly compiled methods through an event polling performed by OracleJVM at regular intervals. When there is a compiled code for a given method, future executions of that method use the compiled version.

Conversely, when a Java class is invalidated, the persisted compiled code of the class or method is removed.

This feature brings two major benefits to Java in the database:

- Increased performance of pure Java execution in the database resulting from the transparent compilation of Java into native machine code, similar in behavior to a JDK JIT.

- Ease-of-use. Unlike the Static Native Java compiler (also known as NCOMP or JAccelerator) in previous releases, the new JIT Compiler (when enabled at database configuration level) is activated transparently, without the need of an explicit command, when Java is executed in the database.

In addition, this feature removes the need for a C compiler.

> **See Also:**
>
> *Oracle Database Java Developer's Guide* for details

1.1.5.4 OracleJVM Ease-of-Use: JDK-Like Interface

Similar to the JDK VM, this feature provides a command-line interface to OracleJVM through new PL/SQL interfaces, for example:

```
FUNCTION runjava (cmdline VARCHAR2) RETURN VARCHAR2;
or
FUNCTION runjava_in_current_session (cmdline VARCHAR2) RETURN VARCHAR2;
```

Analogous to the JDK/JRE Java shell command, the JDK-like interface takes a subset of the JDK/JRE command-line syntax including:

```
[-options] classname [arguments...]
or
[-options] -jar jarfile [arguments...]
```

The supported options include:

- `{cp|classpath} < ":" separated list of file system directories/JAR/ZIP to search for class files>`

- `D<name>=<value>` set a system property

- `Xbootclasspath:<":" separated directories/zip/jar files to search for bootstrap classes and resources>`

- `Xbootclasspath/a:<":" separated directories/zip/jar files to append to the end of bootstrap class path>`

- `Xbootclasspath/p:<":" directories/zip/jar files to prepend in front of bootstrap class path>`

This feature makes Java easy to use using command-line syntax, which is similar to the JDK JAVA command line. In addition to database resident Java classes, this feature allows the execution in the database runtime environment of file system resident Java class, using the standard `classpath` syntax. In addition, it allows setting system properties using the standard `-D` syntax.

See Also:

Oracle Database Java Developer's Guide for details

1.1.5.5 OracleJVM Ease-of-Use: Property Management Interface

This feature provides `DBMS_JAVA` functions to emulate JDK System property setting for the duration of the session. The settings are `set_property`, `get_property`, `remove_property`, and `show_property`.

Java developers can now emulate familiar JDK behavior such as system properties support when using Java in the database.

1.1.5.6 OracleJVM Ease-of-Use: Output Redirect

This feature provides `DBMS_JAVA` functions and procedures for controlling the redirection of Java output, for example:

```
{set | remove | enable | disable | query}_output_to_{sql | java | file}
```

And for enabling or disabling sending output to trace, for example:

```
trace {enable | disable}_output_to_trc
```

Java developers can now emulate familiar JDK behavior such as output redirect. In addition, they can now disable sending output to a trace file, which was not possible in previous releases.

1.1.5.7 OracleJVM Ease of Use: Database Resident Jars

This feature allows supporting JAR as a unit of operations (and as a result of support for signed JARs) using new `loadjava` options, `-jarsasdbobjects` and `-prependjarnames`.

When these new options are not used, `loadjava` explodes the jar into individual classes, while `dropjava` goes back to the original jar (from which the classes are loaded) and drop the classes each at a time.

Java developers using OracleJVM benefit from the same JDK features and behavior such as JAR support.

1.1.5.8 OracleJVM Ease of Use: Two-Tier Java Session Termination

This feature allows the termination of Java within a database session, with and without retention of system properties, using `endsession` and `endsession_and_related_state`.

Java developers can terminate Java in a session while keeping the property settings so as to speed up Java reinitialization within the same database session. This feature also allows immediate termination of Java sessions, without requiring the use of `System.exit`.

1.1.5.9 OracleJVM Utilities Enhancements

Enhancements to existing OracleJVM utilities (`loadjava`, `dropjava`, and `ojvmjava`) include the deprecation of least-frequently-used options of `loadjava` and the ability to load Java classes over the web (HTTP URL support). For example:

```
loadjava -u scott/tiger -r
-v http://my.server.com/this/is/the/path/my.jar
```

Enhancements to `dropjava` include:

- list-based deletion
- exception handling
- ability to connect to a different user
- invalid command reporting

`ojvmjava` has been extended to support the JDK-like command such as `runjava` and its options. For example:

```
ojvmjava -u scott/tiger -runjava -d <machine-name-or-ip>:1521:<sid> -t
```

And, then followed with:

```
$java -cp .<class>
```

In addition, a new standalone Class Closure Tool (`ojvmtc`) is provided for ensuring that a given list of file system resident classes and deployable JARs to be closed as much as possible before loading (for example, `loadjava` invocation).

The benefits are increased usability and simplified deployment of Java applications in Oracle Database.

> **See Also:**
>
> *Oracle Database Java Developer's Guide* for details

1.1.5.10 OracleJVM Manageability: JMX Interface

This feature provides support for OracleJVM with the standard JMX interface.

Support for JMX interface provides:

- Ease-of-use (any standard JMX-enabled GUI or console)
- Runtime monitoring of OracleJVM (memory objects heaps, threads, GC and classes)
- Ability to trigger GC from the console

1.1.6 JDBC and SQLJ

For Java developers, this release provides the following enhancements to JDBC and SQLJ APIs:

- Support for the latest Java standards including Java SE 1.5 and JDBC 4.0.
- A faster protocol for the Streams/AQ Java interface.
- Support for Oracle advanced security in JDBC-Thin.
- Enhancements to SQLJ code generation and customizer.
- Enhancements to XA support.

■ Support of event notification.

1.1.6.1 JDBC 4.0 Connection and Statement Enhancements

JDBC 4.0 introduces new methods for:

■ Validating a connection.

■ Setting and fetching client information on a connection.

■ Returning a list of client properties supported by the driver.

■ Allowing an application to mark that a statement should not be pooled.

■ Allowing a statement pool manager to register a `StatementEventListener` on the pooled connection.

In addition, it provides a new `StatementEventListener` interface and a new `StatementEvent` class.

These JDBC 4.0 enhancements improve the scalability of JDBC applications and provide better control and management of statement objects.

> **See Also:**
>
> *Oracle Database JDBC Developer's Guide* for details

1.1.6.2 JDBC 4.0 Support for New Data Types

This feature provides new standard types including `java.sql.XMLType`, `java.sql.RowID`, and `java.sql.Nclob` to support the National Character Set (`NCHAR`, `NVARCHAR`, `LONGVARCHAR`). In addition, the JDBC 4.0 implementation enhances `BLOB` and `CLOB` data type manipulation.

Standard support for new SQL types significantly increases the portability of JDBC applications across databases.

> **See Also:**
>
> *Oracle Database JDBC Developer's Guide* for details

1.1.6.3 JDBC Support for ANYTYPE, ANYDATA, and ANYDATASET

The following support has been added to JDBC in this release:

■ `ANYTYPE` which represents any SQL type including persistent database scalar or object types, and transient unnamed types.

■ `ANYDATA` as a table column type to store heterogeneous values (built-in and user-defined) within the same column, along with the metadata.

■ `ANYDATASET` which represents a set of data instances of the same type, along with the metadata.

The benefits include:

■ Mapping any Java types to a general purpose SQL data type (`ANYTYPE`).

■ Persisting Java objects to the same database table column (`ANYDATA`).

■ Mapping any Java collections into `ANYDATASET`.

> **See Also:**
>
> *Oracle Database JDBC Developer's Guide* for details

1.1.6.4 JDBC Support for Continuous Query Notification

This feature allows Java Applications or Containers to subscribe to SQL queries watch list in the database and be notified when:

- DML or DDL changes affect the objects associated with the queries, or

- DML or DDL changes affect the result set

The notifications are published when the transaction (DML or DDL) commits.

Java Applications and Container prevents unnecessary database querying and implements efficient and consistent result set caching and cache invalidation.

> **See Also:**
>
> *Oracle Database JDBC Developer's Guide* for details

1.1.6.5 JDBC Support for Database Diagnosability

Through this feature, JDBC provides better support for logging, diagnosability and DMS monitoring:

- Logging has been reimplemented to fully utilize `java.util.logging`. This allows you to selectively log events that occur during the execution of the JDBC driver code such as SQL exceptions, SQL statement executions, or entry or exit from internal JDBC methods.

- JDBC support for performance monitoring uses DMS to capture performance-critical JDBC events including:

  - Connection cache statistics

  - Connection creation

  - Statement cache statistics

  - Query execution

  - Result fetch

  - Oracle RAC events

  - Load balancing statistics

  - New Oracle Database 11*g* features

- JDBC support for Diagnosability Management consists of an MBean, `oracle.jdbc.DiagnosabilityMBean`. This MBean provides the means to enable and disable JDBC logging and provides additional statistics about JDBC internals.

This feature simplifies JDBC logging and makes the interpretation of the results easy. Dynamic Monitoring System (DMS) metrics give Java applications the ability to track JDBC events and performance metrics, as well as a programmatic control and configuration of diagnosability features.

> **See Also:**
>
> *Oracle Database JDBC Developer's Guide* for details

1.1.6.6 JDBC Support for Database Startup and Shutdown

You can now perform database startup or shutdown using the following JDBC Oracle Connection methods:

- `DBSTARTUP_NO_RESTRICTION`

- `DBSTARTUP_FORCE`

- `DBSTARTUP_RESTRICT`

- `DBSHUTDOWN_CONNECT`

- `DBSHUTDOWN_TRANSACTIONAL`

- `DBSHUTDOWN_TRANSACTIONAL_LOCAL`

- `DBSHUTDOWN_IMMEDIATE`

- `DBSHUTDOWN_ABORT`

- `DBSHUTDOWN_FINAL`

Note that you must be connected to a dedicated server as `SYSOPER` or `SYSDBA` in `PRELIM_AUTH` mode.

> **See Also:**
>
> *Oracle Database JDBC Developer's Guide* for details

1.1.6.7 JDBC Support for Native Streams AQ Protocol

This feature exposes a new Java package (`oracle.jdbc.aq`) that leverages the new native JDBC APIs for Streams AQ. This package contains the following classes and interfaces:

- `AQQueue`

- `AQMessage`

- `AQAgent`

- `AQDequeueOptions`

- `AQEnqueueOptions`

- `AQMessageProperties`

- `AQRawPayload`

- `AQObjectPayload`

- `AQNotificationEvent`

- `AQNotificationListener`

- `AQNotificationRegistration`

Java and JMS applications run faster (using Native AQ operations as opposed to PL/SQL procedures) when accessing Streams AQ and OJMS queues. In addition, these APIs allow asynchronous notifications of AQ events and exchange rich message payloads including `RAW` data type, `ANYDATA` data type, and `OBJECT` data type.

> **See Also:**
>
> *Oracle Database JDBC Developer's Guide* for details

1.1.6.8 JDBC-Thin Support for Oracle Advanced Security

This feature provides complete JDBC-Thin support for Oracle Advanced Security in terms of encryption, password authentication, strong authentication, and data integrity or checksum.

New security features include Secure Sockets Layer (SSL), Remote Authentication Dial-In User Service (RADIUS) and Kerberos authentication, AES encryption, and

SHA1 checksum algorithm. Advanced security support in JDBC-Thin simplifies the development and deployment of secure Java applications.

See Also:

Oracle Database JDBC Developer's Guide for details

1.1.6.9 SQLJ: Enhanced Profile Print Option of the Customizer

SQLJ customizer is used during ISO standard code generation. The customizer tailors SQLJ profiles for a particular database implementation and vendor-specific features and data types. The profile print option (`-P-print`) of SQLJ has been enhanced to print all the customizer options that were passed into the SQLJ translator.

This feature provides information about customizations that SQLJ has received during the precompilation phase to help diagnose any problems that may arise and are specific to different customizations.

See Also:

Oracle Database SQLJ Developer's Guide for details

1.1.6.10 SQLJ: Outline Generation Option

An application can support SQL execution plan stability by using the outline feature of Oracle. An outline is implemented as a set of optimizer hints that are associated with the SQL statement. If the use of the outline is enabled for the statement, Oracle automatically considers the stored hints and tries to generate an execution plan in accordance with those hints.

A new translation time option is provided in SQLJ to generate and run a SQL file containing the Oracle outline creation commands for all the SQL statements present in a SQLJ application.

This feature helps in ensuring the expected behavior of a database query at development time and helps to better diagnose problems.

See Also:

Oracle Database SQLJ Developer's Guide for details

1.1.6.11 SQLJ: Support for JDK 1.5

SQLJ applications can now be translated and run under JDK 1.5.

The JDK 1.4 and JDK 1.5 specific language structures outside the #sql region are recognized by the translator.

See Also:

Oracle Database SQLJ Developer's Guide for details

1.1.6.12 SQLJ: XA Support

The `javax.sql.XADataSource` interface of JDBC outlines standard functionality of XA data sources. An XA data source is a factory for XA connections. Oracle JDBC implements the `XADataSource` interface through the `OracleXADataSource` class. The `getConnection()` method of the `OracleXADataSource` class returns an XA connection to the underlying data source.

The benefit of this feature is that these connections can be used in SQLJ applications by creating the appropriate connection contexts in SQLJ.

See Also:

Oracle Database SQLJ Developer's Guide for details

1.1.7 Oracle Developer Tools for Visual Studio .NET

These features enhance developer productivity for building .NET applications with Oracle Database. They provide native integration with ADO.NET 2.0 functionality using Visual Studio development environments with which .NET developers are familiar. In addition, these features reduce the time it takes to develop .NET applications for Oracle.

1.1.7.1 General Visual Studio 2005 Support

This feature provides users of Visual Studio 2005 the same support currently offered for Oracle Develop Tools (ODT) users using Visual Studio 2003. The changes include using different icons and integrating visually with the Visual Studio 2005 look and feel.

This feature makes it easy for customers to continue using ODT when they migrate from Visual Studio 2003 to the Visual Studio 2005 environment.

1.1.7.2 Oracle PL/SQL Debugger in Visual Studio

The Oracle PL/SQL debugger is now integrated directly into Visual Studio. Developers can debug code, set breakpoints, and watch variables exactly as they would for any other language. Three types of debugging are supported:

- Direct Database Debugging

  The developer runs the stored procedure from the Oracle Explorer inside of Visual Studio and enters in the parameters.

- Multi-Tier Application Debugging

  The developer steps from .NET code directly into PL/SQL and back again with the parameters automatically being passed.

- External Application Debugging

  The developer runs any 9.2 or later Oracle application that can be located anywhere (for example, from SQL*Plus). When it calls the stored procedure, the break point in Visual Studio fires and the developer can begin debugging with the stored procedure parameters being passed automatically.

This feature enables developers to seamlessly debug .NET applications that make calls to PL/SQL stored procedures. Developers can debug both .NET code and PL/SQL without having to leave Visual Studio. They can also debug PL/SQL stored procedures that are called from any arbitrary Oracle program, written in any language, and located on the same machine or on a remote machine.

1.1.7.3 Support for Visual Studio 2005 Data Sources Window and Designers

This feature exposes the DDEX interface to enable some Visual Studio 2005 designers as well as integrate with the Data Sources window to enable the new style drag-drop code generation in Visual Studio 2005.

1.1.8 Oracle Data Provider for .NET (ODP.NET)

These features enhance developer productivity for building .NET applications with Oracle Database. They provide native integration with ADO.NET 2.0 functionality

using Oracle Data Provider for .NET (ODP.NET) development environments with which .NET developers are familiar. In addition, these features reduce the time it takes to develop .NET applications for Oracle.

1.1.8.1 Provider-Independent API Using Base Classes and Factories

ODP.NET inherits from the `System.Data.Common` namespace to provide access through a single API to databases across multiple providers.

ADO.NET developers can build applications that work with multiple vendors' database servers with one set of data access code and with a minimum of vendor-specific code. Developers can now be more productive when building one set of code that works against multiple vendors' databases.

> **See Also:**
>
> *Oracle Data Provider for .NET Developer's Guide* for details

1.1.8.2 Connection String Builder

ODP.NET provides a strongly typed connection string builder class that inherits from `DbConnectionStringBuilder`. The connection string builder lets developers programmatically create syntactically correct connection strings based on user input, as well as parse and rebuild existing connection strings by using class methods. Connection string builders provide strongly-typed properties that correspond to known key/values pairs.

ODP.NET connection strings are now more secure and more manageable. Because connection string parameters can now be strongly typed, they are less subject to string injection attacks. Connection string parameters can now be referenced in configuration files or as ODP.NET properties, making managing connection information easier.

> **See Also:**
>
> *Oracle Data Provider for .NET Developer's Guide* for details

1.1.8.3 OracleDataAdapter Batch Processing

During updates to Oracle Database, the ODP.NET `OracleDataAdapter` can batch a set of rows back to the database per round trip. The number of rows per round trip can be set by the developer.

This feature enhances application performance by reducing the number of round trips to the database when applying updates from the DataSet.

> **See Also:**
>
> *Oracle Data Provider for .NET Developer's Guide* for details

1.1.8.4 Row-Level Change Notifications

.NET developers often used client-side cached data. When another user changes the original data on the server, the database sends a change notification message to the client, indicating that the client data is now invalid. Previously, change notifications were registered at the table level. In this release, developers can register their queries at the row level so that they are alerted only when the rows they selected have changed.

.NET clients can now receive more accurate change notification alerts.

See Also:

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

1.1.8.5 Schema Discovery

ODP.NET schema discovery allows applications to request and return information about the database schema to the client. Different database schema elements such as tables, columns, and stored procedures are exposed through the GetSchema methods of the OracleConnection class.

ODP.NET can retrieve a set of generic and Oracle-specific schema information.

Schema discovery supplies a common API for all ADO.NET 2.0 drivers to retrieve database schema information to the client application.

See Also:

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

1.1.8.6 Data Source Enumeration

The ODP.NET OracleDataSourceEnumerator class provides a list of active Oracle database servers.

Data source enumeration can be retrieved programmatically in a generic manner common to all ADO.NET 2.0 providers.

See Also:

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

1.1.8.7 Faster LOB Retrieval

ODP.NET now makes fewer round trips to the database server to retrieve LOB information.

ODP.NET LOB retrieval performance is now faster.

See Also:

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

1.1.8.8 Improved Parameter Context Caching in Statement Caching

For statement caching, ODP.NET now caches parameter contexts after the first statement execution.

Many applications execute the same SQL or PL/SQL statement repeatedly. In the first execution of the statement, ODP.NET can now cache parameter contexts so that they can be reused. This improves the performance of subsequent statement executions.

See Also:

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

1.1.9 Oracle Provider for OLE DB

These features improve data access performance from Windows clients to Oracle Database. Many of these features are transparent, meaning applications experience faster performance by upgrading the database server and client without any code changes.

1.1.9.1 Faster Performance

Oracle Provider for OLE DB has implemented a number of new features in its provider to improve query performance for the end user.

> **See Also:**
>
> *Oracle Provider for OLE DB Developer's Guide for Microsoft Windows* for details

1.1.9.2 Improved Metadata Caching

For metadata caching, Oracle Provider for OLE DB now caches the metadata after the first execution when statement caching is turned off. This is useful when a large amount of column metadata is retrieved with a small amount of row data. Oracle Provider for OLE DB has also improved metadata caching when statement caching is turned on.

Many applications execute the same SQL or PL/SQL statement repeatedly. In the first execution of the statement, Oracle Provider for OLE DB can cache a query's metadata so that it can be reused. This improves the performance of subsequent statement executions.

> **See Also:**
>
> *Oracle Provider for OLE DB Developer's Guide for Microsoft Windows* for details

1.1.9.3 Improved Statement Caching

For statement caching, Oracle Provider for OLE DB now caches large fetch arrays and OCI error handles after the first execution.

Many applications execute the same SQL or PL/SQL statement repeatedly. In the first execution of the statement, Oracle Provider for OLE DB can now cache a statement's parse information so that it can be reused. This improves the performance of subsequent statement executions.

> **See Also:**
>
> *Oracle Provider for OLE DB Developer's Guide for Microsoft Windows* for details

1.1.10 PHP

The features in the following sections describe the enhancements to Oracle's client-side stack so that data access open source PHP driver performs better than PHP data access drivers for other databases.

1.1.10.1 Database Resident Connection Pooling (DRCP)

Prior to this release, session sharing capabilities were available for multithreaded applications only. With this new feature, different application processes can share sessions within the same host machine as well as across multiple host machines. The pool is maintained on the database server by a new background process.

This feature is useful for applications that cannot be deployed as multithreaded (for example, PHP applications in an Apache Web server environment) and database requests are done in a session state independent manner. These applications no longer have to hold the connections to the database persistently, thereby increasing the scalability of the database.

The pool is managed by the DBA and the OCI and OCCI applications can reuse the sessions by specifying a connection class name within which the application leaves sessions of similar state.

> **See Also:**
>
> *Oracle Call Interface Programmer's Guide* for details

1.1.11 PL/SQL

The following sections discuss features that:

- Reduce downtime associated with application patching
- Enhance performance and scalability (TPCC, APPS)
- Provide integration between PL/SQL and SQL

1.1.11.1 XA API Available Within PL/SQL

The functionality available in the XA interface to support transactions involving multiple resource managers, such as databases and queues, is now available from PL/SQL.

Application developers are now able to switch or share transactions across SQL*Plus sessions or processes using PL/SQL.

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for details

1.1.11.2 Allow Sequences in PL/SQL Expressions

In the previous release of Oracle Database, when a PL/SQL program needed to get a value from syntax, it used SQL. For example:

```
DECLARE n NUMBER;
BEGIN
SELECT Seq.Nextval INTO n FROM Dual;
END;
```

This posed a usability problem for PL/SQL programmers and causes runtime performance and scalability problems for the application particularly because it requires the use of a cursor.

In Oracle Database 11*g*, it is now possible to simply use `Seq.Nextval` in a PL/SQL expression. For example:

```
DECLARE n NUMBER := Seq.Nextval;
BEGIN
...
```

The same is possible for the `CURRVAL` pseudocolumn.

This feature brings improved usability for the PL/SQL programmer and improved runtime performance and scalability.

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

1.1.11.3 Enhancements to the Regular Expression Built-Ins

There is new syntax to access the *n*th subexpression in `REGEXP_SUBSTR` and `REGEXP_INSTR`. This is useful if, for example, the matching string is always surrounded by tags like `<string>` and `</string>` but you want to extract just what lies between these tags. `REGEXP_SUBSTR` returns the subexpression. `REGEXP_INSTR` returns the position.

The new built-in `REGEXP_COUNT` returns the number of times the pattern is matched in the input string.

This new functionality is available both in SQL and in PL/SQL.

The benefit is increased functionality for the regular expression built-ins. Earlier applications that needed this functionality had to write relatively elaborate application code which is no longer the case.

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for details

1.1.11.4 Functional Completeness of Dynamic SQL for PL/SQL

In Oracle Database 11*g*, native dynamic SQL now supports statements bigger than 32K characters by allowing a `CLOB` argument. Also included with this feature:

- `DBMS_SQL.PARSE()` gains a `CLOB` overload

- A `REF CURSOR` can be converted to a `DBMS_SQL` cursor and vice versa to support interoperability

- `DBMS_SQL` supports the full range of data types (including collections and object types)

- `DBMS_SQL` allows bulk binds using user-define collection types

The features for executing dynamic SQL from PL/SQL had some restrictions in Oracle Database 10*g*. `DBMS_SQL` was needed for Method 4 scenarios but it could not handle the full range of data types and its cursor representation was not usable by a client to the database. Native dynamic SQL was more convenient for non-Method 4 scenarios, but it did not support statements bigger than 32K. Oracle Database 11*g* removes these and other restrictions to make the support of dynamic SQL from PL/SQL functionally complete.

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

1.1.11.5 Method Invocation Scoping Operator

A subtype usually adds specific data attributes to its more generic supertype. Typically, a `NOT FINAL` member procedure, for example, `Show_Attributes()`, is implemented at each level in the type hierarchy. It is natural to implement this at one level by reusing the implementation at the parent level and by adding new code just for the attributes of the subtype.

Oracle Database 11*g* supports the ANSI SQL generalized-invocation syntax to allow the following:

`(<type-instance> AS <type-name>).<method-name>.`

Without this syntax, programmers were forced to use a cumbersome workaround which relied on using static subprograms rather than member subprograms.

This feature brings the implementation of object-oriented programming in Oracle Database closer to completeness.

> **See Also:**
>
> *Oracle Database Object-Relational Developer's Guide* for details

1.1.11.6 Named and Mixed Notation for User-Defined PL/SQL Subprogram in a SQL Statement

Consider a schema-level function `f()` declared as:

```
FUNCTION f(
p1 IN INTEGER := 1,
p2 IN INTEGER := 2,
...
pn IN INTEGER := 99)
RETURN INTEGER
```

Beginning in this release, it is now possible to invoke the function in a SQL statement. For example, named notation syntax is:

```
SELECT f(pn=>3, p2=>2, p1=>1) FROM dual
```

Or, mixed notation is:

```
SELECT f(1, pn=>3) FROM dual
```

In previous releases, attempting named or mixed notation resulted in an error.

The benefit is improved usability when a PL/SQL function, with many default parameters, is invoked in a SQL statement and only one or a small number of the actual parameters need to differ from their default values.

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

1.1.11.7 New Compound Trigger Type

A compound trigger has a section for each of the `BEFORE STATEMENT`, `BEFORE EACH ROW`, `AFTER EACH ROW`, and `AFTER STATEMENT` timing points. All of these sections can access a common PL/SQL state. The common state is created when the firing statement starts and destroyed when it completes, even when the firing statement causes an error. A compound trigger can be used to speed performance in an auditing scenario where audit rows can be accumulated in the EACH ROW section and bulk inserted to the audit table in the `AFTER STATEMENT` section. A similar approach is sometimes required to avoid the mutating table error. The alternative to using compound triggers is to model the common state with an ancillary package. This approach is both cumbersome to program and subject to memory leak when the triggering statement causes an error and the `AFTER STATEMENT` trigger does not fire.

This feature brings improved usability for the PL/SQL programmer and improved runtime performance and scalability.

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

1.1.11.8 PL/Scope

PL/Scope allows you to browse PL/SQL source code analogously to the way that Cscope (see http://cscope.sourceforge.net/) allows you to browse C source code. You can search for and display all types of definitions, declarations, assignments and references in the PL/SQL source code.

The PL/SQL compiler can optionally derive the metadata needed to support PL/Scope and store it in the database catalog. The metadata takes into account the nuances of the language, including scoping and overloading. You can generate reports (especially hyperlinked HTML reports) with supplied report generators. This feature is exposed through Oracle SQL Developer's interactive PL/SQL development environment.

PL/Scope supports increased developer productivity, especially for those who need to maintain someone else's code.

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for details

1.1.11.9 PL/SQL CONTINUE Statement

Many programming languages allow transfer of control either out of the loop entirely or within a loop back to a new iteration. In previous releases, PL/SQL only allowed transfer of control out of the loop using the EXIT statement. In Oracle Database 11*g* Release 1 (11.1), the new CONTINUE statement implements the ability to transfer control within a loop back to a new iteration.

Many algorithms require the loop control that this new CONTINUE statement now provides. With this explicit feature, you are no longer required to code several lines of PL/SQL and maintain commonly-adopted style rules.

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

1.1.11.10 PL/SQL Hierarchical Profiler

PL/SQL Hierarchical Profiler identifies hot spots and performance tuning opportunities in PL/SQL database applications. It reports the dynamic execution program profile organized by subprogram calls. SQL and PL/SQL execution times are accounted for separately. PL/SQL Hierarchical Profiler requires no special source or compile time preparation.

Start and stop procedures in the new package, DBMS_HPROF, control the recording of raw PL/SQL Hierarchical Profiler data. Then other subprograms in this package are used to transform the raw data into schema-level tables to allow the generation of reports. For example, hyperlinked XML or HTML reports.

This feature is part of Oracle SQL Developer's interactive PL/SQL development environment.

Programmer productivity and program performance are both enhanced because PL/SQL Hierarchical Profiler directs tuning effort towards those PL/SQL functions that will benefit most.

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for details

1.1.11.11 PL/SQL Inlining Optimization

Modern PL/SQL software includes many procedures frequently called in various programs. The power of the PL/SQL optimizing compiler is increased by incorporating the inlining optimization (replacement of a procedure call with a copy of the procedure body). The revised code executes more quickly because call overhead has been eliminated and because further optimizations are possible when the compiler knows the actual values with which the inlined procedure is called.

Performance gains can be substantial. The compiler does this optimization automatically. However, you may control the optimization in detail when necessary. Many PL/SQL programs execute faster, as a direct consequence of upgrade, simply by recompiling existing units.

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

1.1.11.12 SIMPLE_INTEGER Data Type

The new PL/SQL `SIMPLE_INTEGER` data type is a binary integer for use with native compilation which is neither null checked nor overflow checked. The range is all integer values in the inclusive interval [-2147483648..2147483647] and does not include a null value. These rules allow a direct hardware implementation when used in combination with PL/SQL native compilation.

The `SIMPLE_INTEGER` data type yields significant performance benefits compared to `PLS_INTEGER`. It can be used when the value will never be null and when overflow check is not needed.

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

1.1.12 XML Application Development

The features described in the following sections ensure that PL/SQL, Java, C and .NET programmers can leverage the full capabilities of Oracle XML DB and the Oracle XML DB repository. These new features also extend the infrastructure components to support Binary XML and provides an efficient method for handling very large XML documents. There is also support relevant to JSRs for XML application development including JSR-170 for content repository access and the emerging JSR-225 standard for XQuery. The Oracle XML libraries continue to deliver the best possible application development framework for XML developers working in J2EE and .NET environments, regardless of whether they are using the Oracle XML DB feature. These new features deliver highly optimized XSLT and XQuery engines for use with Oracle Application Server and Fusion Middleware.

1.1.12.1 Binary XML Support for Java

The new XDK binary XML processors include a binary XML encoder, a binary XML decoder, and a token manager to convert the schema or nonschema-based binary XML stream to and from XML 1.0 text and `SAX` events. The format is the same as the format used within Oracle XML DB.

This feature allows XML processing applications to leverage a compact, pre-parsed XML format which could save in memory and CPU consumption as well as network bandwidth. With a single format supported by Oracle XML DB and Oracle XDK and the Oracle XML DB support within the binary XML processors, this feature allows the performance optimization across tier XML applications.

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

1.1.12.2 Binary XML Support for Oracle XML DB

No single approach to persisting XML can solve all of the possible use cases presented by the XML application development paradigm. Binary XML (which is a way of representing the structure of an XML document) adds a third storage model for persisting XML in the database. It complements, rather than replaces, the existing object storage and text storage models.

Binary XML has two significant benefits:

- It enables significant optimizations of XML operations whether or not an XML schema is available.

- It allows the internal representation of XML to be the same regardless of whether the XML is in disk, in memory, or on the wire.

As with other storage mechanisms, the application developer can continue to use `XMLType` and its associated methods and operators without regard to the details of storage.

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

1.1.12.3 Configurable DOM Support for Java

The following options are provided for building Document Object Model (DOM):

- `static final String ACCESS_MODE = "DOM_ACCESS_MODE";`

- `static final String READ_ONLY = "READ_ONLY";`

- `static final String UPDATEABLE = "UPDATEABLE";`

- `static final String FORWARD_READ = "FORWARD_READ";`

- `static final String STREAMING = "STREAM_READ";`

This feature enables XML applications to achieve maximum efficiency in terms of memory usage and performance by configuring the DOM building with different access patterns such as read-only, forward read, or streaming. This feature prevents light-weight XML applications from having to deal with a heavy DOM object tree.

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

1.1.12.4 Event-Based XML Pull Parsing for C

XML pull parsing supports the stream-based parsing model which limits memory footprints by not preserving any document states in memory. Compared with `SAX`, the new pull parsing model only parses events requested by the application and accepts multiple input XML documents.

This feature improves the C XML processing performance and ensures the scalability with the new stream-based XML pull parsing. This feature also limits memory footprints by not preserving any document states in memory and improves the performance of the XML processing by only parsing stream events requested by the application. The pull parsing also makes the XML processing easy-to-use by allowing one parsing process to handle multiple XML documents.

See Also:

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

1.1.12.5 High Performance XPath Query for Java

Support JAXP 1.3 APIs for query XML content using XPath with the following extensions:

- Supports variable resolver in both static and dynamic context.
- Allows users to register runtime context.

This feature improves the performance and the usability of Java XPath/XSLT implementation.

1.1.12.6 In-Place XML Schema Evolution

This feature lets users make changes to a registered XML schema without having to unload and reload all of the XML documents that are compliant with the XML schema. Existing documents are not modified during in-place evolution, so only changes that do not invalidate the existing corpus of XML documents are permitted.

This feature enables rapid responses to changes in requirements by allowing certain common changes to XML schemas be made without having to unload and reload data. Organizations can take advantage of the benefits of XML schema-based XML storage in environments where the dynamic nature of the application or the large volume of XML content might make unloading and reloading the entire XML corpus to accommodate a change in the XML schema unrealistic.

The feature greatly reduces the overhead involved in making changes to XML schema in the majority of cases. Evolving an XML schema in cases where there are large number of documents compliant with the XML schema is a very expensive process that can take days to complete. In-place XML schema evolution allows the majority of common changes to be made almost instantaneously, with zero downtime, resulting in significant cost savings for the organization.

See Also:

Oracle XML DB Developer's Guide for details

1.1.12.7 Mid-Tier XQuery Engine Performance Improvements

XQuery performance improvements have been made in this release.

XQuery performance improvements for the mid-tier speed up applications relying on the mid-tier XQuery services.

1.1.12.8 Pluggable DOM Support for Java

The pluggable DOM splits the DOM implementation into two separate layers, the DOM API layer and the data layer. The API layer consists of a lightweight, transient node fragment that links to underlying data, which can be backed by external storage for scalability. The data has two varieties, internal data and plug-in data. The plug-in data support is through the InfosetReader implementations.

This feature allows XML applications to build DOM based on different forms of XML data through a unified interface. This feature improves portability of the XML applications.

See Also:

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

1.1.12.9 Scalable and High Performance XML Validation for C

The stream-based XML Schema validation allows XML applications to validate XML content against XML schemas in XML pull parsing process.

High performance XML processing and its scalability are very critical for enterprise XML applications. This feature improves the C XML processing performance and ensures the scalability with the new stream-based XML Schema validation.

See Also:

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

1.1.12.10 Scalable DOM Support for Java

This feature ensures the scalability of Database Object Model (DOM) processing for XML applications with the following support:

- Allows XML applications to process large XML documents by loading and saving DOM nodes in memory and using a page manager for physical binary data management.

- Allows storing DOM updates with references to the source data in Binary XML.

- Allows multiple applications sharing same DOM data source and operating on a lightweight, transient node fragment that links to underlying data, which can be backed by external storage for scalability.

- Supports Binary XML as the data source or the output.

- Supports shadow copy in DOM (`cloneNode()` or `importNode()`) which does not require the whole content being deep copied.

Scalability is the key to the success of the enterprise XML applications. This feature ensures the scalability of DOM processing for XML applications.

See Also:

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

1.1.12.11 Unified Java API for XML

The new unified Java API for XML allows mid-tier Java programs to leverage all of the benefits of `XMLType` by providing a disconnected mode of operation that allows `XMLType` to be used with a session pool model of connection management.

This allows the `XMLType` object to be disconnected from the database session that was used to create it.

This feature improves Java programmer productivity by allowing a single API to be used to manipulate XML content stored in Oracle XML DB and XML content stored in other sources.

Unified Java API also provides a disconnection mode of operation that allows `XMLType` to be used in a mid-tier environment where session pooling makes it impractical to hold the database connection that was used to instantiate the `XMLType`.

1.1.12.12 Unified Java API for XML

This feature allows mid-tier Java programs to leverage all of the benefits of `XMLType` by providing a disconnected mode of operation that allows `XMLType` to be used with a session pool model of connection management. This allows the `XMLType` object to be disconnected from the database session that was used to create it.

This feature improves Java programming productivity by allowing a single API to be used to manipulate the XML content stored in Oracle XML DB and the XML content stored in other sources.

This feature provides a disconnection mode of operation that allows `XMLType` to be used in a mid-tier environment where session pooling makes it impractical to hold the database connection that was used to instantiate the `XMLType`.

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

1.1.12.13 XMLDiff Support for C

The `XMLDiff` detects the differences between the two XML documents and represents the difference in XML. The `XMLPatch` takes the differences created by `XMLDiff` and applies the changes on the target XML document.

This feature allows high performance XML document comparison and user controlled updates. With the support for detecting differences between two XML documents, XML applications can easily control document versions, merge and synchronize the content. The feature also reduces the processor demand by handling only the changed data, reduces bandwidth demands by transmitting only updates, and increases security by splitting sensitive data into separate streams which can then be reassembled on delivery.

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

1.1.12.14 XMLIndex Enhancements

This feature provides an asynchronous mode for `XMLIndex`, where the indexing operations take place separately from the `INSERT` or `UPDATE` operations. Applications no longer have to wait for indexing to complete (which, for large documents, can be a significant operation) before being able to complete a transaction.

This feature also provides the ability to define the set of XPath expressions that are indexed by a given `XMLIndex`. The subset can be defined using `INCLUDE` or `EXCLUDE` syntax.

Subsetting XPath expressions that are indexed allows organizations to balance index size with disk usage, and to balance performance on insert with flexibility on retrieval.

This feature provides more flexibility for the organizations using `XMLIndex`. `XMLIndex` can be used in a wide range of applications. The feature improves application performance by allowing indexing to take place separately from `INSERT` and `UPDATE` operations. It also allows organizations to balance performance, flexibility, and disk usage when using `XMLIndex`.

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

1.2 Availability

The following sections describe Availability features for Oracle Database 11*g* Release 1 (11.1).

1.2.1 Availability Interfaces in Oracle Enterprise Manager

The following sections describe improvements to the Enterprise Manager interfaces for high availability features.

1.2.1.1 Browser-Based Enterprise Manager Integrated Interface for LogMiner

This feature now makes it possible to use the browser-based Oracle Enterprise Manager Database Control interface for LogMiner. In prior releases, administrators were required to use the standalone Java Console to use LogMiner. The Console was not integrated with the rest of Enterprise Manager and was cumbersome to install. With this new interface, administrators have a task-based, intuitive approach to using LogMiner.

This new feature improves the manageability of LogMiner. The task-based work flows enable log mining and are integrated with Flashback Transactions.

1.2.2 Data Guard Improvements

The features in the following sections describe enhancements to Oracle Data Guard.

1.2.2.1 Fast-Start Failover for Maximum Performance Mode in a Data Guard Configuration

This feature enables fast-start failover to be used in a Data Guard configuration that is set up in the maximum performance protection mode. Since there is some possibility of data loss when a Data Guard failover occurs in maximum performance mode, administrators can now choose not to do a fast-start failover if the redo loss exposure exceeds a certain amount.

This enhancement allows a larger number of disaster recovery configurations to take advantage of Data Guard's automatic failover feature.

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

1.2.2.2 Compression of Redo Traffic (Only for Gap Resolution) Over the Network in a Data Guard Configuration

Archived redo logs, if needed to be transmitted to the standby database in an Oracle Data Guard configuration, can now be compressed as they are sent over the network.

This feature improves the network bandwidth utilization in an Oracle Data Guard configuration.

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

1.2.2.3 Real-Time Query Capability of Physical Standby Database

It is now possible to query a physical standby database while Redo Apply is active.

This new capability increases your return on investment in Data Guard technology because a physical standby database can now be used to offload queries from the primary database in addition to providing data protection.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.2.4 Fast Role Transitions in a Data Guard Configuration

This feature improves the performance of Oracle Data Guard role transitions.

The benefit is increased flexibility and manageability of your disaster recovery configurations using Oracle Data Guard.

1.2.2.5 User Configurable Conditions to Initiate Fast-Start Failover in a Data Guard Configuration

This feature enables the administrator to choose and configure a list of conditions which, if they occur, will initiate fast-start failover.

This feature increases the flexibility and manageability of customers' disaster recovery configurations.

See Also:

Oracle Data Guard Broker for details

1.2.3 Data Guard Integration, Simplification, and Performance

The new features in the following sections simplify the configuration and use of Oracle Data Guard. For example, some features provide a smaller set of integrated parameters, a unified SQL/Broker syntax, and better integration with other High Availability features like RMAN and Oracle RAC. Other features enhance the performance of key Oracle Data Guard features like redo transport, gap resolution, switchover/failover times, and Logical Standby Apply.

1.2.3.1 Dynamic Setting of Oracle Data Guard SQL Apply Parameters

It is now possible to configure specific SQL Apply parameters without requiring SQL Apply to be restarted.

This ability improves the manageability, uptime, and automation of a Logical Standby configuration.

1.2.3.2 Enhanced Data Guard Broker Based Management Framework

The enhancements for this release include:

- Data Guard Broker improved logging and tracing

- OMF support for Data Guard Broker configuration files

- Data Guard Broker integration with database startup

- Data Guard Broker support for advanced redo transport settings

- Data Guard Broker support of prepared switchovers for Logical Standby

These enhancements make it possible to use Data Guard Broker in a wider variety of disaster recovery configurations.

See Also:

Oracle Data Guard Broker for details

1.2.3.3 Enhanced Data Guard Management Interface (Using SQL*Plus)

This feature makes it much simpler to use Oracle Data Guard administration-related SQL statements and initialization parameters.

The benefit of this feature is increased flexibility and manageability of disaster recovery configurations using Oracle Data Guard.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.3.4 Histogram for Redo Transport Response Time in a Data Guard Configuration

This feature advises on a suitable value for NET_TIMEOUT based on the response times for SYNC redo transport destinations.

The benefit is an increase in the flexibility and manageability of disaster recovery configurations using Data Guard.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.3.5 Snapshot Standby

A physical standby database can be open temporarily (that is, activated) for read/write activities such as reporting and testing. A physical standby database in such a state can still receive redo data from the primary database, thereby providing data protection for the primary database while still in the reporting database role.

This feature provides the combined benefit of disaster recovery, and reporting and testing using a physical standby database.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.3.6 Strong Authentication for Data Guard Redo Transport

Redo transport network sessions can now be authenticated using SSL. This provides strong authentication and makes the use of remote login password files optional in a Data Guard configuration.

The benefit is increased security of your disaster recovery configurations using Oracle Data Guard.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.3.7 Enhanced DDL Handling in Oracle Data Guard SQL Apply

SQL Apply executes in parallel (based on availability of parallel servers) DDLs that were executed with a parallel hint at the primary database.

1.2.3.8 Enhanced Oracle RAC Switchover Support for Logical Standby Databases

When switching over to a logical standby database where either the primary database or the standby database is using Oracle RAC, the SWITCHOVER command can be used without having to shut down any instance either at the primary or at the logical standby database.

1.2.4 Data Guard Logical Apply Completeness

The following sections describe features that add support for other database server features that are not currently handled by Oracle Data Guard. For example:

- LogMiner support for unsupported data types
- Objects and XML

1.2.4.1 Oracle Scheduler Support in Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for a primary database in which the DBMS_SCHEDULER package was executed.

This enhancement allows Data Guard SQL Apply (logical standby database) to be used for a wider variety of applications.

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

1.2.4.2 Fine-Grained Auditing (FGA) Support in Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for primary databases using Oracle Fine-Grained Auditing (FGA).

This new support allows Data Guard SQL Apply (logical standby database) to be used for a wider variety of applications. In this case, applications with security features built-in.

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

1.2.4.3 Support Transparent Data Encryption (TDE) with Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for the primary database tables with Transparent Data Encryption (TDE).

This new support allows Data Guard SQL Apply (logical standby database) to be used for a wider variety of applications. In the case, applications with security features built-in.

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

1.2.4.4 Support XMLType Data Type (Only CLOB) in Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for primary database tables with the XMLType data type (only the CLOB representation).

This new support allows Data Guard SQL Apply (logical standby database) to be used in a wider variety of applications.

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

1.2.4.5 Virtual Private Database (VPD) Support in Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for primary database tables with Oracle Virtual Private Database (VPD).

This new support allows Data Guard SQL Apply (logical standby database) to be used for a wider variety of applications. In this case, applications with security features built-in.

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

1.2.5 High Availability Improvements

The following sections describe general improvements in High Availability.

1.2.5.1 Flashback Transaction

Flashback transaction is a new feature that can easily back out a transaction and its dependent transactions. This recovery operation utilizes undo data to create and execute the corresponding, compensating transactions that revert the affected data back to its original state.

This feature increases availability during logical recovery by easily and quickly backing out a specific transaction or set of transactions and their dependent transactions, with one command, while the database remains online.

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for details

1.2.5.2 SMP Scalable Redo Apply

Redo Apply performance has been significantly improved as a result of some internal optimizations in media recovery. Wait event generation in media recovery has also been enhanced.

This feature enables faster performance of media recovery and also Data Guard Redo Apply (physical standby database), thereby enabling real-time disaster recovery for more voluminous and highly transactional databases.

1.2.5.3 Transportable Databases Between Linux and Windows

This feature enables simple moving of databases between Linux and Windows.

This results in transportable databases between Linux and Windows enabling cross-platform physical standby and simple moving of databases between Linux and Windows.

1.2.6 Information Life Cycle Management

The features described in the following sections provide automated and policy-based movement of data between storage tiers and they provide mechanisms for complying with retention and immutability regulations.

1.2.6.1 Flashback Data Archive

The historical data store provides the ability to store and track all transactional changes to a record for the duration of its lifetime.

The database maintains every state of the record during its lifetime. You no longer need to build this intelligence into the application. This feature is useful for compliance reporting and audit reports.

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for details

1.2.7 Integration, Simplification, and Performance of Availability Features

These features describe improvements to the interoperability, usability, performance and robustness of existing functionality.

1.2.7.1 Automatic Reporting of Corrupt Blocks

During instance recovery, if corrupt blocks are encountered, the DBA_CORRUPTION_ LIST is automatically populated.

Block validation occurs at every level of backup, media recovery, and instance recovery.

1.2.7.2 Block Media Recovery Performance Improvements

In prior releases, block media recovery needed to restore original block images from disk or tape backup before applying needed archived logs. In this release, if flashback logging is enabled and contains older, uncorrupted blocks of the corrupt blocks in question, then these blocks will be used, speeding up the recovery operation.

The benefit is a reduction in the time it takes for block media recovery by restoring block images from flashback logs instead of from disk or tape backups.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* for details

1.2.7.3 Creation and Recovery of Long-Term Backups Improved

Long-term backups created with the KEEP option only back up the archive logs needed to make the backup consistent. No further archive log backups are retained.

This feature reduces the archive log backup storage needed for online, long-term KEEP backups by only backing up the archive logs needed to recover the long-term backup.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* for details

1.2.7.4 End-to-End Redo Validation

This feature enhances redo validation.

End-to-end redo validation provides foreground redo checksum at generation time. This improves CPU and cache efficiency for the checksum computation.

1.2.7.5 Merge Catalogs

The new IMPORT CATALOG command allows one catalog schema to be merged into another, either the whole schema or just the metadata for specific database IDs.

This feature simplifies catalog management by allowing separate catalog schemas, created at different versions, to be merged into a single catalog schema.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* for details

1.2.7.6 Parallel Backup and Restore for Very Large Files

Backups of large data files now use multiple parallel server processes to efficiently distribute the workload for each file. This is especially useful for very large files.

This features improves the performance backups of large data files by parallelizing the workload for each file.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* for details

1.2.8 Intelligent Data Protection and Repair

The following sections describe features that automate the process of determining what kind of recovery is needed and applying that needed recovery.

1.2.8.1 Data Recovery Advisor

Data Recovery Advisor is a built-in tool that automatically diagnoses data failures and reports the appropriate repair options.

Data Recovery Advisor ensures the correct repair is performed for a failure and decreases the amount of time to recover from a failure.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* for details

1.2.8.2 Enhance LogMiner to Support Automatic Data Repair

This new feature adds support for Automatic Data Repair (ADR).

Automatic Data Repair (ADR) requires the ability to lookup metadata when the database is not available. This project enhances the existing functionality in LogMiner to support ADR.

1.2.8.3 Integrated Protection From Various Data Corruptions

Through the creation of a new initialization parameter, `DB_ULTRA_SAFE={off, data_only, data_and_index}`, this capability provides an integrated mechanism to offer protection from various possible data corruptions that may impact Oracle Database. This parameter controls the setting of other related parameters such as `DB_BLOCK_CHECKING`, `DB_BLOCK_CHECKSUM`, and `DB_LOST_WRITE_PROTECT`, and also controls other data protection behavior within Oracle Database (for example, requiring ASM to do sequential mirror writes).

This feature improves the data protection capabilities of Oracle Database by offering an integrated mechanism to control various levels of protection from data corruptions. By making it possible to detect data corruptions in a timely manner, this feature also provides critical high availability benefits for Oracle Database.

1.2.8.4 Lost-Write Detection Using a Physical Standby Database

Lost-write is a serious form of data corruption that can adversely impact a database. Lost-write occurs when an I/O subsystem acknowledges the completion of a block write in the database while, in fact, the write did not occur in the persistent storage. This arises mostly because of hardware or software malfunctioning. This feature enables the use of the physical standby database to detect data corruptions that could arise as a result of lost-writes on the primary database.

This feature allows for more comprehensive data corruption detection using a physical standby database, helping you get increased value from your disaster recovery investment and maintain the uptime requirements of your business.

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

1.2.9 Online Application Maintenance and Upgrade

The following features can significantly reduce the downtime required to make changes to an application's database objects.

1.2.9.1 DDL With the WAIT Option

Data definition language (DDL) commands require exclusive locks on internal structures. If DDL commands are issued, these locks may not be available causing the statement to immediately fail even though the DDL could have possibly succeeded sub-seconds later. This can be resolved by setting the WAIT option in the DDL_LOCK_ TIMEOUT initialization parameter.

The WAIT option is the new default. The wait time is specified instance-wide (in the initialization parameter file) and can be modified on a session level.

The WAIT option gives you more flexibility to define grace periods for such commands to succeed instead of raising an error right away, thus requiring additional application logic to handle such errors.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details

1.2.9.2 Enhanced ADD COLUMN Functionality

Default values of columns are maintained in the data dictionary for columns specified as NOT NULL.

Adding new columns with DEFAULT values and NOT NULL constraint 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 consumes no space.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details

1.2.9.3 Finer Grained Dependencies

In previous releases, metadata recorded mutual dependencies between objects with the granularity of the whole object. For example, PL/SQL unit P depends on PL/SQL unit Q or that view V depends on table T. This means that dependent objects were sometimes invalidated when there was no logical requirement to do so. For example, if view V depends only on columns C1, C2, and C3 in table T and a new column, C99, is

added, the validity of view `V` is not logically affected. Nevertheless, in earlier releases, `V` was invalidated by the addition of column `C99`.

Oracle Database 11*g* records dependency metatdata at a finer level of granularity so that the addition of `C99` does not invalidate view `V`. Similarly, if procedure `P` depends only on elements `E1` and `E2` in package `PKG`, then if element `E99` is added to `PKG`, procedure `P` is not invalidated. (In Oracle Database 10*g*, this change to `PKG` would invalidate procedure `P`.)

By reducing the consequential invalidation of dependent objects in response to changes in the objects they depend upon, application availability is increased. The benefit is felt both in the development environment and when a live application is parsed or upgraded. The benefit occurs when an Oracle Database patchset is applied because changes to schema objects are required to be compatible and, therefore, not cause consequential invalidations.

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

1.2.9.4 Invisible Indexes

An invisible index is an alternative to making an index unusable or even to drop it. An invisible index is maintained for any DML operation but is not used by the optimizer unless you explicitly specify the index with a hint.

Applications often have to be modified without being able to bring the complete application offline. Create invisible indexes temporarily for specialized non-standard operations, such as online application upgrades, without affecting the behavior of any existing application. Furthermore, invisible indexes can be used to test the removal of an index without dropping it right away, thus enabling a grace period for testing in production environments.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details

1.2.9.5 Minimize Dependent PL/SQL Recompilation After Online Table Redefinition

This feature minimizes the need to recompile dependent PL/SQL packages after an online table redefinition. If the redefinition does not logically affect the PL/SQL packages, recompilation is not needed. This optimization is on by default.

This feature reduces the time and effort to manually recompile dependent PL/SQL after an online table redefinition. This also includes views, synonyms, and other table dependent objects (with the exception of triggers) that are not logically affected by the redefinition.

1.2.9.6 Online Index Creation and Rebuild Enhancements

In highly concurrent environments, the requirement of acquiring a DML-blocking lock at the beginning and end of an online index creation and rebuild could lead to spikes of waiting DML operations and, therefore, a short drop and spike of system usage. While this is not an overall problem for the database, this anomaly in system usage could trigger operating system alarm levels. This feature eliminates the need for DML-blocking locks when creating or rebuilding an online index.

Online index creation and rebuild prior to this release required a DML-blocking lock at the beginning and end of the rebuild for a short period of time. This meant that there would be two points at which DML activity came to a halt. This DML-blocking lock is no longer required, making these online index operations fully transparent.

See Also:

Oracle Database Administrator's Guide for details

1.2.9.7 Online Redefinition for Tables with Materialized View Logs

In previous releases, a table could not be redefined if it had a log or materialized views defined.

With this feature, this restriction has been lifted.

You can now clone the materialized view log onto the interim just like triggers, indexes, and other similar dependent objects. At the end of the redefinition, ROWID logs are invalidated. Initially, all dependent materialized views need to do a complete refresh.

See Also:

Oracle Database Administrator's Guide for details

1.2.9.8 Read-Only Tables

Oracle Database 11*g* introduces new ALTER TABLE syntax. For example:

```
ALTER TABLE <name> READ ONLY
and
ALTER TABLE <name> READ WRITE
```

The operating system sets the precedent to make a file read-only even for its owner. Earlier, a table could be made read-only (by granting only SELECT on it) to users other than the owner of the table. With this feature, the owner too can be prevented from doing unintended DML to a table.

1.2.10 RMAN Integration, Simplification, and Performance

The following sections describe RMAN integration, simplification, and performance features for Oracle Database 11*g* Release 1 (11.1).

1.2.10.1 Archive Log Management Improvements

This feature provides the following enhancements:

- Ensure that archive logs are deleted only when not needed by required components (for example, Data Guard, Streams, and Flashback).

- In a Data Guard environment, allow all standby destinations to be considered where logs are applied (instead of just mandatory destinations), before marking archive logs to be deleted. This configuration is specified using CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY.

- Allow optional archive log destination to be utilized in the event that the flash recovery area is inaccessible during backup. Archive logs in this optional destination can be deleted using BACKUP .. DELETE INPUT or DELETE ARCHIVELOG.

This feature simplifies archive log management when used by multiple components. It also increases availability when backing up archive logs, when an archive log in the flash recovery area is missing or inaccessible. In this case, the backup will failover to an optional archive log destination to continue backing up the archive logs.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.10.2 Fast Incremental Backups on Physical Standby Database

You can enable block change tracking on a physical standby database. RMAN uses the change tracking file, on incremental backups, to quickly identify the changed blocks since the last incremental backup and to read and write just those blocks.

This feature enables faster incremental backups on a physical standby database than in previous releases.

1.2.10.3 Improved Backup Compression Performance

In this release, when creating compressed backup sets, performance will be noticeably faster than in previous releases.

The benefit is faster backup compression performance.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.10.4 Improved Integration with Data Guard

Improved integration with Data Guard includes the following:

- Persistent configurations can be set for each primary and standby database. You now have the ability to modify database-specific configurations in the catalog without having to connect to the specific primary or standby database.

- Allow conversion of a backup control file to a standby control file and vice versa. This eliminates the need to explicitly create a standby control file from the primary database.

- Share backups among primary and standby databases in a storage area network (SAN) by supporting restore to or from different mount points.

- Define which backups are available to particular databases, for restore operations, based on the network topology.

This feature simplifies backup and recovery operations across all primary and physical standby databases when using a catalog.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.10.5 Network-Aware DUPLICATE Command

A clone database on a remote site can now be easily created directly over the network with the enhanced `DUPLICATE` command without existing backups. ASM-to-ASM `DUPLICATE` over the network is also supported.

This feature eliminates the need to copy or move backups to the remote site before executing the `DUPLICATE` command. This reduces DBA time and effort, and eliminates storage for the additional copy at the remote site.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.10.6 Optimized Undo Backup

During backup, undo that is not needed for recovery of the backup (for example, for transactions that have already been committed), is not backed up. In past releases, all undo was backed up. This new feature is enabled with `CONFIGURE BACKUP UNDO OPTIMIZATION`. If you wish to preserve active undo within the `UNDO_RETENTION` period in the backup (for example, for future flashback query or flashback table operations), an option is available to only eliminate the inactive undo beyond the `UNDO_RETENTION` period.

The benefit is reduced overall backup time and storage by not backing up undo that applies to transactions that have already been committed by the time the backup starts.

1.2.10.7 Read-Only Transported Tablespaces Backup

Transported tablespaces can be backed up, whether in read-only or read/write mode. This feature removes the restriction in previous releases, that transported tablespaces must be enabled for read/write before making a backup.

The feature allows transported tablespaces to be immediately backed up, without having to be made read/write first.

> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* for details

1.3 Database Overall

The following sections describe overall database features for Oracle Database 11*g* Release 1 (11.1).

1.3.1 Utilities

The database utilities give you the ability to move large volumes of data and metadata into and between Oracle databases with a high degree of performance, reliability, and manageability. This area includes Upgrade/Downgrade, Data Pump Export and Import, the Metadata API, and the SQL*Loader utility.

1.3.1.1 Compressed Dump File Sets

Data Pump provides the ability to compress the metadata of a dump file set, which can shrink the dump files by 10-15%.

In this release, Data Pump adds the ability to compress the entire dump file set, including data and metadata.

This feature lets you specify compression for metadata, row data, or the entire dump file set.

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

1.3.1.2 Data Pump API Enhancements

The Data Pump API has been enhanced to allow individual worker processes to be restarted after a failure without terminating the Data Pump job. This enhancement improves availability when using the parallel capabilities of Data Pump.

The new `DATA_REMAP` API allows you to specify transformations to be applied to data during export or import. This can be useful when you need to obfuscate sensitive data while retaining its data type, structure, or format which may be needed when moving data from a production system to a test environment.

1.3.1.3 Database Upgrade/Downgrade Enhancements

Oracle Database 11*g* upgrade now includes enhanced information gathering and reporting facilities. The pre-upgrade information utility has improved resource estimations, and the post-upgrade status tool has improved error message generation and display.

Over the past three database releases, improvements have been made to simplify both manual upgrades and upgrades done using the Database Upgrade Assistant (DBUA). This feature further reduces the complexity of database upgrades and gives DBAs more information they can use to determine whether their database upgrade was successful.

> **See Also:**
>
> *Oracle Database Upgrade Guide* for details

1.3.1.4 Direct Path API Multiple Subtype Support

Prior to Oracle Database 11*g*, only one object type could be loaded into an object column or object table when using the Direct Path API.

This feature extends that support to allow loading of multiple subtypes into a column object or object table using Direct Path `INSERT`, thereby providing more robust support for objects and for XML schema data. Data Pump now supports the export and import of XML data.

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

1.3.1.5 Encrypted Dump File Sets

In this release, Data Pump adds the ability to encrypt the dump file set, including data and metadata. You can choose to encrypt part or all of the dump file set by specifying encryption for encrypted columns, metadata, row data, all, or none.

Customers want sensitive data residing in dump files to be secure.

1.3.1.6 Enhancements to Metadata API

The Metadata API heterogeneous object types have been enhanced to support all object types currently supported by original Export (exp). In particular, XML Schemas and schema-based tables are now supported.

Data Pump also supports editable XML. You can extract metadata from the dictionary, manipulate it, and then convert the metadata to DDL so that the object can be re-created on the same database or on another database.

The `TABLE_EXPORT` heterogeneous object type is enhanced to properly support AQ tables. This allows AQ tables to be exported. All dependent objects for the table are also exported.

The Metadata API provides the ability to transform a local index defined on a single partition into an index on a nonpartitioned table.

1.3.1.7 Oracle Data Pump External Table Enhancement

In previous releases of Data Pump, a row error in an external table caused the entire table load to abort, requiring you to reload the table.

This feature ensures that a row error does not cause a table load to abort, thus saving load time.

1.3.1.8 Single Partition Transportable for Oracle Data Pump

Until this release, the Transportable Tablespaces mechanism for Data Pump could be used to specify only the physical tablespaces to be exported. This feature adds a partition mode, which can be used to move one or more partitions or subpartitions of a table without having to move the entire table or exchange out the partition or subpartition.

In addition, partitions can be imported to the target database either as part of an existing table or as a separate table for each partition.

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

1.3.1.9 SQL*Plus BLOB Support

Queries in SQL*Plus now support BLOB columns.

This feature allows you to verify the contents of BLOB columns.

> **See Also:**
>
> *SQL*Plus User's Guide and Reference* for details

1.4 Business Intelligence and Data Warehousing

The following sections describe Business Intelligence and Data Warehousing features for Oracle Database 11*g* Release 1 (11.1)

1.4.1 Complete the Information Cycle

These features tighten the integration of operational and analytical databases, to enable higher quality and more up-to-date information.

1.4.1.1 Change Data Purge Enhancements

In earlier releases, purging the change table would always remove all consumed change data from the change table. The change data purge enhancements feature introduces a date range-based model to purge change data.

This feature increases the flexibility of purging change data from the change tables. Users can specify a date range for which data should be purged.

> **See Also:**
>
> *Oracle Database Data Warehousing Guide* for details

1.4.1.2 Change Subscription Enhancements

In this release, the change data subscription window can now be moved forward and backward.

This feature makes it easier to maintain a subscription window to change data. You are given more control over the definition of the subscription window.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.3 Enabling or Disabling Synchronous Change Data Capture

This feature adds an interface to enable users to turn synchronous Change Data Capture (CDC) off once it has been started, and turn it on when needed.

With this feature, users have more flexibility and control of the use of synchronous CDC to meet their needs.

1.4.1.4 Enhanced DML Tracking

The enhanced DML tracking feature enables change capture for all DML operations, including direct path loads and partition maintenance operations. A separate record in the change table indicates that such an operation has taken place. This feature also extends Change Data Capture support for data types including `CLOB`, `NCLOB`, and `BLOB` in the asynchronous modes.

Change Data Capture formerly did not always capture direct path load operations or implicit data changes as the result of partition maintenance operations. The enhanced DML tracking feature relieves this restriction.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.5 Enhanced Materialized View Catalog Views

New catalog views display the partition change tracking (PCT) information for a given materialized view by showing which sections of the materialized views data are fresh or stale.

This is a critical piece of information for the user to be able to view the partition staleness information of the materialized view. It affects the usability and maintainability of the materialized view.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.6 Query Rewrite Enhancement to Support Queries Containing Inline Views

In this release, query rewrite has been enhanced to support queries containing inline views.

With this feature, more queries are eligible for query rewrite thus improving system throughput and performance.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.7 Query Rewrite Support of Queries on Remote Tables

In this release, query rewrite has been enhanced to support queries when the tables are remote.

With this feature, more queries are eligible for query rewrite thus improving system throughput and performance.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.8 Refresh Performance Improvements

Refresh operations on materialized views are now faster with the following improvements:

- Refresh statement combinations (merge and delete)

- Removal of unnecessary refresh hint

- Index creation for `UNION ALL MV`

- PCT refresh possible for `UNION ALL MV`

With this feature, the time required to refresh materialized views is reduced.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.2 Enable the Information Grid

The new features in the following sections:

- Provide a single source for information, handling both operational and analytic workloads.

- Support database systems with hundreds of CPU's, hundreds of TBs, thousands of users, and dozens of applications.

1.4.2.1 Composite List-Hash Partitioning

Composite list-hash partitioning enables hash subpartitioning of a list-partitioned object to, for example, enable partition-wise joins.

Composite list-hash partitioning provides an additional modeling strategy to optimally map a business need to the partitioning of an object.

1.4.2.2 Composite List-List Partitioning

Composite list-list partitioning enables logical list partitioning along two dimensions. For example, list partition by `country_id` and list subpartition by `sales_channel`.

Composite list-list partitioning provides an additional modeling strategy to optimally map a business need to the partitioning of an object.

1.4.2.3 Composite List-Range Partitioning

Composite list-range partitioning enables logical range subpartitioning within a given list partitioning strategy. For example, list partition by `country_id` and range subpartition by `order_date`.

Composite list-range partitioning provides an additional modeling strategy to optimally map a business need to the partitioning of an object.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.4.2.4 Composite Range-Range Partitioning

Composite range-range partitioning enables logical range partitioning along two dimensions. For example, partition by `order_date` and range subpartition by `shipping_date`.

Composite range-range partitioning provides an additional modeling strategy to optimally map a business need to the partitioning of an object.

1.4.2.5 Interval Partitioning

This new partitioning strategy fully automates the partition creation for range. In other words, new partitions will be created when they are needed. By defining the interval criteria, the database knows when to create new partitions for new or modified data.

Managing the creation of new partitions can be a cumbersome and highly repetitive task. This is especially true for predictable additions of partitions covering small ranges, such as adding new daily partitions. Interval partitioning automates this operation by creating partitions on-demand.

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

1.4.2.6 Metadata for SQL Built-In Operators and Functions

In this release, metadata for all Oracle SQL built-in operations and functions has been made accessible through fixed tables in the dictionary.

This feature enables all third party tools to leverage any built-in Oracle SQL operation or function without having to maintain the function usage metadata redundantly in the application layer.

1.4.2.7 OLTP Table Compression

The Oracle compression feature allows heap tables to be stored in a compressed format, resulting in significant savings in disk storage, I/O, and redo logs. This feature extends the compression feature to OLTP environments so that compression works seamlessly for all DMLs.

Oracle's heap segment compression (HSC) has been well received in data warehousing environments. With this feature compression is extended to OLTP environments so that compression works well with all DMLs including conventional `INSERT`, `UPDATE`, and `DELETE`.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details

1.4.2.8 Reference Partitioning

The partitioning key is resolved through an existing parent-child relationship, enforced by active primary key or foreign key constraints.

The benefit of this feature is that tables with a parent-child relationship can be logically equi-partitioned by inheriting the partition key from the parent table without duplicating the key columns. The logical dependency will also automatically cascade partition maintenance operations, thus making application development easier and less error-prone.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.4.2.9 SQL PIVOT and UNPIVOT Operators

The `PIVOT` and `UNPIVOT` operators are extensions to the table expression in the `FROM` clause of a `SELECT` statement. `PIVOT` spreads values from multiple rows into multiple columns, aggregating data in the process. PIVOT is commonly used to create a result set with more columns and fewer rows than the source data. The `PIVOT` operator supports multiple pivot columns, multiple aggregates, wildcards, and aliasing. `UNPIVOT` moves data in the opposite direction from `PIVOT`. For each input row, `UNPIVOT` moves values from multiple columns into multiple output rows. The process creates a result set with fewer columns and more rows than the source data. `UNPIVOT` supports multiple unpivot columns, multiple measure columns, and aliasing.

`PIVOT` can create aggregated cross-tabular output that condenses many rows into a compact result set. For example, input data holding sales of 1 month per row can be pivoted into output holding 12 months per row, with each month in its own column. Another use of `PIVOT` is to combine multiple input rows into a single output row, enabling inter-row comparison without a table self-join. `UNPIVOT` reshapes data into a format useful for further relational operations. For example, if a source data set presents 12 months of sales values per row, `UNPIVOT` can reshape each source row into 12 output rows, each holding 1 month of sales data. The unpivoted results can, in turn, be manipulated with much simpler and more efficient SQL than required for the source data set.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.2.10 System Partitioning

System partitioning enables application-controlled partitioning for arbitrary tables and indexes. The database simply provides the ability to break down an object into partitions without any data placement rules. All aspects of data placement and retrieval are controlled by the application.

System partitioning provides the well-known benefits of partitioning (scalability, availability, and manageability), but the partitioning and actual data placement are controlled by the application.

See Also:

Oracle Database Data Cartridge Developer's Guide for details

1.4.2.11 Virtual Columns

Virtual columns are defined by evaluating an expression the results of which become the metadata of the columns for tables. Virtual columns can be defined at table creation or modification time.

Virtual columns enable application developers to define computations and transformations as the column (metadata) definition of tables without space consumption. This makes application development easier and less error-prone, as well as enhances query optimization by providing additional statistics to the optimizer for these virtual columns.

1.4.2.12 Virtual Column-Based Partitioning

In Oracle Database 11*g*, you can now partition key columns defined on virtual columns of a table.

Frequently, business requirements to logically partition objects does not match existing columns in a one-to-one manner. Oracle partitioning has been enhanced to allow a partitioning strategy being defined on virtual columns, thus enabling a more comprehensive match of the business requirements.

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

1.4.3 OLAP in Every Data Warehouse

The following sections describe the integration of OLAP and relational technology to broaden the appeal of OLAP for relational data warehouses, and also extend core OLAP capabilities.

1.4.3.1 Cost-Based Aggregation of OLAP Cubes

Cost-based aggregation of cubes provides a statistics-based approach to determining the best data within the cube to preaggregate and store for both efficient updates of the cube and efficient query.

Cost-based aggregation provides the DBA with an easy-to-use method of optimizing the cube as a general purpose summary management solution. The DBA needs to indicate only the amount of resources that should be applied to the task of optimizing summary data. Cost-based aggregation automatically determines and executes the best summary management plan to support ad-hoc query and reporting of the cube. This results in shorter refresh times and better query times.

1.4.3.2 Cube Organized Materialized Views

Cube organized materialized views represent the cube to SQL-based applications as materialized views that can be used for both refresh and query rewrite. Cube organized views are metadata objects that access data stored or calculated with the OLAP cube; they do not replicate data from the cube into new storage.

Cube organized materialized views can improve the ease and efficiency of dimension and cube updates and improve query performance.

Cube organized materialized views allow the database administrator to manage OLAP dimension and cube updates using the materialized view refresh system of Oracle Database, thus leveraging existing DBA skills and incremental data loading using materialized view log tables.

Cube organized materialized views can improve query performance by allowing SQL-based applications querying detail relational tables to automatically and transparently access summary data in the OLAP cube.

> **See Also:**
>
> *Oracle OLAP User's Guide* for details

1.4.3.3 Fine-Grained Partitioning of OLAP Cubes

Scalability improvements in cube partitioning increase the number of partitions that can be efficiently maintained in a cube.

Fine-grained partitioning of OLAP cubes provide support for larger cubes and more efficient cube update processing. Fine-grained partitioning allows the cube to be partitioned by more members of a dimension (as compared to Oracle Database 10*g* Release 2). Support for many more partitions can be used to create larger cubes (for example, retaining more time periods). Smaller partitions can be processed more quickly during a cube update.

1.4.3.4 OLAP Security Enhancements

Security of OLAP cubes and dimensions is enhanced to include support for Oracle Database object privileges (`SELECT`, `INSERT`, `UPDATE` and `DELETE`) on analytic workspaces, cubes and dimensions. For example, `GRANT SELECT ON sales_cube`.

Fine-grained access control to OLAP dimensions and cubes, which provides support for granting access privileges (SELECT, INSERT, UPDATE and DELETE) on data within a dimension and cube, is also supported in OLAP 11*g* and Analytic Workspace Manager.

For the general OLAP or database user, this collection of features enhances the integration of multidimensional data types with standard operational features of the database with the goal of making multidimensional data types more secure and easier to manage.

> **See Also:**
>
> *Oracle OLAP User's Guide* for details

1.4.3.5 SQL Optimizer Support for OLAP Option Cubes and Dimensions

The SQL Optimizer adds support for OLAP cube and dimension statistics and for pushing joins between cubes and dimensions into the OLAP cube. Optimizations also include efficient joins between relational tables and dimension and cube views.

SQL Optimizer support for OLAP cubes and dimensions provides general performance improvements and support for a wider variety of SQL (including counts and joins) to be efficiently executed against the cube. The main benefit is that a wider variety of SQL-based tools and applications will be able to easily query the cube.

> **See Also:**
>
> *Oracle Database PL/SQL Packages and Types Reference* for details

1.4.3.6 Storage and Access Improvements of Compressed Cubes

This feature provides internal scalability and performance improvements of compressed cubes.

This feature supports larger compressed cubes and broadens the applicability of compressed cubes to include a wider variety of data sets.

1.4.3.7 Unified OLAP API Metadata Change Management

Unified OLAP API metadata change management involves initial deployment of Analytic Workspace (AW) dimension objects, dependency tracking for these objects, and merging new versions of AWs and AW metadata with site-specific customizations.

Unified OLAP API metadata change management allows you to make site-specific modifications to the dimensional model. This feature is primarily in support of Oracle applications.

1.4.4 Predictive Analytics and Data Mining

Oracle Database 11*g* simplifies the management of data mining processing, increases the integration of data mining with SQL, and introduces new analytic capabilities.

1.4.4.1 Automated and Embedded Data Transformations for Mining (Supermodels)

This new feature in Oracle Database 11*g* can:

- Perform automatic data preparation for data mining models to reduce the burden on the user and provide an improved out-of-the-box experience (supermodels).

- Encapsulate the transformations with the model metadata to simplify model deployment (the transformations are automatically deployed with the model itself).

- Allow advanced users to specify some amount of transformations to the model build routine, which simplifies deployment for the advanced user as well.

This feature adds support to automate data preparation, the most difficult part of the data mining process. Another benefit is the simplification and automation of the overall data preparation process which simplifies model building and model scoring activities.

1.4.4.2 Data Mining Schema Objects

In Oracle Database 10*g* Release 2, Data Mining models were not defined in the database at the same class level as other DBMS objects. In Oracle Database 11*g*, models and the Data Mining environment are enhanced as follows:

- Eliminate the `DMSYS` schema, migrating necessary information to `SYS`

- Enhance existing catalog views to be Data Mining model aware

- Add new catalog views to expose Data Mining model metadata

- Add specific privileges for building, viewing, and scoring Data Mining models

Data Mining models are now more similar to other database objects. This allows you to assign access privileges and, in general, manage Data Mining models as full database objects.

1.4.4.3 Java API for Oracle Data Mining (JSR-73)

The Oracle Data Mining Java API provides algorithms and functions for new features in Oracle Data Mining 11*g*; for example, Multivariate Linear and Logistic Regression. Equivalent functionality between the SQL and PL/SQL APIs and the Java API is maintained in 11*g*.

Enhancements to the Oracle Data Mining Java API conform to the Java Data Mining standard (JSR-73).

> **See Also:**
>
> *Oracle Data Mining Application Developer's Guide* for details

1.4.4.4 Multivariate Linear Regression

Multivariate Linear Regression (MLinR) enables data analysts to build classic statistical models of the relationships between a range of input attributes and a target (numerical) output attribute. Additionally, MLinR finds the strength of the relationships between variables. Oracle Data Mining MLinR supports a large number of input attributes. This is a differentiator as most MLinR implementations cannot do this well. MLinR models can score data using the PREDICTION operators introduced in Oracle Database 10*g* Release 2. Several model diagnostics, both at global and attribute level, can be reported.

Oracle Database 11*g* augments the statistical capabilities available in Oracle Database by adding support for Multivariate Linear Regression in Oracle Data Mining.

1.4.4.5 Multivariate Logistic Regression

Multivariate Logistic Regression (MLogR) enables data analysts to build classic statistical models of the relationships between a range of input attributes and a target (categorical) output attribute. Additionally, MLogR finds the strength of the relationships between variables. The Oracle Data Mining Multivariate Linear Regression (MLinR) supports a large number of input attributes. This is a differentiator as most MLogR implementations cannot do this well. MLogR models can score data using the PREDICTION operators introduced in Oracle Database 10*g* Release 2. Several model diagnostics, both at global and attribute level, can be reported.

Oracle Database 11*g* Release 1 (11.1) augments the statistical capabilities available in the database by adding support for Multivariate Logistic Regression in Oracle Data Mining.

> **See Also:**
>
> *Oracle Data Mining Concepts* for details

1.4.4.6 Predictive Analytics: PROFILE

The Predictive Analytics (PA) package, introduced in Oracle Database 10*g* Release 2, provided a very simple interface for business users to leverage the power of Data Mining without the requisite statistical background. Also, in Oracle Database 10*g* Release 2, functions were provided to PREDICT (classification and regression) and to EXPLAIN (attribute importance). In Oracle Database 11*g*, a new Predictive Analytics (PA) method is added to develop profiles of a specified target attribute. PROFILE uses the decision tree algorithm.

PROFILE Predictive Analytics automatically sorts through the records and finds patterns and relationships of similar records (or customers) given a specified target attribute which designates a business problem. For example, PROFILE can be used to find the various profiles of customers who churn versus those who remain loyal. Similarly, PROFILE can be used to find the profiles of highly valued customers, customers who commit fraud, patients who contract a disease, and so on.

> **See Also:**
>
> *Oracle Data Mining Concepts* for details

1.4.4.7 SQL Prediction Enhancements

SQL built-in functions for Data Mining scoring (for example, PREDICTION) were added in Oracle Database 10*g* Release 2. Oracle Database 11*g* Release 1 is enhanced to:

- Allow cost matrix identification in the PREDICTION family to push down cost computations and provide a simple, best prediction by cost.

- Support additional algorithms and provide additional details in the GET_MODEL_ DETAILS_* functions.

- Include a new function, PREDICTION_BOUNDS, which provides the confidence bounds for generalized linear models.

> **See Also:**
>
> *Oracle Data Mining Application Developer's Guide* for details

1.5 Clustering

The following sections describe clustering features for Oracle Database 11*g* Release 1 (11.1).

1.5.1 Oracle Real Application Clusters Ease-of-Use

The following sections describe Oracle RAC enhancements that make clustering and Oracle RAC setup and installation easier to perform by anyone unfamiliar with both Oracle RAC and clustering.

1.5.1.1 Enhanced Oracle RAC Monitoring and Diagnostics in Enterprise Manager

This feature improves Oracle RAC performance monitoring and diagnostics through the GUI interface in Enterprise Manager. Specifically:

- Enhanced service-based monitoring.

- Summary and tile-based views for monitoring performance metrics.

This enhanced GUI interface eases management of Oracle RAC.

> **See Also:**
>
> *Oracle Database 2 Day + Real Application Clusters Guide* for details

1.5.1.2 Enhanced Oracle Real Application Clusters Configuration Assistants

This feature updates the graphical configuration utilities (DBCA, DBUA and NetCA) for all features of Oracle RAC. In addition, the rconfig utility is enhanced to convert single instance ASM to clustered ASM.

The benefit is a reduction in the management costs of cluster database environments through wizard-driven GUI management tools.

> **See Also:**
>
> Your platform-specific Oracle Real Application Clusters installation and configuration guide for details

1.5.1.3 OCI Runtime Connection Load Balancing

In Oracle RAC environments, session pools to services use service metrics to load balance sessions across different instances at runtime. OCI and OCCI applications have this feature enabled by default provided the necessary events are turned on.

Applications using OCI Session Pools have the best possible throughput for the resources provided. No application changes are required. Load balancing works toward the goal specified for the service. If an instance becomes hung or unresponsive, connections are directed to other instances that can provide better service.

See Also:

Oracle Call Interface Programmer's Guide for details

1.5.1.4 Parallel Execution for Oracle Real Application Clusters

Parallel execution is now aware of the service definition and automatically takes on the appropriate PARALLEL_INSTANCE_GROUP setting, thus making the explicit setting of PARALLEL_INSTANCE_GROUP for a services-controlled information grid environment unnecessary.

Oracle RAC services and the control of eligible nodes for parallel execution are now integrated and controlled by the services setup, making parallel execution fully transparent in an Oracle RAC services-controlled environment.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for details

1.5.1.5 Support for Distributed Transactions in an Oracle RAC Environment

An XA transaction can now span Oracle RAC instances. Individual branches have the ability to execute on different instances in the cluster.

This feature allows any application using XA to take full advantage of the Oracle RAC environment, thus enhancing high availability and scalability of the application.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.6 Content Management Services

The following sections describe enhancements to Content Management Services.

1.6.1 Oracle SecureFiles

The following sections describe enhancements to the LOB infrastructure in this release, called Oracle SecureFiles. Enhancements include built-in compression and encryption.

1.6.1.1 Fast Bulk Data Transfers

Oracle Database now uses a larger session data unit (SDU) by default. Optimizations to the Network Session SDU size result in significant performance gains for large data transfers.

This feature also takes advantage of zero copy and vectored I/O at the network level.

Fast bulk data transfers result in significant performance improvements for large data transfers.

See Also:

Oracle Database Net Services Administrator's Guide for details

1.6.1.2 LOB Prefetch

OCI now allows programs to set the amounts of prefetching of LOB data for optimized information retrieval, especially of smaller data sizes.

Business applications that often access smaller LOBs perform faster, because there are fewer round trips to the server for data.

> **See Also:**
>
> *Oracle Call Interface Programmer's Guide* for details

1.6.1.3 SecureFiles

This feature introduces a completely reengineered large object (LOB) data type to dramatically improve performance, manageability, and ease of application development. The new implementation also offers advanced, next-generation functionality such as intelligent compression and transparent encryption.

This feature significantly strengthens the native content management capabilities of Oracle Database.

> **See Also:**
>
> *Oracle Database SecureFiles and Large Objects Developer's Guide* for details

1.6.1.4 SecureFiles: Compression

This feature allows you to explicitly compress SecureFiles to gain disk, I/O, and redo logging savings.

The benefits of this feature are:

- Reduced costs due to the most efficient utilization of space.
- Improved performance of SecureFiles as compression reduces I/O and redo logging (at some CPU expense).

> **See Also:**
>
> *Oracle Database SecureFiles and Large Objects Developer's Guide* for details

1.6.1.5 SecureFiles: Deduplication

Oracle Database can now automatically detect duplicate SecureFiles LOB data and conserve space by storing only one copy.

This feature implements disk storage, I/O, and redo logging savings for SecureFiles.

> **See Also:**
>
> *Oracle Database SecureFiles and Large Objects Developer's Guide* for details

1.6.1.6 SecureFiles: Encryption

This feature introduces a new encryption facility for SecureFiles. The encrypted data is now stored in-place and is available for random reads and writes.

The benefit of this feature is enhanced data security.

See Also:

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

1.6.2 Text Manageability

The following sections describe enhancements to Text Manageability, which include:

- Improving ease-of-use and self-manageability of the text subsystem by providing Enterprise Manager support.

- Exposing the text subsystem as Text services through Java and Web for ease of application development.

1.6.2.1 Improved Support for Advanced Features in More Languages

Oracle Database 11*g* includes multilingual and linguistic support that improves Oracle Text handling of documents in different languages. Two new components are introduced:

- A new lexer that automatically detects the language of the document.

- A new document service that detects the character set and language of a document.

Oracle Text supports keyword searches in all Unicode languages. However, advanced features such as stemming and alternate spelling require lexing support in different languages. The new multilingual and linguistic support make advanced search available in 28 languages including Arabic, Hebrew, and Russian.

See Also:

Oracle Text Reference for details

1.6.2.2 Incremental Indexing Enhancements

This feature introduces three improvements that facilitate large text index creation:

- The new `ctx_ddl.populate_pending` interface
- New `sync_index` enhancements
- `NOPOPULATE` support

In large text warehouses, applications cannot afford to have the indexing process running continuously. This feature provides interfaces that let applications create large indexes in a manageable way.

See Also:

Oracle Text Reference for details

1.6.2.3 Oracle Enterprise Manager Support for Text

This feature provides improved Oracle Enterprise Manager support for Oracle Text.

DBAs can now administer Oracle Text from Oracle Enterprise Manager.

See Also:

Oracle Text Reference for details

1.6.2.4 Re-Create Index Online

This feature provides the ability to re-create an Oracle Text index without producing any undesirable query results until the application is ready to switch over to the modified index.

Users can now re-create an Oracle Text index with new preference values, while preserving base table DML and query capability during the re-creation. Users can re-create the index in one operation or can step through each stage of the re-creation manually.

> **See Also:**
>
> *Oracle Text Reference* for details

1.6.3 Text Performance and Scalability

The following sections describe feature integration with Oracle RAC for scalability and other subsystems such as the optimizer for performance.

1.6.3.1 Composite Index

This feature facilitates structured `ORDER BY` criteria, structured range, or combinations of both. This is accomplished by allowing the specification of `FILTER BY` and `ORDER BY` structured column(s) at index creation time.

This feature provides better performance for mixed queries involving relational and text predicates required by today's Web applications.

> **See Also:**
>
> *Oracle Text Reference* for details

1.6.3.2 More Types of Operations Allowed on Document Sections

This feature introduces a new type of document section called Structured/Sort Data (`SDATA`). The content of an `SDATA` section is typed and not tokenized. `SDATA` sections support range and equality query operations.

The benefit of this feature is faster queries on document metadata by pushing more metadata into the Text index.

> **See Also:**
>
> *Oracle Text Reference* for details

1.6.3.3 Text Support for Very Large Number of Partitions

Until now, the maximum number of partitions allowed has been 9,999. This limit has been increased to 1,223,054.

This increase is of significant benefit to Text users.

> **See Also:**
>
> *Oracle Text Application Developer's Guide* for details

1.6.3.4 User-Defined Score

User-defined scoring offers users a mechanism to define how the `CONTAINS` query will score textual content. This mechanism can use the `DEFINESCORE` or the `DEFINEMERGE` operator.

In some cases, the application would improve from more direct control of how to score documents based on structured values like date. The user-defined score feature allows applications to customize scoring of textual content.

> **See Also:**
>
> *Oracle Text Reference* for details

1.6.4 XML Content Management Platform

The following sections describe enhancements to XML-based Content Management.

1.6.4.1 Content Repository Management (JSR-170) Support for Java

This feature supports JSR-170 1.0 defined operations on both unstructured and semi-structured content and provides a comprehensive Oracle-proprietary content management platform, which includes:

- API extensions for access control, based on the DAV Access Control model
- Support for type definition and evolution, based on Oracle XML DB support for XML Schema
- Support for Oracle XML DB repository events

This feature provides a set of simple, consistent, and standard-based APIs to traverse, query, access, and operate on data in Oracle XML DB when building mid-tier content-centric applications. The feature streamlines the development of internet applications and encourages customer uptake and partner integration on Oracle XML DB.

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

1.6.5 XML Database

The following sections describe enhancements to the Oracle XML database technology.

1.6.5.1 DAV ACL Support

The DAV ACL standard is an industry standard for defining security in a WebDAV environment. Support for DAV ACL allows the XML DB repository to interact directly with other tools that understand the DAV ACL standard.

This feature enables interoperability between the Oracle XML DB repository and programs and tools that implement the DAV ACL standard. It enables the implementation of document-level security using industry standards, leading to reduced costs and increased productivity. Finally, it allows organizations to adopt authoring and security products that best meet their business needs.

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

1.6.5.2 Enable the XDB HTTP Server for SOA

This feature allows Oracle Database to be treated as simply another service provider in a service-oriented architecture (SOA) environment. The database is now capable of exposing PL/SQL packages, procedures, and functions as Web services. The database can also execute dynamic XQuery and SQL queries.

These enhancements reduce complexity and cost by allowing the database HTTP server to expose native Web services in an SOA environment without requiring any additional Oracle or third-party software components. The feature provides performance enhancements over previous architectures in cases where Oracle mid-tier software is already providing SOA capabilities on top of an Oracle Database.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.3 Large Text Node Handling

In earlier releases, nodes with an Oracle XML DB document were limited to 64K of text. In most cases this was not a problem, because 64K translates to approximately 32 pages of text. However, when an XML document has non-XML documents embedded inside it, this limitation can be a significant problem.

For example, a text node might contain an embedded HTML or RTF document such as an abstract or Web page. Or an XML document might contain embedded binary content, such as JPEG images, Word Documents, or PDF files.

The XML standard does not allow an XML document to contain binary content, so the content of the embedded documents has to be encoded into a character-based format using algorithms such as base64, binHex, or UUENCODE. The resulting in text node can easily exceed 64K.

This feature allows Oracle XML DB to handle this class of XML document by eliminating the current 64K limit on the size of a text node and by providing a streaming API that makes it possible to manipulate nodes that contain large amounts of text in a memory-efficient manner.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.4 Oracle XML DB Repository Triggers

This feature provides a series of events related to repository operations. Developers can attach code to these events in the same way that they can attach code to database triggers. This code can then perform application logic based on repository events in the same way that triggers allow application code to be executed based on operations on database triggers.

This feature provides a framework for developing intelligent content management systems based on the Oracle XML DB repository. It allows businesses to reduce costs and ensure compliance by allowing business logic to be associated with document-centric applications that make use of the Oracle XML DB repository. Organizations can ensure that they remain in compliance with document and content management regulations in legislation such as Sarbens/Oxley.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.5 Recursive Schema Handling

This feature allows Oracle XML DB to deliver improved performance execution of XPath-based operations on XML documents that contain highly recursive structures. It enables XPath Rewrite in cases where all of the predicates operate on nodes that can occur at an arbitrary depth within the recursive structure.

The ability to easily describe and manipulate highly recursive structures is a key benefit of the XML paradigm. This project results in significant performance improvements when XML schema-based documents of this nature are stored and manipulated using Oracle XML DB.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.6 Repository Performance Improvements

Performance is improved in the Oracle XML DB repository for complex folder-restricted queries.

This feature improves the response time for certain kinds of SQL operations on the Oracle XML DB repository.

1.6.5.7 Scalable XSL Output

Until now, the XSLT capabilities of XML DB could not be used to generate HTML or Text formatted documents. This feature allows XSLT processing to write directly to a CLOB or Output stream, thus removing this limitation and allowing the XML DB XSLT capabilities to be used in a much wider set of cases without requiring any additional infrastructure.

This feature helps organizations to reduce cost and complexity by generating large HTML and Text formatted documents directly from the database without requiring additional software components.

1.6.5.8 SQL/XML Standards Compliance and Performance Optimizations

This feature implements the XMLExists and XMLCast operators, which are part of the SQL:2005 standard. In addition, this feature:

- Adds support for the * operator with XMLForest(*).

- Adds support for complex storage parameters in the STORE AS CLOB clause of XMLType column DLL.

- Adds enhancements to the XMLAgg operator. The XMLAgg operator is used very commonly when generating XML from relation tables, or reconstructing XML from the internal object representation. XMLAgg is required whenever an element occurs multiple times within the document being generated. The proposed optimizations result in a significant reduction in the resources required to aggregate the target nodes, leading to noticeable performance improvements for these operations.

This feature provides significant performance improvements related to common XML generation operations. Reduced coding and maintenance costs result from a reduction in the amount of code required to perform common XML-centric tasks. Better support for relevant SQL standards enable the development of more portable application code.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.9 Support for More Types of Links in XDB

This feature adds support for a variety of links to Oracle XML DB, for example:

- Symbolic links to local XDB paths

- URL links

- Weak links (these store the OID, as do hard links, but do not affect persistence)

- Fixed links (these links can never be updated or deleted once created except by DDL operations)

- Pathname cache for fixed links

- Links as attributes

- The ability to globally disallow hard links to folders

- Preferred links

This variety of links enhances the XML Content Repository.

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

1.6.5.10 XLink Support in XML DB

Oracle Database 11*g* provides the XML equivalent of referential integrity between XML documents using well understood W3C standards. XBRL validation, XLink validation, and compound document support are all enabled.

This feature allows organizations to use the W3C XLink recommendation to define additional rules about the relationships between documents. XLink is a popular W3C standard that allows integrity rules and other relationships be specified in an XML-centric manner. The XLink specification is used by a number of important XML standards, including XBRL, the eXtensible Business Reporting Language, which is being adopted by a number of regulatory agencies worldwide, including the FDIC and SEC in the USA.

Support of XLink also offers organizations additional flexibility on how XML documents are stored in the database. For example, they can now store a Book as a set of Chapters, providing more granularity and flexibility.

An XLink-aware repository ensures that the rules specified in the XLink language are enforced while the content is stored in the Oracle XML DB repository, making the use of the XML language a more compelling story.

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

1.6.5.11 XML DB Operational Completeness

Oracle Streams now enables the propagation and management of data, transactions, and events in a data stream either within a database or from one database to another. The stream routes published information to subscribed destinations. The result provides greater functionality and flexibility than traditional solutions for capturing and managing events and for sharing the events with other databases and applications. As users' needs change, they can implement a new capability of Oracle Streams, without sacrificing existing capabilities.

Oracle Streams and Logical Standby can now be used in conjunction with the `XMLType` data type.

Oracle Streams delivers a common infrastructure that can meet the complex information sharing needs of today's organizations. These capabilities are now being extended so that they can be used in conjunction with all of the storage paradigms supported by the Oracle `XMLType` data type.

1.6.5.12 XML Query Optimization with Schemaless or Weakly-Typed Schema-Based XML

This feature enables cost-based optimizations of XQuery and XPath operations in cases where the underlying storage of some or all of the XML content is CLOB or CSX (binary XML). This results in significant performance improvements for queries over this kind of structure.

This feature allows organizations that cannot or do not want to leverage the schema-based storage model to improve the performance of their applications.

1.6.5.13 XML Translation Support

Operations on some classes of XML documents are now language-aware. As a result, the information returned by an operation on an XML document will be based on the users language preferences. This enhancement can significantly reduce the amount of code required to develop applications that are independent of a users language preferences. Until now, this capability was available only for documents stored using the Binary XML storage model.

This feature reduces costs associated with developing applications that are independent of the target users preferred language.

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

1.6.5.14 XML Update Performance Optimizations

This feature provides significant optimizations in the way in which UPDATE, DELETE, and INSERT operations on existing documents are implemented. The changes, which are totally transparent to the application, result in significant reductions on the amount of code that is executed internally and the amount of I/O and log that is generated when the structure of an existing document is updated.

The performance of update operations on XML content stored in the database is greatly improved.

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

1.6.5.15 XMLIndex

XMLIndex addresses the serious limitations inherent in the design and operation of the CTXSYS.CTXXPATH index. Among its benefits, XMLIndex:

- Does not require any prior knowledge of shape or content of the XML.

- Does not require any particular storage model for the XMLType.

- Can resolve any Xpath expression without requiring a subsequent filter of the result set identified by the index.

- Is transactional.

- Can support date and numeric range searching.

- Can support text-based searching of textnodes and attribute values through the ora:contains() XPath extension function.

This feature allows an organization to efficiently index XML documents without requiring explicit prior knowledge of shape or content of the XML that will be

indexed. It provides full support for extremely efficient XPath-based searching of the indexed XML content.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.16 XQuery and SQL/XML Performance Improvements

This feature implements performance improvements in a number of areas for SQL/XML and XQuery in the database.

XQuery and SQL/XML enhancements provide customers with performance gains when they upgrade to the current release.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.17 XQuery Standards Conformance

Oracle XQuery is in compliance with the final version of the W3C XQuery Standards and the JSR 225 XQJ standard.

Oracle XQuery standards compliance allows customers to build portable applications based on industry standards.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.18 XSLT Performance Improvements

This feature optimizes some classes of XSLT transformation by rewriting the operation into an equivalent XQuery expression and then executing the XQuery. The XQuery then becomes the subject of XQuery optimizations, which can result in set-based rather than iterative execution. This in turn can improve performance significantly.

XSLT transformation is commonly used to generate one format of XML from another format of XML. This feature provides significant performance improvements for some classes of XML to XML transformation.

See Also:

Oracle XML DB Developer's Guide for details

1.7 Fault Diagnosability

The following sections describe diagnosability features for Oracle Database 11*g* Release 1 (11.1).

1.7.1 Ease Diagnosis

The following sections describe new features for automatic information generation, packaging for problems or errors, and tools to ease diagnosis.

1.7.1.1 Automatic Inter-Component Deadlock Detection

Automatic inter-component deadlock detection automatically executes deadlock detection when such conditions are created or persist in the database.

This feature also provides the necessary infrastructure for other features in Oracle Database 11*g* that minimize database and application upgrade downtime.

See Also:

Oracle Database Performance Tuning Guide for details

1.7.1.2 Hang Manager

The Hang Manager automatically detects, analyzes, and dumps diagnostic information for hangs in Oracle Database environments, including Oracle RAC and ASM databases.

This feature simplifies database management by enabling you to more quickly resolve database hangs.

See Also:

Oracle Database Concepts for details

1.7.1.3 Incident Packaging Service (IPS)

Providing appropriate information to Oracle Support or Development is a tedious and sometimes time-consuming task for users. IPS provides a facility that extracts diagnostic and test case data associated with product exceptions (incidents) from Automatic Diagnostic Repository (ADR) and packages the data for transport to Oracle.

IPS also provides mechanisms to automatically generate test cases, such as SQL test cases, so that support and development personnel can easily reproduce the problems in-house for analysis and resolution.

See Also:

Oracle Database Administrator's Guide for details

1.7.1.4 SQL Test Case Builder

SQL Test Case Builder is a PL/SQL package that gathers all of the information at the customer site that is needed to reproduce a problem on a different system.

Obtaining a reproducible test case is the single most important factor in the speed of incident resolution. The SQL Test Case Builder gathers as much information as possible related to a SQL incident and packages it in a way that allows the problem to be reproduced.

See Also:

Oracle Database Performance Tuning Guide for details

1.7.2 First-Failure Capture

The following sections describe complete and efficient capture of diagnostic information during failures.

1.7.2.1 Automatic Diagnostic Repository

Automatic Diagnostic Repository (ADR) is a new system-managed repository for storing and organizing trace files and other error diagnostic data. ADR provides a comprehensive view of all the serious errors encountered by the database and maintains all relevant data needed for problem diagnostics and their eventual resolution.

Automatic Diagnostic Repository (ADR) provides a uniform and consistent mechanism to store, format, and locate all database diagnostic information. Customers can now correlate errors across various components such as Oracle RAC, Oracle Clusterware, OCI, Net, processes, and so forth. ADR automatically generates incidents for serious errors and provides incident management functionality. As a result, ADR significantly reduces time to problem resolution time for customers.

See Also:

Oracle Database Administrator's Guide for details

1.7.2.2 Enhanced Network Diagnosability

This feature enhances network diagnosability by consolidating Oracle Net diagnostics for various clients with the database infrastructure and presenting the diagnostic information in a standardized readable format. The network diagnostics information for Oracle products such as clients, application server, and database is stored in the Automatic Diagnostic Repository (ADR).

Oracle Net can now use the same diagnostic infrastructure as Oracle Database. This feature helps users diagnose Oracle Net related problems more easily than in earlier releases.

See Also:

Oracle Database Net Services Administrator's Guide for details

1.7.2.3 OCI Diagnosability

The following enhancements have been added in Oracle Database 11*g* to ease problem diagnosis in OCI applications:

- Record appropriate application, process, network, or other state information and dump it to log files when necessary.
- Add correlation information between client and server to diagnose problems easily.

The support for enhanced diagnosability mechanisms in OCI results in faster resolution of customer problems.

See Also:

Oracle Call Interface Programmer's Guide for details

1.7.3 Intelligent Resolution

These features describe the system-guided repair and resolution of problems.

1.7.3.1 Improved Block Corruption Detection

In earlier releases, RMAN-detected block corruptions were recorded in `V$DATABASE_BLOCK_CORRUPTION`. In Oracle Database 11*g*, live block corruptions are also recorded in that view. This view is automatically updated when corruptions are repaired; for example, using block media recovery or data file recovery.

This feature shortens the time to discover block corruptions.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.7.3.2 SQL Repair Advisor

When a problematic SQL statement is identified, the SQL Repair Advisor reloads the SQL and recompiles it in a more detailed diagnostic mode. The SQL Repair Advisor gathers more information on the incident and provides a workaround or recommendations to the DBA.

The SQL Repair Advisor is a new tool that helps DBAs diagnose problematic SQL statements and automatically provides workarounds for them.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details

1.7.4 Problem Prevention

The following sections describe features that are proactive in early problem detection to prevent catastrophes and limit damage to your system.

1.7.4.1 Automatic Health Monitoring

Automatic Health Monitoring proactively checks the health of the database and identifies any issues affecting the database. Problems are detected even before users run into them and before they cause widespread damage. The results of Automatic Health Monitoring are reflected in a simple meter (the Health Meter) designed to allow the DBA to quickly gauge the health of the database. With the help of Automatic Health Monitoring, a DBA can get a comprehensive picture of the current health of the database as well as help on how to remedy the problems affecting the database.

Automatic Health Monitoring proactively detects problems early in their life-cycle and notifies the DBA of the user impact and recommended action to take. Depending on the component or situation, Automatic Health Monitoring can either quarantine the offending component/resource or provide a fix to a problem. This feature helps businesses minimize their downtime and plan ahead for outages.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details

1.7.4.2 Automatic Quarantine of Corrupt Undo Segments

This feature automatically quarantines corrupt undo segments and prevents future transactions from using the same undo segments.

This enhancement limits corruption propagation and thus limits the damage caused on the system.

1.7.4.3 Fast ANALYZE: Finding Table-Index Corruptions

`ANALYZE VALIDATE CASCADE` operations are extremely expensive for large indexes. This feature introduces a hashing scheme for validating mismatch conditions.

The hashing is significantly faster than traditional cascade validation. The hash is designed so that both full table scans and index fast full scans produce the same hash value.

Many customers use `ANALYZE TABLE ... VALIDATE STRUCTURE CASCADE` statements to detect incompatibilities or corruptions between tables and indexes. `ANALYZE` operations in earlier releases took a long time for large tables and often could not complete in the diagnostics or maintenance window. With this feature, these `ANALYZE` operations now run much faster.

See Also:

Oracle Database Administrator's Guide for details

1.7.4.4 Holistic Undo Health Monitoring

This feature performs a number of health checks to determine the holistic health of the transactional engine. It proactively examines data and index blocks for transactional consistency and validates undo segments for possible corruption. Any corruption detected results in a proactive dump of all the redo related to the block.

This feature protects user data from software and hardware malfunction and helps prevent corruption propagation.

1.7.5 Support Workbench

The following sections describe features for Enterprise Manager's GUI workbench for customers and support to ease diagnosis and resolution of database errors.

1.7.5.1 Support Workbench

The Support Workbench provides an easy-to-use interface that presents database health-related incidents on the system to the DBAs in a timely manner along with information on how to manage the incidents. It also assists DBAs in viewing diagnostic information from multiple Oracle products (such as Net, clients, ASM, Oracle RAC, etc.), running health checks, packaging incident data to Oracle Support, and managing incidents.

The Support Workbench significantly reduces problem resolution time for customers by providing a simple workflow interface to view and diagnose incident data and package it for Oracle Support.

See Also:

Oracle Database 2 Day DBA for details

1.8 Information Integration

The following sections describe information integration features for Oracle Database 11*g* Release 1 (11.1).

1.8.1 Information Grid: Computation at Scale

The following sections describe support for:

- 2007-scale computation
- 2007 data center with very many nodes and databases
- Extremely large deployments without scaling costs linearly as the number of participants grows
- Extremely large data volumes and rapid provisioning of them
- Continuous Query Notification enhancements

1.8.1.1 Event Notification Grouping and Management

Event notifications can be grouped by a time interval, enabling a single batch notification for the group rather than multiple individual event notifications. Notifications may be retained for user-specified durations if the client is unavailable.

Notification configuration and management is more flexible.

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

1.8.1.2 Oracle Scheduler for Streams Jobs

Jobs scheduled for propagation and notification for Streams are now managed with the DBMS_SCHEDULER package.

This conversion to the standard Oracle Scheduler simplifies administration of Streams.

> **See Also:**
>
> *Oracle Streams Concepts and Administration* for details

1.8.1.3 Scalability for Streams Notifications

The notification server (EMON) is now a multiprocess server, consisting of 5 processes.

This feature enables large numbers of simultaneous notifications system-wide.

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

1.8.2 Information Grid: Heterogeneous Information

New features in the following sections describe:

- Virtualizing and provisioning information from many sources.

- Supporting many representations of information.

- Providing synchronous, replica management and change notification for many sources.

- Extending events to include implicit events in non-Oracle Database systems.

1.8.2.1 Messaging Gateway Service-Based Failover

Messaging Gateway agents can now be started by the Oracle Scheduler. The scheduler uses the Oracle RAC service feature, through which users can specify a primary instance and associated machine to host the Messaging Gateway agent. When the primary instance fails, the service fails over to another instance. The Messaging Gateway is started on the machine hosting that instance and connects to the instance on that node.

With this feature, you can now use Oracle RAC services to control where the Messaging Gateway will fail over, allowing for high availability without the need to make every database instance ready to run the Messaging Gateway agent.

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

1.8.2.2 Multiple Messaging Gateway Agents in Oracle RAC Environments

This feature provides support for running multiple Messaging Gateway agents on a single database. Users can now scale the messaging workload with the number of instances, by balancing the workload across multiple Oracle RAC instances.

With this feature, you can now build messaging solutions with greater throughput that scales across multiple Oracle RAC instances.

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

1.8.2.3 Simplified Messaging Gateway Propagation Job Configuration

A new enhanced PL/SQL API consolidates the propagation subscriber and the propagation schedule into a new propagation job.

It is now easier and less confusing to create and schedule a propagation job for a Messaging Gateway. A new API provides a more intuitive means for configuring these jobs.

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

1.8.2.4 Database Gateways: Performance Improvements

With this feature, you experience significant improvement when loading large volumes of data from the non-Oracle systems. Performance of the gateways also improves in general.

This feature improves the performance of Database Gateways.

> **See Also:**
>
> *Oracle Database Heterogeneous Connectivity User's Guide* for details

1.8.2.5 Database Gateway for Adabas

Oracle Database 11*g* now provides a Database Gateway for Adabas.

Database Gateway for Adabas allows you to integrate data from Adabas using standard Oracle SQL.

1.8.2.6 Database Gateway for IMS

Oracle Database 11*g* provides a Database Gateway for IMS.

Database Gateway for IMS enables a customer to integrate data from IMS using standard Oracle SQL.

1.8.2.7 Database Gateway for VSAM

Oracle Database 11*g* provides a Database Gateway for VSAM.

Database Gateway for VSAM allows a customer to integrate data from VSAM using standard Oracle SQL.

1.8.3 More Granular Change Notifications

The following section describes Continuous Query Notification enhancements.

1.8.3.1 Continuous Query Notification Enhancements

Continuous Query Notification has been enhanced to provide notifications when the result set for a query changes on the server. These notifications can be registered at the row-level, rather than just at the table-level. Thus, alerts will only be sent when changes to the specific rows selected have been modified.

Applications, such as Radio Frequency Identification (RFID), can now receive notifications directly from the database rather than build this functionality into the application.

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for details

1.8.4 Streams Support for XMLType and TDE

The following sections describe support for:

- Oracle Streams data types
- Logical Standby
- Transparent Data Encryption (TDE)

1.8.4.1 Additional Data Type Support for Streams

Streams provides support the XMLType data type and for Transparent Data Encryption (TDE).

Data can be shared more easily between applications and databases.

> **See Also:**
>
> *Oracle Streams Concepts and Administration* for details

1.9 Location Services and Specialty Data

The new features in the following sections describe:

- Extending Oracle leadership in managing spatial and multimedia content.
- Addressing the unique performance, scalability and application development needs of rapidly growing machine-generated data created by sensors, RFID, semantic web, medical, life sciences, satellite imaging, and remote sensing technologies.

> **Note:** The name Oracle *inter*Media has been changed to Oracle Multimedia in Oracle Database 11*g* Release 1 (11.1). The feature remains the same, only the name has changed. References to Oracle *inter*Media will be replaced with Oracle Multimedia, however some references to Oracle *inter*Media or *inter*Media may still appear in graphical user interfaces, code examples, and related documents in Oracle Database documentation library for 11*g* Release 1 (11.1).

1.9.1 New Capabilities for Management of Geospatial and Multimedia Data

This release further enhances industry-leading spatial and multimedia features used in internet, geospatial, and media-oriented applications. New Web services, XML services, and usability features support mainstream business applications.

Enhancements for three-dimensional data, very large network data models, and GeoRaster data models support business-critical applications in government, homeland security and defense, life sciences, utilities, energy exploration, transportation, urban planning, simulation, and geoengineering.

1.9.1.1 Spatial: 3-D Geometry, Surface, and Point Cloud Storage and Indexing Functions

Oracle Spatial provides native storage, querying, and retrieval for three-dimensional (3-D) data including points, lines, surfaces, triangulated irregular networks (TINs - an alternative to rasters), and point clouds. Spatial R-tree indexing now supports 3-D data; SQL operators and analysis functions for 3-D data are also provided.

Oracle Spatial supports very large 3-D data sets such as urban models, point clouds, and terrain models with open, secure, scalable, and improved data management for urban planning and design, government, homeland security, military, oil and gas exploration, transportation engineering, gaming and simulation, geoengineering, and LIDAR-based map production.

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

1.9.1.2 Spatial Web Services

Oracle Web Services now support the following XML-based geospatial Web Services standards on a variety of client technologies and platforms:

- OGC OpenLS 1.1

- Web Feature Service 1.0

- Web Feature Service - Transactional 1.0

- Catalogue Service 2.0

Oracle Database and Oracle Application Server provides security for these Web services standards including authorization, authentication, and transport confidentiality and integrity. Java and PL/SQL client APIs are provided.

Oracle Spatial Web Services are location-ready and tightly integrated with Oracle Database and Oracle Application Server to provide a secure, transactional, service-oriented architecture platform to access and incorporate any published services for routing, geocoding, directory, catalog, feature, and mapping.

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

1.9.1.3 Spatial Routing Engine Enhancements

The Oracle Spatial routing engine now generates driving directions in German, French, Spanish, and Italian, and in turn-specific geometries.

The Oracle Spatial routing engine provides driving directions in various Western European languages and turning diagrams for linear asset management and travel and transport applications.

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

1.9.1.4 Spatial Network Data Model Enhancements

The Oracle Spatial network data model supports multiple concurrent Application Server requests with its thread-safe Java API and supports an XML interface for Web services queries.

New analysis and modeling capabilities include database-level handling of user or application-specific attributes. These features enable the network data model to manage nonconnectivity-related information. You can also selectively extract a subset of the network using SQL-like filters. Other enhancements include path arithmetic support such as path addition, subtraction, intersection, comparison, and duration modeling.

The Oracle Spatial network data model provides Java and XML APIs for Service Oriented Architectures and high query volume applications such as field service scheduling and logistics.

New analysis and modeling capabilities are provided for applications that analyze very large networks with complex attributes and relationships such as utilities, logistics, transportation, life sciences, and web information management.

> **See Also:**
>
> *Oracle Spatial Topology and Network Data Models Developer's Guide* for details

1.9.1.5 Load-On-Demand for Very Large Networks in Oracle Spatial Network Data Model

Oracle Spatial now supports load-on-demand for partitioned networks, which eliminates physical memory as a limitation for in-memory analysis. Large networks can be partitioned into manageable subnetworks and incrementally loaded into memory as needed for performance and scalable analysis. Partitioning utilities are available for partitioning large spatial networks.

Oracle Spatial network data model can analyze very large networks without loading the entire network into memory. This ability benefits applications such as utility and street networks.

> **See Also:**
>
> *Oracle Spatial Topology and Network Data Models Developer's Guide* for details

1.9.1.6 Spatial GeoRaster: Manageability, Reliability, Usability Enhancements

GeoRaster DML triggers are created and monitored by the system automatically. New tools automate GeoRaster upgrade, migration, and use. GeoRaster has support for Workspace Manager long transactions and fine-grained Label Security.

More metadata and data types are now supported. GeoRaster supports a generic polynomial georeferencing model, which includes direct linear transportation and rapid positioning capability georeferencing for nonrectified airborne photos and satellite images. It also supports bitmap masks, multiple NODATA values and value ranges, and has limited data type support.

Advanced operations have been added for advanced mosaic, GeoRaster object or layer union, GeoRaster templates, and related functions. GeoRaster also fully supports load and export of GeoTIFF geometadata and DigitalGlobe RPC file format.

Additional raster processing capabilities include GeoTIFF geometadata, loading, and exporting.

Oracle Spatial GeoRaster supports more data types and file formats, editing of image and grid data with long transaction versioning, fine-grained security, and new management tools and functionality.

1.9.1.7 Enhanced Workspace Manager Performance and Data Storage Options

User-specified optimizer hints on Workspace Manager operations are now supported. Partition operations (such as `ADD`, `DELETE`, `MERGE`, and `EXCHANGE`) are supported on a versioned table. New online operations on versioned tables include `ALTER INDEX` and `ALTER TABLE` operations, including moving a table to another tablespace and changing the percent free space. Row-level locking is implemented for merge and refresh operations. Compression is optional for the RemoveWorkspace operation.

This release of Workspace Manager supports greater tuning of queries for workspace operations and more physical storage options. Merge and refresh operations can be performed faster by executing them in parallel and making garbage collection optional if the workspace is removed.

1.9.1.8 Enhanced Workspace Manager Security and Administration

Oracle Label Security policies can now be set and altered on a versioned table. Changes to tables involved in a referential integrity constraint can be merged as a group. The name of the workspace that creates and retires a history row is recorded.

Workspace Manager allows Oracle Label Security policies to be applied and altered for long transactions. Changes to groups of tables can be merged as a set. Changes to data can be tracked by project.

1.9.1.9 Improved Workspace Manager Valid Time Usability

In this release, the unique `ROWID` for history rows with a Valid Time is available to query. Initial Valid Time timestamp values can be specified when a table is versioned. Valid Time timestamps can be presented as `TIMESTAMP WITH TIME ZONE` scalar type as an alternative to using the `WM_PERIOD` object type. Changes to Valid Time timestamps are now allowed for an Oracle Spatial feature stored in a parent table of a referential integrity constraint.

Valid Time, also known as effective dating, is a capability of Workspace Manager that allows versioned data to be stored with an associated time range that indicates when the data is valid. This feature includes several usability enhancements. A specific historical change with Valid Time can be presented to a user. Valid Time timestamp values can be initialized with user-defined values when a table is versioned and timestamps can be used as simple scalar values in a Microsoft .Net environment. Valid Time timestamps can be updated anytime an Oracle Spatial feature participates in a referential integrity constraint.

1.9.1.10 Oracle Multimedia Performance and Scalability

Earlier versions of Oracle *inter*Media (now called Oracle Multimedia) object types (`ORDImage`, `ORDAudio`, `ORDVideo`, and `ORDDoc`) allowed storage of electronic media objects of up to 4 Gbytes. In Oracle Database 11*g*, this media size limit is extended to the `BLOB` size limit, between 8 and 128 terabytes.

In addition, Oracle Multimedia uses less memory for scale-down and thumbnail generation functions, raising the practical limit for images to be processed in this manner.

Applications in the security, media and entertainment, medical, and life sciences markets are generating more detailed and higher fidelity digital media. Advances in capture technology and reductions in the cost of storage present the need to organize more and larger digital images, audio, and video. In the past, individual digital pictures, audio, and video rarely exceeded a gigabyte. With Oracle Database 11*g*, Oracle Multimedia supports emerging applications that may require tens of gigabytes for a single multimedia file.

> **See Also:**
>
> *Oracle Multimedia User's Guide* for details

1.9.2 New Capabilities for Medical Data

This following sections describe new features for the management of DICOM format medical data for health care and life sciences imagery.

1.9.2.1 Multimedia DICOM Medical Imaging Format Support

Oracle Multimedia now fully supports the format universally recognized as the standard for medical imaging, Digital Imaging and Communications in Medicine (DICOM) version 3. Applications can now use Multimedia JAVA and PL/SQL APIs to store, manage, and manipulate DICOM content.

It is now quite reasonable to build large archives of medical content such as single frame and multiframe images, waveforms, slices of 3-D volumes, video segments, and structured reports that are managed and secured using powerful Oracle Database tools. Complete DICOM metadata support makes it very easy to index and search DICOM content for research. Central storage of DICOM content makes telemedicine practical. Incorporating DICOM content in a database makes it feasible to build electronic health care records applications using powerful application development tools from Oracle or other third-party vendors.

> **See Also:**
>
> *Oracle Multimedia DICOM Developer's Guide* for details

1.9.2.2 ORDDicom Object Type

A new Multimedia object type, `ORDDicom`, has been defined to hold the data associated with DICOM content such as single or multiframe images, waveforms, or structured reports, and to implement the methods to manipulate the DICOM content. A Java proxy class, `OrdDicom`, is also defined. It provides access to the `ORDDicom` database object through JDBC in a Java application. For applications that already store DICOM content directly in `BLOB` or `BFILE`, a relational interface is also provided.

By presenting DICOM content stored in a database as objects, Oracle enables both rapid application development and easy, secure management of large medical archives.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.3 DICOM Metadata Extraction

In the initial release of Oracle Multimedia DICOM support, the most important metadata tags associated with DICOM content could be extracted into an XML document that could then be indexed and searched to find DICOM content that matched certain conditions. With Oracle Database 11*g*, that support is extended to offer complete and extensible metadata extraction. Customers can use either the supplied DICOM mapping document to extract metadata into an XML document, or they can create their own mapping documents to include support for private tags or subsets of the standard DICOM metadata tags. The extracted metadata can then be stored in a table to facilitate DICOM content searching based on standard or private DICOM attributes.

With this significant enhancement to the DICOM metadata extraction ability, customers can build large archives of DICOM content that will prove invaluable for research. By customizing DICOM mapping documents, customers can easily create highly specialized indexes to DICOM content such as single and multiframe images, waveforms, and structured reports based on both standard and private metadata tags.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.4 DICOM Conformance Validation

Given DICOM content and a set of user-specified conformance rules, Oracle Multimedia can verify that the DICOM content adheres to the conformance rules.

DICOM content is produced by many modalities. Most content conforms to the DICOM standard, but inevitably some does not. It is worthwhile to be able to identify DICOM content that does not conform to the standard. Validating DICOM content for conformance can ensure the consistency of a DICOM archive. It enables a database to accept DICOM content from multiple sources and to verify the integrity of the DICOM metadata.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.5 DICOM Image Processing

Oracle Multimedia includes the `ORDDICOM` data type to natively support imagery produced by medical imaging devices. This release adds methods and functions to copy and convert images from DICOM to JPEG, GIF, PNG, TIFF, and other formats and to generate scaled versions and thumbnails.

In order to view medical images stored in the DICOM format in Web applications, the image must be reformatted and converted into a browser-ready format. Oracle Database 11*g* provides methods to reformat and deliver DICOM images to applications that require popular industry-standard image formats.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.6 Making DICOM Content Anonymous

Oracle Database 11*g* offers a `makeAnonymous()` method which can be used to ensure that users of a DICOM medical archive see only the metadata that they are authorized to see.

Rules for making DICOM content anonymous can be customized in an anonymity document that specifies the set of attributes that should be made anonymous and how each should be obfuscated.

Clinicians need to be able to view all the metadata included in DICOM content for a patient being treated. Researchers need to be able to see the same DICOM content but, due to privacy regulations, must not be able to see any personally identifying information from that same DICOM content used.

By providing anonymity services in the database, Oracle Multimedia allows appropriate access for different classes of users of DICOM archives regardless of the application used to access the data.

> **See Also:**
>
> *Oracle Multimedia DICOM Developer's Guide* for details

1.9.2.7 Creation of DICOM Content

This release of Oracle Multimedia includes the ability to generate new DICOM content by combining digital images with an XML representation of the associated DICOM metadata. The result of this operation, well-formed and validated DICOM content, can be stored in a table in the database or delivered to a DICOM viewer.

Storing and retrieving paper- or film-based medical images is expensive and prone to error. However, many old images must be retained for a very long time and might be useful for both research and clinical purposes. By storing scanned images with the known metadata about those images in DICOM format, non-DICOM images become useful again and both storage and retrieval costs are reduced. New DICOM format content can also be generated to correct metadata errors in the original DICOM content using this technique.

> **See Also:**
>
> *Oracle Multimedia DICOM Developer's Guide* for details

1.9.2.8 Runtime Updatable DICOM Data Model

The runtime behavior of Oracle Multimedia DICOM support is determined by a set of user-configurable documents. This set of documents is collectively managed by a DICOM data model repository. An administrator can modify this data model repository to configure DICOM features for a particular database instance. With this design, a customer can upgrade DICOM features at any time, without interfering with an active DICOM archive.

Hospitals need their DICOM archives to be operational 24x7x365. They cannot afford to take down the system to update to a new version of the DICOM standard, to incorporate private tags for a new piece of equipment, to change their DICOM conformance rules, or to modify the set of tags they extract from DICOM content or the XML document they produce from the extracted tags. The data model repository implemented by this feature means that users can update their systems to accommodate new equipment and standards without interrupting the operation of the archive.

> **See Also:**
>
> *Oracle Multimedia DICOM Developer's Guide* for details

1.9.3 New Capabilities for RFID

This following section describes management of Sensor and RFID data.

1.9.3.1 RFID Identity Code Data Type for EPC, DoD, and Custom Sensor Tags

A new Identity Code object type is implemented in Oracle Database to support custom, Electronic Product Code (EPC), and Department of Defense standard tag formats. Metadata is used to define rules for encoding and decoding any tag information into Identity Code objects. All EPC standard tag types and metadata standards are supported.

With this new feature, application developers can define and process identity code tags faster and with less programming in Oracle Database 11*g*.

> **See Also:**
>
> *Oracle Database Advanced Application Developer's Guide* for details

1.9.4 New Capabilities for Semantic Data

This following sections describe new features and capabilities for semantic data.

1.9.4.1 Improved Performance for Bulk Loading

Oracle provides native storage, inference, and querying of semantic data sets often containing hundreds of millions of triples (modeled in canonical Resource Description Framework (RDF) <Subject Predicate Object> format). A new bulk loading utility is now introduced that significantly improves the ability to handle large volumes of triples. Query performance on semantic data has been improved, especially for queries returning large result sets, using a query rewrite technique and Oracle Database optimizer. Support for typed literals in semantic data has been enhanced to include xsd:date and xsd:time. Also, xsd:dateTime with time zone is now supported.

Oracle Database provides scalable, secure, integrated, and efficient support to store, inference, and query large semantic data sets described using W3C standards. Performance for loading and querying of semantic data has been improved to enable scaling for large data sets used in the defense and intelligence, life sciences, and geospatial domains. Date and time information, optionally with time zone, can be stored and queried.

> **See Also:**
>
> *Oracle Database Semantic Technologies Developer's Guide* for details

1.9.4.2 Support for Storage and Query of Semantic Content

Oracle Database 11*g* extends its semantic capabilities with native, lightweight Web Ontology Language (OWL) inferencing that is a practical subset of the OWL-DL standard. Ontologies (sets of terms, associated properties, and the relationships among them) can be stored in the database to enable searching based on relationships described in the ontology using new operators, SEM_RELATED and SEM_DISTANCE.

Inferencing support includes efficient and scalable reasoning for a subset of OWL-DL constructs and APIs to generate proofs for inferred triples and to detect inconsistency in semantic data sets. Advanced users can develop custom inferencing rules.

Semantic operators can be used for filtering based on semantic relatedness (`SEM_RELATED`), and the results can be further restricted or ordered using proximity measures (`SEM_DISTANCE`). A new index type (`SEM_INDEXTYPE`) allows efficient execution of such queries, enabling scalable performance over large data sets.

These new semantic operators enable Oracle Database to query relational data not only through keyword matching but also using concepts and terms related to the keyword. These ontology-assisted queries are based on semantic relationships between the column value data and ontology terms. This enables more complete search results without requiring as much prior knowledge of the data set. The OWL inferencing capability enables discovery of new relationships in RDF and OWL data. This is useful in applications in life sciences, health care, and business enterprise information integration. The W3C has adopted RDF, RDFS, and OWL as standards to represent semantic data.

> **See Also:**
>
> *Oracle Database Semantic Technologies Developer's Guide* for details

1.10 Manageability

The following sections describe server manageability features for Oracle Database 11*g* Release 1 (11.1).

1.10.1 Automatic Storage Management

The new features in Automatic Storage Management (ASM) extend the storage management automation, improve scalability, and further simplify management for Oracle Database files.

1.10.1.1 ASM Fast Mirror Resync

A new SQL statement, `ALTER DISKGROUP ... DISK ONLINE`, can be executed after a failed disk has been repaired. The command first brings the disk online for writes so that no new writes are missed. Subsequently, it initiates a copy of all extents marked as stale on a disk from their redundant copies.

This feature significantly reduces the time it takes to repair a failed diskgroup, potentially from hours to minutes. The repair time is proportional to the number of extents that have been written to or modified since the failure.

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

1.10.1.2 ASM Manageability Enhancements

The new storage administration features for ASM manageability include the following:

- New attributes for disk group compatibility

  To enable some of the new ASM features, you can use two new disk group compatibility attributes, `compatible.rdbms` and `compatible.asm`. These attributes specify the minimum software version that is required to use disk groups for the database and for ASM, respectively. This feature enables heterogeneous environments with disk groups from both Oracle Database 10*g* and Oracle Database 11*g*. By default, both attributes are set to 10.1. You must advance these attributes to take advantage of the new features.

■ New ASM command-line utility (ASMCMD) commands and options

ASMCMD allows ASM disk identification, disk bad block repair, and backup and restore operations in your ASM environment for faster recovery.

■ ASM fast rebalance

Rebalance operations that occur while a disk group is in RESTRICTED mode eliminate the lock and unlock extent map messaging between ASM instances in Oracle RAC environments, thus improving overall rebalance throughput.

This collection of ASM management features simplifies and automates storage management for Oracle databases.

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

1.10.1.3 ASM Preferred Mirror Read

When ASM failure groups are defined, ASM can now read from the extent that is closest to it, rather than always reading the primary copy. A new initialization parameter, ASM_PREFERRED_READ_FAILURE_GROUPS, lets the ASM administrator specify a list of failure group names that contain the preferred read disks for each node in a cluster.

In an extended cluster configuration, reading from a local copy provides a great performance advantage. Every node can read from its local diskgroup (failure group), resulting in higher efficiency and performance and reduced network traffic.

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

1.10.1.4 ASM Rolling Upgrade

Rolling upgrade is the ability of clustered software to function when one or more of the nodes in the cluster are at different software versions. The various versions of the software can still communicate with each other and provide a single system image. The rolling upgrade capability will be available when upgrading from Oracle Database 11*g* Release 1 (11.1).

This feature allows independent nodes of an ASM cluster to be migrated or patched without affecting the availability of the database. Rolling upgrade provides higher uptime and graceful migration to new releases.

> **See Also:**
>
> *Oracle Real Application Clusters Installation Guide for Linux and UNIX* for details

1.10.1.5 ASM Scalability and Performance Enhancements

This feature increases the maximum data file size that Oracle can support to 128 TB. ASM supports file sizes greater than 128 TB in any redundancy mode. This provides near unlimited capacity for future growth. The ASM file size limits are:

■ External redundancy - 140 PB

■ Normal redundancy - 42 PB

- High redundancy - 15 PB

Customers can also increase the allocation unit size for a disk group in powers of 2 up to 64 MB.

These improvements reduce database startup time and memory requirements, and allow support for larger ASM files, making it feasible to implement several hundred TB or even PB Oracle databases on ASM. Larger allocation units provide better sequential read performance.

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

1.10.1.6 Convert Single-Instance ASM to Clustered ASM

This feature provides support within Enterprise Manager to convert a non-clustered ASM database to a clustered ASM database by implicitly configuring ASM on all nodes. It also extends the single-instance to Oracle RAC conversion utility to support standby databases.

Simplifying the conversion makes it easier for customers to migrate their databases and achieve the benefits of scalability and high availability provided by Oracle RAC.

> **See Also:**
>
> Your platform-specific Oracle Real Application Clusters installation and configuration guide for details

1.10.1.7 New SYSASM Privilege for ASM Administration

This feature introduces the new SYSASM privilege to allow for separation of database management and storage management responsibilities.

The SYSASM privilege allows an administrator to manage the disk groups that can be shared by multiple databases. The SYSASM privilege provides a clear separation of duties from the SYSDBA privilege.

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

1.10.2 Change Assurance

The following sections describe new features for automatic capture and replay of workloads before and after changes to analyze impact.

1.10.2.1 Database Replay

Before making changes, such as hardware or software upgrades, companies typically conduct extensive testing to validate the changes. However, when the change is made on the production system, problems are often encountered because the testing was not performed on a realistic workload. Tools are available in the market to construct "synthetic workloads" involving multiple users. However, they are not able to adequately simulate the complexities of a real-world workload, such as interactions between concurrent activities and unpredictability of the workload profile.

The Database Replay feature addresses this need by enabling users to perform real-world testing by capturing the actual database workload on the production

system and replaying it on the test system. It also provides analysis and reporting to highlight potential problems (for example, errors encountered and divergence in performance) and recommend ways to remedy the problems.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.2.2 SQL Performance Analyzer

SQL performance regression is always a concern during system changes such as database upgrades, initialization parameter changes, and addition or dropping of indexes. The SQL Performance Analyzer feature alleviates this concern by providing an easy way to assess the impact of a change on the performance of SQL statements by comparing and contrasting their response times before and after the change. SQL Performance Analyzer allows you to capture the SQL workload from the source system, such as the production database, and to replay it on the test system where the change has been applied.

Any differences in response time of SQL statements, execution plan regressions, and so forth, are reported and precise recommendations on how to tune the under-performing SQL statements are provided.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.3 Database Control

The following sections describe Enterprise Manager solutions that help ease database management.

1.10.3.1 Advanced Replication Cutover

In previous releases, Advanced Replication was a trigger-based method of replication. With Oracle Database 11*g*, this feature is a cutover of existing functionality to Web-based Enterprise Manager Database Control.

This feature provides support for existing Advanced Replication customers by providing the ability to manage Advanced Replication from 11*g* Enterprise Manager Database Control.

1.10.3.2 ASM Enhancements

This feature provides several enhanced ASM features for the Enterprise Manager user interface.

Enterprise Manager has been enhanced for ASM File Access Control, OCR and Voting Disk in ASM, Disk Resync, ASM Rolling Migrations, Disk Zones, ASM Manageability and Infrastructure, ACL Enhancements, and Security Classes for Fusion Security.

1.10.3.3 Change Manager - Sync and Propagate

This feature helps users deploy an application to multiple sites and to upgrade an earlier version of an application to the current version at multiple sites.

Users can now propagate metadata definitions, and some data, from a database or baseline to multiple database targets.

1.10.3.4 Database Cloning Enhancements

This feature provides the following improvements in Enterprise Manager database cloning pages:

- Removes the requirement for source and destination stage area.

- Supports cloning from a generic RMAN backup set.

- Enhances the job summary user interface.

These improvements facilitate database cloning from Enterprise Manager using new RMAN features.

1.10.3.5 Database Configuration (ECM) Collection Enhancements

This new feature provides enhancements to Database Configuration (ECM collections) maintenance.

The benefit is an improved configuration collection framework.

1.10.3.6 Database Home Page and Performance Page Improvements

This feature provides new charts, changes the layout, includes more details, and makes other improvements to the Database Home page and Performance page.

Enhancements to the Database Home page and Performance page lead to a better utility and ease-of-use.

> **See Also:**
>
> *Oracle Database 2 Day + Performance Tuning Guide* for details

1.10.3.7 DBCA Enhancements

The following enhancements have been added to DBCA:

- Added support to configure new database options

  The following options in Oracle Database 11*g* can be configured using Database Configuration Assistant:

  - Oracle Application Express

  - Oracle Database Vault

  - Oracle Warehouse Builder

- Automatic memory management

  This is a new initialization parameter in Oracle Database 11*g* to automate the memory allocation. By default, Database Configuration Assistant now uses `MEMORY_TARGET` instead of specifying individual values for `SGA_TARGET` and `PGA_AGGREGATE_TARGET`. The memory management page of Database Configuration Assistant has new option to select automatic memory management.

- Oracle Data Mining

  In Oracle Database 11*g*, the data mining metadata is created when you run the `catproc.sql` script as the `SYS` user. You no longer configure this option through the Database Features screen of Database Configuration Assistant.

- `ORACLE_BASE` and Diagnostic Destination configuration

  The directory that you specify when you are prompted for `ORACLE_BASE` by Oracle Universal Installer is stored in the Oracle home inventory. Database Configuration Assistant uses this value to derive the default database locations

and the `DIAGNOSTIC_DEST` parameter. The diagnostic destination location contains all ADR directories (diagnostic files such as Alert logs and so on). Starting with Oracle Database 11*g* Release 1 (11.1), the initialization parameter settings for background dump, user dump, and core dump destinations are replaced by the Diagnostic Destination.

- Secure database configuration

  Oracle Database 11*g* has new defaults for audit and password profiles. Database Configuration Assistant has a new screen to enable the new security settings during the database creation and existing database configuration.

- Switching a database from Database Control to Grid Control configuration

  In previous releases, Database Configuration Assistant contained functionality to configure a database either with Database Control or with Grid Control. You could configure a database either while creating it or later. However, reconfiguring a database from Database Control to Grid Control required significant manual effort. With Oracle Database 11*g*, Database Configuration Assistant provides the Enterprise Manager Configuration plug-in, which automates the process to switch configuration of a database from Database Control to Grid Control.

1.10.3.8 DBUA Enhancements

The following enhancements have been added to DBUA:

- Express Edition upgrade

  For single-instance databases, Oracle Database Upgrade Assistant configuration utility enables you to upgrade from Oracle Database Express Edition (Oracle Database XE) to Oracle Database 11*g*. The XE database files reside under the path `ORACLE_BASE/oradata/XE`. These files must be copied to a new location as the user may remove the XE Home after upgrade.

- Moving datafiles into ASM, SAN, and other file systems

  You can move datafiles to ASM, OFS, or other storage devices, such as Storage Area Networks (SAN) and Network Area Storage (NAS), as part of the upgrade. If you move the database files during the upgrade, you can benefit from the typical downtime for this tablespace by rebalancing disks and moving files to a better storage device, such as SAN, NAS, or ASM.

- `ORACLE_BASE` and Diagnostic Destination configuration

  The directory that you specify when you are prompted for `ORACLE_BASE` by Oracle Universal Installer is stored in the Oracle home inventory. Database Upgrade Assistant uses this value to derive the default database locations and the `DIAGNOSTIC_DEST` parameter. The diagnostic destination location contains all ADR directories (diagnostic files such as Alert logs and so on). This diagnostic destination directory is required while upgrading an earlier Oracle Database release to Oracle Database 11*g* release of the database. If the `ORACLE_BASE` directory already exists, Oracle Database Upgrade Assistant automatically retrieves this information and populates its path. Starting with Oracle Database 11*g* Release 1 (11.1), the initialization parameter settings for background dump, user dump, and core dump destinations are replaced by the Diagnostic Destination.

- Command-line option to `AUTOEXTEND` system files

  The command line option `AUTOEXTEND` facilitates auto extending of the data files as a part of the upgrade. This option autoextends the data files during the upgrade and turns the autoextend back to its original settings after the upgrade. This

option is useful if there is enough room on the disk, and if you do not need to add new datafiles or manually increase the size of the files.

These features allow XE customers to upgrade to Standard Edition or Express Edition database for support and scalability.

These features also provide the ability to move to better storage devices as part of upgrade. You can move datafiles to be in compliance with OFA standards.

1.10.3.9 Enhanced Adaptive Metric Thresholds

The Adaptive Metric Thresholds feature simplifies and improves the selection of alert thresholds for database performance metrics. Key enhancements include:

- Full integration with AWR baselines as the source for metric statistics.
- Quick configuration option offers one-click starter set of thresholds based on OLTP or Data Warehouse system profiles.
- Automated time group selection for SYSTEM moving window baseline.
- Guided discovery of candidate alert metrics based on response levels to known problem.
- Improved metric charting and analysis.

This feature helps DBAs configure high quality, baseline-driven performance alert thresholds with minimal effort.

1.10.3.10 Fusion Requirements

This feature fulfills the requirements for Fusion in Enterprise Manager, including:

- Group copy of tnsnames.ora
- Monitor and validate init.ora
- Monitor database session trace options
- View database trace files
- Archive and purge trace and alert files

This feature allows Oracle Application Server to use Enterprise Manager for many monitoring and configuration-related tasks.

1.10.3.11 Management of Oracle Text Indexes

This feature provides a user interface to support management of Oracle Text indexes.

Oracle Enterprise Manager provides an interface for configuration, maintenance, and administration of Oracle Text, including:

- Viewing details of Oracle Text indexes
- Synchronizing Oracle Text indexes
- Optimizing Oracle Text indexes
- Rebuilding indexes
- Editing Oracle Text indexes
- Querying log analysis

> **See Also:**
>
> *Oracle Text Application Developer's Guide* for details

1.10.3.12 Migrate Database to ASM - Enhancements in Enterprise Manager

The Enterprise Manager pages for Migrate Database to ASM have been enhanced to include notification when a job is suspended and support for tablespace level migration.

This feature makes it easier to use Enterprise Manager to migrate non-ASM databases to ASM.

1.10.3.13 Storage / Audit Report and Metric Enhancement

Enterprise Manager has been enhanced to provide better storage and audit reports and metrics.

This feature facilitates monitoring in the storage area and audit area.

1.10.3.14 Storage: Schema, Security, and Configuration Enhancements

Enterprise Manager has been enhanced in the following areas: export, import, Data Pump, storage administration, schema administration, OLAP, summary management, and partition maintenance.

This feature improves Enterprise Manager support in several areas.

1.10.3.15 Wait Activity Detail Enhancement

Until now, Oracle Enterprise Manager displayed the Wait Activity Drilldown detail in a chart format.

This feature provides more detailed Wait Activity information for the client, service, module, and action pages in Oracle Enterprise Manager.

> **See Also:**
>
> *Oracle Database 2 Day + Performance Tuning Guide* for details

1.10.3.16 Workspace Manager

The Oracle Enterprise Manager interface for the Workspace Manager features has been enhanced to include support for compress workspace, manage system parameters, privilege management, `FindRICSet`, and others.

This feature facilitates the administration of Workspace Manager from Oracle Enterprise Manager.

1.10.4 Holistic Database Management

The following sections describe database-wide, holistic, consolidation management and control enhancements.

1.10.4.1 ADDM for Oracle Real Application Clusters

ADDM has been enhanced to provide comprehensive cluster-wide performance diagnostic and tuning advice. A special mode of ADDM analyzes an Oracle RAC database and reports on issues that are affecting the entire cluster as well as those that are affecting individual instances.

This feature is particularly helpful in tuning global resources such as I/O and interconnect traffic and makes the tuning of Oracle RAC databases easier and more precise.

See Also:

Oracle Database 2 Day + Performance Tuning Guide for details

1.10.5 Intelligent Infrastructure

This section describes Statistics, Alerts, Advisor and Automated Maintenance Tasks infrastructure that enable self-management.

1.10.5.1 Auto-Task

In Oracle Database 11*g*, Auto-Task can run hundreds or thousands of jobs in the manageability windows. This feature adds the necessary scheduler infrastructure to automatically manage this very high manageability job load.

This feature provides Auto-Task the necessary infrastructure to handle large numbers of jobs.

See Also:

Oracle Database Administrator's Guide for details

1.10.5.2 Automatic Maintenance Tasks Management

This feature provides out-of-the-box management of resource distribution (CPU and I/O) among the various database maintenance tasks such as Automatic Optimizer Statistics Collection, Automatic Segment Advisor, and others. The CPU is automatically managed. I/O is managed only if the I/O Resource Manager is enabled.

This feature ensures that work during maintenance operations is not affected and that user activity gets the necessary resources to complete.

See Also:

Oracle Database Administrator's Guide for details

1.10.5.3 Automatic Workload Repository (AWR) Baselines

Automatic Workload Repository (AWR) baselines enable accurate performance comparison by providing the ability to tag a certain period of time as a period of interest against which comparisons can be made at some time in the future. All the related performance statistics for this period are maintained and are available for use for any performance analysis with another time period.

AWR baselines provide powerful capabilities for defining dynamic and future baselines and considerably simplify the process of creating and managing performance data for comparison purposes.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.5.4 Database Statistics Model Consolidation

In this release, the `V$SYSSTAT` and `V$SYS_TIME_MODEL` infrastructures have been consolidated into one advanced component.

This feature also implements a model to break out wait and general statistics into divisions that are usable for automated and manual tuning of the system.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.5.5 Enhanced Active Session History

The Active Session History infrastructure has been enhanced to allow for improved database performance diagnosis and monitoring.

This feature includes enhancements to row source information for query progress monitoring and time model statistics.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.5.6 Enhanced Advisor Framework

In this release, the Advisor Framework has been enhanced to support Health Monitor features.

The Advisor Framework also supports AWR import and export features.

1.10.5.7 Lightweight Jobs

Lightweight jobs have lower creation overhead and much less redo. You can now create large numbers of short, frequently executing jobs and still maintain performance.

See Also:

Oracle Database Administrator's Guide for details

1.10.5.8 Scheduling External Jobs on Multiple Nodes

With this new feature, you can create external jobs (for example, shell scripts) and provide a list of target machines where the jobs should be executed. For example, you can schedule a shell script `x.sh` to execute on machines A, B, and C. The target machines need not necessarily have an Oracle database installed. The jobs are managed from a central location.

This feature enables you to schedule an external job on multiple nodes while managing it from a central location.

See Also:

Oracle Database Administrator's Guide for details

1.10.5.9 Simplified Initialization Parameter Management

Several enhancements in this release make the management of server parameter file and initialization parameter values easier. Some of these enhancements are:

- More fault tolerant `SPFILE`
- Easier recovery from loss of `SPFILE`
- Preventing users from setting invalid parameter values in the `SPFILE`
- Redesigned, more intuitive Enterprise Manager initialization parameter management interface

This feature improves database manageability by making it easier to manage the server parameter file and changes to initialization parameter values.

See Also:

Oracle Database Administrator's Guide for details

1.10.6 Resource Manageability

The following sections describe enhancements to Resource Manageability.

1.10.6.1 I/O Calibration

The DBA can assess the I/O capability of the database's storage system by using the PL/SQL function DBMS_RESOURCE_MANAGER.CALIBRATE_IO(). This routine issues a very I/O intensive read-only workload to the database's files to assess the maximum IOPS (I/O requests per second) and MBPS (megabytes of I/O per second) that can be sustained. This data can be reexamined at any time using the DBA table, DBA_RSRC_IO_CALIBRATE, that stores I/O calibration results.

The calibration should be performed when the database is idle and during off-peak hours to minimize the heavy I/O workload from interfering with the regular workload and vice versa.

I/O calibration can be used to understand the performance of the storage subsystem and figure out whether I/O performance problems stem from the database or the storage subsystem. Unlike various external I/O calibration tools, this tool uses the Oracle code stack and issues I/O randomly rather than sequentially. The results, therefore, much more closely match the actual database performance.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

1.10.6.2 I/O Statistics

I/O Statistics provide consistent and complete statistics from all database clients by:

- I/O type (single and multi block, read and write)
- Component
- Data file
- Consumer group

The I/O statistics collected are used in AWR for monitoring, diagnostics, and tuning. The information is also used by the I/O Resource Manager.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.6.3 Per Session I/O Limits

In previous versions of Oracle, the DBA could specify the maximum amount of time a session could run before some action was taken; for example, the call was aborted, the session was killed, or the session was migrated to a new consumer group. This feature is configured on a per consumer group basis in a resource plan.

In Oracle Database 11*g*, DBAs can also specify the maximum number of I/O requests or the maximum megabytes of I/O that a session can issue before the same set of actions are taken.

This feature is used for two purposes. The first is to automatically identify runaway queries. The second is to move sessions executing long-running calls to lower-priority consumer groups.

See Also:

Oracle Database Administrator's Guide for details

1.10.6.4 Resource Manager Statistics in AWR

AWR contains new historical views for Resource Manager statistics. These new views, `DBA_HIST_RSRC_PLAN` and `DBA_HIST_RSRC_CONSUMER_GROUP`, contain the historical version of the statistics in the views V$RESOURCE_PLAN and `V$RESOURCE_CONSUMER_GROUP`. AWR also contains per-minute metrics for resource consumption and Resource Manager-induced waits in the view `V$RSRCMGRMETRIC`.

This feature makes it easier to view historical Resource Manager statistics.

See Also:

Oracle Database Administrator's Guide for details

1.10.6.5 Resource Plans

This feature provides sophisticated new built-in resource plans:

- Mixed-workload plan

  Provides resource management for a mixed environment consisting of OLTP and DSS/batch jobs.

- Data Warehouse plan

  Provides resource management for a data warehousing environment.

- Maintenance Plan

  Provides resource management for the maintenance window.

The resource plans are shipped with the database and provide resource management directives that should bring immediate benefits in most customers situations.

See Also:

Oracle Database Administrator's Guide for details

1.10.7 Simplify Configuration

The following sections describe how to streamline management of Oracle Database.

1.10.7.1 Enhanced Optimal Flexible Architecture (OFA) Support

The database installation process has been redesigned to be based on the ORACLE_ BASE environment variable. Until now, setting this variable has been optional and the only required variable has been ORACLE_HOME. With this feature, ORACLE_BASE is the only required input, and the ORACLE_HOME setting will be derived from ORACLE_ BASE. This change streamlines how Oracle software installation is organized, thereby making ongoing management easier.

This feature improves manageability by making default Oracle Database installations more compliant with Optimal Flexible Architecture (OFA) specifications.

1.10.8 Space, Object and Transaction Manageability

Self-managing database space, object and transaction.

1.10.8.1 AUM Default Setting and Migration Support

Automatic Undo Management (AUM) is now enabled by default. A PL/SQL procedure is provided to help properly size the Undo tablespace for each individual environment.

This feature facilitates seamless migration to AUM from databases being upgraded to the new release so that they can start taking advantage of all the benefits of AUM.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details

1.10.8.2 Enhanced Statistics Collection for Partitioned Objects

An improved statistics collection process for partitioned objects avoids having to regather statistics on partitions that have not been touched by using a summary instead.

Partitioned objects tend to become larger and larger, and statistics collection, and particularly global statistics collection, can become increasingly time and resource intensive. This feature significantly improves the speed and accuracy of statistics collection for partitioned objects.

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

1.10.8.3 Simplified Temp Space Management

This feature demystifies the management of temp space. It allows a DBA to easily determine how much temp space is being used, who is using it, and whether more is needed.

This feature helps reduce errors due to inadequate temp space configuration and identifies application design issues such as runaway queries.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for details

1.10.9 SQL Manageability

The following sections describe self-managing query performance and access methods.

1.10.9.1 Automatic SQL Tuning with Self-Learning Capabilities

This feature takes automatic SQL tuning to the next level by adding self-learning capabilities to it.

Oracle Database now automatically detects high-load SQL statements and then tunes them automatically as needed in the maintenance window by creating appropriate SQL Profiles. It also issues proactive advice to create new access structures such as indexes that will significantly improve the performance of the high-load SQL statements.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.9.2 Enhanced Optimizer Statistics Maintenance

The publication of statistics is now decoupled from the collection of statistics.

Changing optimizer statistics can have an adverse affect on execution plans. By separating the statistics collection from making them fully available to the optimizer, application developers and DBAs have the opportunity to assess the effects of the new statistics on execution plans.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.9.3 Multi-Column Statistics

Multi-column statistics can be now collected for any arbitrary set of columns in a single table. This enhancement improves the costing accuracy of many SQL constructs, such as multi-column filter conditions and joins involving more than one column.

Extending the type of optimizer statistics gathered helps the optimizer choose the best plan with improved accuracy.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.9.4 Partition Advisor

The SQL Access Advisor has been enhanced to include advice on how to partition tables, materialized views, and indexes in order to improve performance of SQL statements.

Oracle Database offers a wide variety of partitioning options whose proper use requires expertise and time. This feature makes it easy for all users to use partitioning in a way that is most suitable for their environments by giving intelligent and accurate advice on exactly how to partition a particular object for optimal performance.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.9.5 SQL Plan Management

The SQL plan management feature enables the optimizer to maintain a history of execution plans for a SQL statement. Using the execution plan history, the optimizer is able to detect a new plan representing a plan change for a SQL statement. When the optimizer detects a new plan, it stores the new plan and marks it for performance evaluation and uses the old (currently known good) plan. The optimizer uses the new plan only after its performance is verified to be better than that of the old plan.

Execution plan change for a SQL statement can cause severe performance degradation on a system. Execution plan changes occur due to system changes such as a new optimizer version, refresh of optimizer statistics, and optimizer parameter changes. This new SQL plan management feature prevents performance regressions caused by execution plan changes.

See Also:

Oracle Database Performance Tuning Guide and *Oracle Database PL/SQL Packages and Types Reference* for details

1.10.10 Streams Manageability

The following sections describe enhancements to self-managing Streams (replication and AQ) operations.

1.10.10.1 Automated Alerts for Streams

Alerts are generated automatically if a Streams process becomes disabled. Alerts can also be generated based on user-defined thresholds for key Streams metrics.

Alerts provide the Streams administrator with immediate information on critical Streams components.

See Also:

Oracle Database 2 Day + Data Replication and Integration Guide for details

1.10.10.2 Divergence Detection and Resolution

The `DBMS_COMPARISON` package enables comparison of table data between databases and provides the ability to resolve any data discrepancies.

This feature can be used to validate the consistency of data within a single database or across databases.

See Also:

Oracle Streams Replication Administrator's Guide for details

1.10.10.3 Split and Merge of Streams

Administrators can split a stream in cases where a replica is unavailable and can allow that stream to replicate at a future point in time. After the replica using the split stream catches up, the stream can be merged into a single stream again.

Use of this feature minimizes the impact of unavailable replicas in a Streams replication.

See Also:

Oracle Streams Replication Administrator's Guide for details

1.10.10.4 Streams Message Tracking

This feature provides a simple way for an administrator to track the path an LCR traverses between processes and across databases within Streams. Streams message tracking is initiated by way of a procedure for a particular session. Subsequent LCR activity generated in that session is marked. As the LCRs move through the stream, the actions on those LCRs are traced, and the information is visible in runtime views.

Streams message tracking allows administrators to easily diagnose problems in a Streams configuration.

See Also:

Oracle Streams Replication Administrator's Guide for details

1.10.10.5 Streams Performance Advisor

This feature provides an advisor to diagnose configuration and performance issues while using Streams. The Streams performance advisor reports on performance of all Streams components and provide suggestions for performance tuning.

The Streams performance advisor provides valuable information on setting up and managing a Streams configuration.

> **See Also:**
>
> *Oracle Streams Concepts and Administration* for details

1.10.10.6 Streams Topology

This feature provides a PL/SQL package and set of data dictionary views for discovery of the Streams topology. Using a single schema with database links and privileges to participating databases, the topology discoverer traverses the various components of Streams (capture, propagation sender/receiver, and apply) in each database to identify correlated components and assign a Streams identifier. Database views provide the administrator with details of the Streams topology.

Administrators can check the Streams set up quickly using the Streams topology views. In addition, the Streams advisors are used in combination with this feature to manage and monitor Streams.

> **See Also:**
>
> *Oracle Streams Concepts and Administration* for details

1.10.10.7 Synchronous Streams Capture

This feature makes it possible to capture changes to database tables as part of a user transaction.

Synchronous Streams Capture minimizes the resource requirements to replicate a small subset of database tables.

> **See Also:**
>
> *Oracle Streams Concepts and Administration* for details

1.11 Performance

The following sections describe performance features for Oracle Database 11*g* Release 1 (11.1).

1.11.1 General Server Performance

The following sections describe general server performance enhancements.

1.11.1.1 Direct NFS Client

This features implements NFS client as part of Oracle Database kernel.

The primary benefits of this feature are improved performance and manageability. The Direct NFS Client improves I/O performance by incorporating Oracle-specific optimizations and eliminating the additional overhead traditionally found in operating system kernel mode implementation of NFS protocol. In addition, it simplifies configuration by eliminating the need to manually tune most of the NFS client parameters.

1.11.1.2 Enhanced Partition Pruning Capabilities

Partition pruning now uses a new filtering mechanism called bloom filtering instead of subquery pruning. While subquery pruning was activated on a cost-based decision and consumed internal (recursive) resources, pruning based on the new scheme is activated all the time since it consumes minimal additional resources.

The performance of partition pruning has been enhanced. Furthermore, partition pruning will be automatically activated for every join with a partitioned object.

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

1.11.1.3 Intelligent Cursor Sharing

This feature helps determine whether a cursor will be shared. For example, instead of blindly reusing an existing cursor with its plan, the optimizer takes ranges of safe values into account when evaluating cursor sharability.

Cursor sharing has been enhanced to a more sophisticated sharing model. This enables you to leverage cursor sharing more commonly while preserving bind variable specific plan optimizations for shared statements.

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for details

1.11.1.4 NLS Performance Improvements

Oracle Database 11*g* includes incremental performance enhancements for NLS-related features.

This feature provides better performance for customers using NLS features.

> **See Also:**
>
> *Oracle Database Globalization Support Guide* for details

1.11.2 Improve Basic SQL and PL/SQL

The following sections describe improved performance of basic SQL and PL/SQL statements.

1.11.2.1 Binding and Defining Multiple Buffers in OCI

This feature allows OCI applications to specify noncontiguous buffers for reading or writing in the define or bind calls.

Applications with disperse data sources with frequent read/writes often have data in noncontiguous memory. With this feature, these applications perform faster and use less memory because the overhead of memory copying to a contiguous chunk is eliminated.

> **See Also:**
>
> *Oracle Call Interface Programmer's Guide* for details

1.11.2.2 Bitmap Join Index Support for Index-Organized Tables

A bitmap join index on index-organized tables (IOTs) provides access to rows of one IOT or heap table based on columns of another IOT or heap table. Bitmap join index support for IOTs can avoid actual joins of tables and reduce the volume of data to be

joined by performing restrictions in advance. Bitmap join indexes are space efficient and can speed up queries through bitwise operations.

This feature provides bitmap join index support for index-organized tables (IOTs) to improve performance.

> **See Also:**
>
> *Oracle Database Data Warehousing Guide* for details

1.11.2.3 Cost-Based Optimizer Enhancements

The cost bases of various SQL execution steps and internal transformations are now determined in a more accurate manner.

Enhancements to the optimizer's internal costing mechanisms improve the accuracy of execution plan generation, thus generating more optimal plans.

1.11.2.4 Network/OCI Fusion

This feature increases the throughput and decreases the overhead of data transfer through OCI/Network layers.

This enhancement improves the performance of SQL statements as well as statements involving large amounts of data transfer.

1.11.2.5 OCI Descriptor Array Allocation

OCI descriptors are opaque data structures meant only for interpretation by OCI but used by applications to signify data such as `OCIDateTime` and `OCILobLocator`.

Until now, descriptors could only be allocated one at a time. This feature allows allocating and freeing an array of descriptors simultaneously.

Applications using many OCI resources can allocate these resources more easily with the new Descriptor Array Allocation interface, providing less code maintenance and less CPU overhead for allocating and freeing memory.

> **See Also:**
>
> *Oracle Call Interface Programmer's Guide* for details

1.11.2.6 OCI Implicit ROWID Fetching

This feature provides the ability to implicitly fetch the `ROWID` of the rows fetched as the result of a `SELECT...FOR UPDATE` query into memory defined at position 0 while doing array fetches. Users need not alter the query to fetch the `ROWID` along with other columns explicitly.

Applications taking advantage of array fetches can now update individual rows as needed, giving more flexibility to high performance applications.

> **See Also:**
>
> *Oracle Call Interface Programmer's Guide* for details

1.11.2.7 PL/SQL Function Result Cache

New in Oracle Database 11*g* is the ability to mark a PL/SQL function to indicate that its result should be cached to allow lookup, rather than recalculation, on the next access when the same parameter values are called. This function result cache saves significant space and time. Oracle does this transparently using the input value as the lookup key. The cache is system-wide so that all distinct sessions invoking the

function benefit. If the result for a given set of values changes, you can use constructs to invalidate the cache entry so that it is properly recalculated on the next access. This feature is especially useful when the function returns a value that is calculated from data selected from schema-level tables. For such uses, the invalidation constructs are simple and declarative.

Concurrent, multi-user applications that use this feature experience better response times. Applications that implement a session-private scheme consume significantly less memory by using this feature and, therefore, experience improved scalability.

See Also:

Oracle Database PL/SQL Language Reference for details

1.11.2.8 PL/SQL Native Compilation Without Needing a Third-Party C Compiler

On many platforms, the PL/SQL compiler may now generate processor-specific native code directly from the PL/SQL source code without needing to use a third-party C compiler. As in Oracle Database 10*g*, the generated code will be stored canonically in the database catalog. When a unit is needed, the Oracle executable loads the code directly into memory from the catalog, without first staging the unit through a `.DLL` or `.so` file.

The execution speed of natively compiled PL/SQL programs will never be slower than in Oracle Database 10*g* and may be improved in some cases by as much as an order of magnitude.

On platforms with this feature, Oracle Database 10*g* parameters, `plsql_native_library_dir` and `plsql_native_library_subdir_count`, will no longer be needed, and need not be set. If upgrading from Oracle Database 10*g*, any existing files in those directories may safely be deleted.

The benefit of PL/SQL native compilation is automatically available with Oracle Database 11*g*. No third-party software (neither a C compiler nor a DLL loader) is needed.

Note that this feature may not be available on all platforms in the first release of Oracle 11*g*. If this feature is not available on a particular platform, Oracle Database 10*g* implementation of native compilation will be unchanged on that platform. Check your platform-specific documentation to verify that this feature is available on your platform.

See Also:

Oracle Database PL/SQL Language Reference for details

1.11.2.9 Query Result Cache

A separate shared memory pool is now used for storing and retrieving cached results. Query retrieval from the query result cache is faster than rerunning the query. Frequently executed queries will see performance improvements when using the query result cache.

The new query result cache enables explicit caching of results in database memory. Subsequent queries using the cached results will experience significant performance improvements.

See Also:

Oracle Database Performance Tuning Guide for details

1.11.3 Optimize Key Platforms

The following sections describe platform-specific optimizations on platforms such as Linux, Windows, and Intel.

1.11.3.1 ODBC Performance Improvement

The Oracle ODBC Driver has been enhanced with the following options that can be set using Setup DDL on Windows or through `odbc.ini` on Linux and Solaris:

- Statement caching

- Removal of optimizer hints for catalog functions

- Support for column binding for `NUMBER` columns as `FLOAT`

ODBC applications will perform faster, with little or no code change.

1.12 Security

Features described in the following sections include security and compliance.

1.12.1 Oracle Advanced Security

The following sections describe Oracle advanced security features.

1.12.1.1 Kerberos Cross Realm Support

This feature updates the Oracle Kerberos implementation to support cross realm authentication. Now a Kerberos principal in one realm can provide authentication to a principal in another realm.

The benefit of this feature is that it makes Kerberos a more viable alternative for customers seeking strong authentication without the overhead of managing certificates and public key infrastructure (PKI).

1.12.1.2 SYSDBA Strong Authentication

All connections to the database can now use strong authentication, including those made as `SYSDBA` and `SYSOPER`.

Oracle has supported strong authentication through PKI, Kerberos, and Radius since Oracle8*i* for all connections to the database except those connections made as `SYSDBA` or `SYSOPER`. This feature completes Oracle support for strong authentication by extending the strong authentication feature to connections as `SYSDBA` and `SYSOPER`.

> **See Also:**
>
> *Oracle Database Security Guide* for details

1.12.1.3 Tablespace Encryption

Tablespace encryption is an enhancement to the Oracle Advanced Security Transparent Data Encryption solution. Using tablespace encryption, customers can encrypt an entire tablespace, encrypting all data within the tablespace. When the database accesses the tablespace, the relevant data blocks are transparently decrypted for the application.

Transparent Data Encryption tablespace encryption provides an alternative to Transparent Data Encryption column encryption by enabling encryption of an entire tablespace. This eliminates the need for granular analysis of applications to determine

which columns to encrypt, especially for applications with a large number of columns containing personally identifiable information (PII). Using tablespace encryption, customers can encrypt entire tables, eliminating the need to identify which columns contain personally identifiable information (PII). Customers who have small amounts of data to encrypt can continue to use the Transparent Data Encryption column encryption solution.

> **See Also:**
>
> *Oracle Database Advanced Security Administrator's Guide* for details

1.12.1.4 Hardware-Based Master Key Protection

This feature allows the Transparent Data Encryption (TDE) master key to be stored in an external Hardware Security Module (HSM) for even stronger security.

This feature provides even stronger security for the Transparent Data Encryption (TDE) master key for customers who are concerned about storing the master key on the operating system.

> **See Also:**
>
> *Oracle Database Advanced Security Administrator's Guide* for details

1.12.2 Secure by Default

The following sections describe out-of-the-box secure configuration with increased password protections, secure file permissions, optional default audit settings, and new controls on network callouts from the database.

1.12.2.1 Audit By Default

This feature automatically configures the database for auditing and turns on auditing for specific events such as database connections.

Auditing is an important part of the security layer. Oracle has turned on some auditing settings by default to help customers better track connections to the database.

> **See Also:**
>
> *Oracle Database Security Guide* for details

1.12.2.2 Built-in Password Complexity Checker

This feature builds into the database the password complexity routine that is documented in the *Oracle Database Security Guide*.

This built-in functionality can be easily turned on to guarantee that complex passwords are used when setting or resetting a password.

> **See Also:**
>
> *Oracle Database Security Guide* for details

1.12.2.3 Built-in User Profile

This feature makes it easier to associate a database user with a profile by providing a build-in profile.

Customers can use a built-in profile to enforce password expiration and reuse policies.

See Also:

Oracle Database Security Guide for details

1.12.2.4 Fine-Grained Access Control on Network Call-outs from the Database

The packages UTL_TCP, UTL_INADDR, UTL_HTTP, UTL_SMTP, and UTL_MAIL allow Oracle users to make network callouts from the database using raw TCP or using higher level protocols built on raw TCP. Until now, when the granularity of a privilege is simply execute on each package, there has been no way to allow a user access to just a list of specified internet hosts. The new package DBMS_NETWORK_ACL_ADMIN allows fine-grained control using ACLs implemented by XML DB.

This feature enables the DBA to carefully control which internet hosts Oracle users can access using the supplied PL/SQL packages.

See Also:

Oracle Database Security Guide for details

1.12.2.5 Improve Security for Network Administration, Registration, and Operation

This feature provides secure service registration with the listener. Listener administration is secure by default. The new model provides an easy, optional, and manageable process administration and registration.

This security improvement prevents unauthorized use of remote and local operations.

See Also:

Oracle Database Net Services Administrator's Guide for details

1.12.2.6 Parameters for Enhanced Security of Database Communication

This feature provides the following enhancements:

- Reporting bad packets received on the database from protocol errors.

- Terminating or resuming server execution on receiving bad packets.

- Configuring the maximum number of authentication attempts.

- Controlling the display of the database version banner.

- Configuring banners for unauthorized access and auditing user actions.

This feature helps to keep data safe and secure.

See Also:

Oracle Call Interface Programmer's Guide, *Oracle Database Security Guide*, and *Oracle Database Net Services Reference* for details

1.12.2.7 Support Nonanonymous LDAP Access for Net Naming

This feature provides support for nonanonymous LDAP lookup for Network naming. For more secure LDAP access, users can specify that clients must identify themselves before a name lookup.

This feature results in improved security between LDAP and Net naming.

See Also:

Oracle Database Net Services Administrator's Guide for details

1.12.3 Security Manageability

The following sections describe replacement and integration of Enterprise Security Manager, Oracle Policy Manager, Selective Audit, and Audit Vault UI into a Java/Enterprise Manager SDK.

1.12.3.1 Integrated Database Security Manageability

Functionality previously provided in Enterprise Security Manager and Oracle Policy Manager has been integrated into Enterprise Manager. In addition, manageability for new features such as Transparent Data Encryption has been added to Enterprise Manager.

Over the years, Oracle has had numerous tools for managing database security. Now all of the functionality in those tools has been integrated into Enterprise Manager, dramatically simplifying security management in the database.

1.12.3.2 Virtual Private Catalog

This enhancement allows the catalog administrator to grant visibility of certain registered databases in the catalog to certain RMAN users.

This feature improves catalog security by allowing an RMAN user to view only his or her authorized, registered databases within a catalog. An RMAN user is no longer able to see all registered databases.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.12.4 Stronger Password Protection

The following sections describe authentication features for Oracle Database 11*g* Release 1 (11.1).

1.12.4.1 Standards-Based Password Algorithm

This feature modifies the verifier or hash used to store Oracle passwords.

This feature provides stronger protection for stored database passwords based on industry standard algorithms and provides enhanced security for password-based authentication by enabling usage of mixed case in passwords.

See Also:

Oracle Database Security Guide for details

1.13 Windows

The following sections describe Windows features for Oracle Database 11*g* Release 1 (11.1).

1.13.1 Integration with Windows Services

These features provide new and enhanced support for Oracle databases running Windows. They integrate natively with Windows services, including improved integration with Active Directory and Volume Shadow Copy Service support.

1.13.1.1 Active Directory Security Enhancements

This feature enhances security through better integration with the default Active Directory configuration, adding support for OS authentication with Active Directory.

This feature improves Net Naming support for Microsoft Active Directory.

1.13.1.2 Volume Shadow Copy Service (VSS) Writer

The Oracle VSS writer allows Oracle Database to participate in VSS-initiated backup and recovery on Windows 2003 and above. VSS is a Windows infrastructure composed of three participants:

- Requestors - management applications that initiate backup and recovery

- Writers - applications that own the data

- Providers - storage/hardware applications that coordinate the reading and writing of the data to appropriate storage

Several leading backup and storage vendors currently provide requestor and provider applications. With the Oracle VSS writer, customers can reliably back up and recover Oracle data in a VSS environment, which interoperates with these vendors' components.

This feature provides full interoperability with Volume Shadow Copy Service (VSS) infrastructure on Windows 2003 and above.

See Also:

Oracle Database Platform Guide for Microsoft Windows for details

2

Oracle Database 11*g* Documentation

This chapter lists the books available online that document Oracle® Database 11*g*.

2.1 Documentation Titles and Online Documentation Structure

All books are available in electronic format.

The following categories of documentation are available:

- Oracle® Database 11g Windows Documentation

  Subcategories include:

  - Developer's Guides

- Oracle® Database 11g Server Library

  Subcategories include:

  - Administration, Concepts, and Reference

  - Application Development

  - Application Express

  - Application Reference

  - Backup and Recovery

  - Data Warehousing

  - Language and Interface

  - Networking and Security

  - Oracle Real Application Clusters

  - Oracle Text

  - Others

  - Streams and Replication

- Oracle® Database 11g OLAP Documentation

- Oracle® Database 11g Data Mining Documentation

- Oracle® Database 11g Spatial Documentation

- Oracle® Database 11g Java Documentation

- Oracle® Database 11g Linux x86 Documentation

- Oracle® Database 11g Release Notes Documentation

- Oracle® Database 11g Installation Guide Documentation
- Oracle® Database 11g Documentation Library

2.1.1 Oracle® Database 11g Windows Documentation

Table 2–1   Oracle® Database Windows Documentation

Documentation	Part Number	Description
Developer's Guides		
Oracle Data Provider for .NET Developer's Guide	B28375-01	Describes provider-specific features and properties of ODP.NET. Oracle Data Provider for .NET (ODP.NET) is an implementation of Microsoft's ADO.NET interface and provides access to Oracle databases.
Oracle Database Extensions for .NET Developer's Guide for Microsoft Windows	B28376-01	Provides introductory, installation, post-installation configuration, and usage information for Oracle Data Provider for .NET.
Oracle Services for Microsoft Transaction Server Developer's Guide for Microsoft Windows	B28377-01	Provides introductory, installation, configuration, usage, and administration information for using Oracle Services for Microsoft Transaction Server. Microsoft Transaction Server is a proprietary Component Object Model transaction processing system that runs on an Internet or network server.
Oracle Objects for OLE Developer's Guide for Microsoft Windows	B28378-01	Describes the concepts important to OO4O and the object classes. Oracle Objects for OLE (OO4O) allows easy access to data stored in Oracle databases with any programming or scripting language that supports the Microsoft COM Automation and ActiveX technology.
Oracle Provider for OLE DB Developer's Guide for Microsoft Windows	B28431-01	Describes provider-specific features and properties of OraOLEDB. Oracle Provider for OLE DB (OraOLEDB) is based on an open standard that uses a set of Component Object Model interfaces for data access, in this case to Oracle databases.
Oracle COM Automation Feature Developer's Guide for Microsoft Windows	B31223-01	Describes how Oracle COM Automation Feature provides a mechanism to manipulate COM objects through PL/SQL. This developer's guide provides introductory, installation, post-installation configuration, AND usage information FOR Oracle COM Automation Feature.

2.1.2 Oracle® Database 11g Server Library

Table 2–2   Oracle® Database Server Library

Documentation	Part Number	Description
Administration, Concepts, and Reference		
Oracle Database Performance Tuning Guide	B28274-01	Describes detailed ways to enhance Oracle performance by writing and tuning SQL properly, using performance tools, and optimizing instance performance. It also explains how to create an initial database for good performance and includes performance-related reference information. This book could be useful for database administrators, application designers, and programmers.
Oracle Database 2 Day + Performance Tuning Guide	B28275-01	A quick start guide that teaches you how to perform day-to-day database performance tuning tasks using features provided by the Oracle Diagnostics Pack, Oracle Tuning Pack, and Oracle Enterprise Manager.
Oracle Database Heterogeneous Connectivity User's Guide	B28277-02	Describes the Oracle approach to information integration in a heterogeneous environment. Describes Oracle Database Gateways and is an administrator's guide for these Oracle products.
Oracle Database Error Messages	B28278-01	Includes all database error messages.
Oracle Database New Features Guide	B28279-01	Oracle Database New Features Guide is addressed to people familiar with previous versions of Oracle Database who would like to become familiar with features, options, and enhancements that are new in this release of the database.
Oracle Database High Availability Overview	B28281-01	Describes Oracle's approach for a highly available database environment. Provides an overview of high availability and helps you to determine your high availability requirements. It also describes Oracle Database products and features that are designed to support high availability and describes the primary database architectures that can help your business achieve high availability.
Oracle Database SQL Language Quick Reference	B28285-01	Contains a high-level text description of the SQL syntax used to manage information in Oracle Database.
Oracle Database SQL Language Reference	B28286-01	Contains a complete description of Oracle SQL, the language used to manage information in Oracle Database.
Oracle Database Globalization Support Guide	B28298-01	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.

Table 2–2   (Cont.) Oracle® Database Server Library

Documentation	Part Number	Description
Oracle Globalization Development Kit Java API Reference	B28299-01	Describes APIs for the Oracle Globalization Development Kit.
Oracle Database Upgrade Guide	B28300-01	Explains the process of planning and executing database upgrades on Oracle Database server. In addition, this guide provides information about compatibility, about upgrading applications to the current release of Oracle, and about important changes in the current release, such as initialization parameter and data dictionary changes.
Oracle Database 2 Day DBA	B28301-01	Provides a single source reference for administering Oracle Database in a small to midsize environment. 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 Administrator's Guide	B28310-01	Explains how to create and to manage Oracle Database. Includes information about distributed processing.
Oracle Database Concepts	B28318-01	Describes how Oracle Database functions and lays a conceptual foundation for more detailed information contained in other Oracle Database manuals.
Oracle Database Utilities	B28319-01	Describes how to use Oracle Database utilities to load data into a database, transfer data between databases, and maintain data. The utilities described include the new Data Pump export and import products, the original export and import products, SQL*Loader, external tables, the Metadata API, LogMiner, DBVERIFY, and DBNEWID.
Oracle Database Reference	B28320-01	Provides reference information about database initialization parameters, static data dictionary views, dynamic performance views, database limits, and SQL scripts that are part of Oracle Database system.
Oracle Database Sample Schemas	B28328-01	Describes the sample schemas included in the seed database that ships with Oracle Database 11*g*. These schemas are used in examples in Oracle documentation and curriculum materials and in demos on Oracle Technology Network.
Oracle Database Enterprise User Security Administrator's Guide	B28528-01	Describes how to implement, configure, and administer Oracle Database users in Oracle Internet Directory, the directory service provided by the Oracle Identity Management platform.
*SQL*Plus User's Guide and Reference*	B31189-01	Introduces SQL*Plus and its uses, and provides a description of each SQL*Plus command.

Table 2–2   (Cont.) Oracle® Database Server Library

Documentation	Part Number	Description
*SQL*Plus Quick Reference*	B31190-01	Shows iSQL*Plus buttons and icons, and iSQL*Plus and SQL*Plus command syntax.
Oracle Database JDBC Developer's Guide	B31224-01	Provides guidance for developers of JDBC-based applications and applets. This document can be read by anyone with an interest in JDBC programming, but assumes at least some prior knowledge of Java, Oracle PL/SQL, and Oracle databases.
Oracle Database JPublisher User's Guide	B31226-01	Describes the JPublisher utility for Java programmers who want classes in their applications to correspond to SQL or PL/SQL entities or server-side Java classes. JPublisher also provides features supporting Web services call-ins to the database and call-outs from the database.
Oracle Database SQLJ Developer's Guide	B31227-01	Describes how SQLJ enables application programmers to embed SQL operations in Java code in a way that is compatible with the Java design philosophy.
Application Development		
Oracle XML DB Developer's Guide	B28369-01	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 XML Developer's Kit Programmer's Guide	B28394-01	Introduces you to the Oracle XML Developer's Kit and how the various language components of the XDK can work together to generate and store XML data in a database or in a document outside the database. Examples and sample applications are introduced where possible.
Oracle Database Advanced Application Developer's Guide	B28424-01	Explains topics that experienced application developers reference repeatedly.
Oracle Database Data Cartridge Developer's Guide	B28425-01	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.
Application Express		
Oracle Database 2 Day + Application Express Developer's Guide	B28551-01	Offers an introduction to application development using the Oracle Application Express development environment. This guide contains instructions that lead you through the process of setting up your development environment to quickly build an application. Through a series of exercises, you become familiar with basic functionality, the product user interface, and the underlying concepts to enable you to develop professional, fast, and secure applications.

Table 2–2   (Cont.) Oracle® Database Server Library

Documentation	Part Number	Description
Oracle Application Express Advanced Tutorials	B28842-01	Features a series of tutorials that demonstrate how to use the Oracle Application Express development environment to develop Web applications. Each tutorial offers step-by-step instructions that explain how to create applications and different types of application components.
Oracle Application Express Release Notes	B32257-01	Contains important information not included in the Oracle Application Express documentation.
Oracle Application Express Application Builder User's Guide	B32258-01	Functions as a comprehensive, task-based reference on using the Oracle Application Express development environment. Oracle Application Express is a rapid Web application development tool for the Oracle database. Using only a Web browser and limited programming experience, you can develop professional, fast, and secure applications. Deployed applications require only a browser and access to an Oracle database running Application Express.
Application Reference		
Oracle Database XML C API Reference	B28388-01	Provides information about XML related C APIs. It primarily lists the syntax of functions, methods, and procedures associated with these API.
Oracle Database XML C++ API Reference	B28389-01	Provides information about XML related C APIs. It primarily lists the syntax of functions, methods, and procedures associated with these APIs.
Backup and Recovery		
Oracle Database Backup and Recovery User's Guide	B28270-01	Provides in-depth information on the mechanics of backup and recovery, and a guide to performing complex and less frequently performed backup and recovery tasks, including user-managed backup and recovery and performance tuning of backup and recovery.
Oracle Database Backup and Recovery Reference	B28273-01	Describes RMAN syntax and recovery catalog views.
Data Warehousing		
Oracle Data Guard Concepts and Administration	B28294-01	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.

Table 2–2   (Cont.) Oracle® Database Server Library

Documentation	Part Number	Description
Oracle Data Guard Broker	B28295-01	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. This guide provides comprehensive descriptions and examples for using both the command-line interface and the Oracle Data Guard Manager graphical user interface.
Oracle Database Data Warehousing Guide	B28313-01	Provides conceptual, reference, and implementation material for using Oracle Database 10 in data warehouses. It covers the full range of data warehousing activities, from physical database design to advanced calculation techniques.
Oracle Database 2 Day + Data Warehousing Guide	B28314-01	Teaches you how to perform common day-to-day tasks necessary to implement and administer a data warehouse.
Language and Interface		
Oracle Database PL/SQL Language Reference	B28370-01	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 PL/SQL Packages and Types Reference	B28419-01	Describes the PL/SQL packages and defined types supplied with Oracle Database. Packages are listed alphabetically with syntax, procedures and functions, and parameters described for each package.
*Pro*C/C++ Programmer's Guide*	B28427-01	Describes how to develop C++ programs that use the SQL and PL/SQL database languages to access and manipulate Oracle data.
*Pro*FORTRAN Supplement to the Oracle Precompilers Guide*	B31229-01	Describes how to write FORTRAN programs that use SQL to access and manipulate Oracle data.
*Pro*PL/1 Supplement to the Oracle Precompilers Guide*	B31230-01	Describes how to write PL/1 programs that use SQL to access and manipulate Oracle data. An understanding of the material in the Programmer's Guide to the Oracle Precompilers is assumed.
Oracle Database Programmer's Guide to the Oracle Precompilers	B31231-01	This guide is a comprehensive user's guide and on-the-job reference to the Pro*COBOL and Pro*FORTRAN precompilers. It shows you step-by-step how to develop applications that use the powerful database language SQL to access and manipulate Oracle data.
Networking and Security		
Oracle Database Net Services Administrator's Guide	B28316-01	Explains how to plan, configure, and manage enterprise-wide connectivity with Oracle Net Services.

Table 2–2 (Cont.) Oracle® Database Server Library

Documentation	Part Number	Description
Oracle Database Net Services Reference	B28317-01	Contains a complete listing and description of the control utility commands and configuration file parameters available for managing components of Oracle Net Services.
Oracle Database 2 Day + Security Guide	B28337-01	A database administration quick start guide that teaches you how to perform day-to-day database security tasks. The goal of this book is to help you understand the concepts behind Oracle database security. This book teaches you how to perform all common security tasks needed to secure your database. The knowledge you gain from completing the tasks in Oracle Database 2 Day + Security Guide helps you not only to better secure your data, but to meet common regulatory compliance requirements, such as the Sarbanes-Oxley Act.
Oracle Label Security Administrator's Guide	B28529-01	Describes how to use Oracle Label Security to protect sensitive data. It explains the basic concepts behind label-based security and provides examples to show how it is used.
Oracle Database Advanced Security Administrator's Guide	B28530-01	Provides an overview and detailed configuration information about database advanced security features. These features include network encryption, strong authentication (RADIUS, Kerberos, and SSL), and centralized user management that uses the Oracle Identity Management infrastructure (Enterprise User Security). The audience for this guide includes network security and enterprise user security administrators.
Oracle Database Security Guide	B28531-01	Provides a comprehensive overview of security for this database release, featuring security as a necessary, primary focus for Oracle installations. It provides the conceptual framework for understanding security requirements and threats as well as introducing the features, techniques, and policies for meeting those needs with Oracle products. The audience for this book includes system managers as well as users, database administrators, and application developers.
Oracle Real Application Clusters		
Oracle Database 2 Day + Real Application Clusters Guide	B28252-01	Describes how to install, configure, and administer Oracle Clusterware and Oracle Real Application Clusters on a two-node system using Red Hat Linux.
Oracle Real Application Clusters Administration and Deployment Guide	B28254-01	Explains how to administer Oracle Real Application Clusters.

Table 2–2   (Cont.) Oracle® Database Server Library

Documentation	Part Number	Description
Oracle Real Application Clusters Installation Guide for Linux and UNIX	B28264-01	Explains Oracle Real Application Clusters setup and configuration procedures for Linux and UNIX as well as how to use Oracle tools for software installation and database creation.
Oracle Text		
Oracle Ultra Search Administrator's Guide	B28330-01	Describes how to build web-based query applications using Oracle Ultra Search. Topics include crawling, indexing, and searching text content in databases or HTML pages. Java Server Pages (JSP) web-application examples are provided.
Others		
Oracle Database Object-Relational Developer's Guide	B28371-01	Describes user-defined object data types and how to use these data types to model complex real-world entities as objects in the database.
Oracle Database SecureFiles and Large Objects Developer's Guide	B28393-01	Describes use of large object data types BLOB, CLOB, NCLOB, and BFILE in application development. You can use these data types to store and manipulate unstructured and semi-structured data in binary or character format. This book also describes APIs for working with LOBs in supported programmatic environments. New APIs that allow you to use LOBs 8 terabytes in size or larger, depending on your configuration, are described.
Streams and Replication		
Oracle Streams Concepts and Administration	B28321-01	Contains conceptual information about Oracle Streams and information about configuring, administering, and monitoring an Oracle Streams environment. Also includes detailed examples of using Oracle Streams for various purposes.
Oracle Streams Replication Administrator's Guide	B28322-01	Contains conceptual information about Oracle Streams replication and about configuring, administering, and monitoring an Oracle Streams replication environment. Also includes detailed examples of configuring Oracle Streams replication environments.
Oracle Database Advanced Replication	B28326-01	Contains conceptual information about Oracle Advanced Replication. Also includes information about planning an Advanced Replication environment, an introduction to the Replication Management tool in Oracle Enterprise Manager, and information about troubleshooting Advanced Replication problems.

Table 2–2   (Cont.) Oracle® Database Server Library

Documentation	Part Number	Description
Oracle Database Advanced Replication Management API Reference	B28327-01	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. Also includes reference information about data dictionary views that are important for Advanced Replication and security considerations for an Advanced Replication environment.
Oracle Streams Advanced Queuing Java API Reference	B28392-01	Describes features of application development and integration using the Oracle messaging system, Advanced Queuing. Includes the PL/SQL, C, Visual Basic, Java, and JMS interfaces to AQ and gateways to non-Oracle messaging systems.
Oracle Database Advanced Queuing User's Guide	B28420-01	Describes features of application development and integration using the Oracle messaging system, Advanced Queuing. Includes the PL/SQL, C, Visual Basic, Java, and JMS interfaces to AQ and gateways to non-Oracle messaging systems.

2.1.3 Oracle® Database 11g OLAP Documentation

Table 2–3   Oracle® Database OLAP Documentation

Documentation	Part Number	Description
Oracle OLAP User's Guide	B28124-01	Explains how applications developers and DBAs can develop business intelligence applications, design and develop dimensional data stores (analytic workspaces), and administer Oracle Database with the OLAP option.
Oracle OLAP DML Reference	B28126-01	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	B28127-01	Assists Java programmers who are responsible for creating applications that perform analysis using Oracle OLAP.
Oracle OLAP Java API Reference	B28128-01	Describes the classes and methods in the Oracle OLAP Java API for querying analytic workspaces and relational data warehouses.

2.1.4 Oracle® Database 11g Data Mining Documentation

Table 2–4   Oracle® Database Data Mining Documentation

Documentation	Part Number	Description
Oracle Data Mining Concepts	B28129-01	Discusses the basic concepts underlying Oracle Data Mining. Details of programming with the Java and PL/SQL interfaces are discussed in the Oracle Data Mining Application Developer's Guide.
Oracle Data Mining Administrator's Guide	B28130-01	Describes how to install the Oracle Data Mining software and how to perform other administrative functions common to all Oracle Data Mining environments on both UNIX and Windows platforms.
Oracle Data Mining Application Developer's Guide	B28131-01	Describes how to use the Oracle Data Mining PL/SQL and Java APIs and the SQL scoring functions to create data mining applications.
Oracle Data Mining Java API Reference	B28132-01	Describes the classes and methods in the Oracle Data Mining Java API, the JDM JSR-73-compliant API for data mining. The Java API supports a full range of data mining activities, including model building and scoring, data preparation, and import/export of models.

2.1.5 Oracle® Database 11g Spatial Documentation

Table 2–5   Oracle® Database Spatial Documentation

Documentation	Part Number	Description
Oracle Database Semantic Technologies Developer's Guide	B28397-01	Provides usage and reference information about Oracle Database support for semantic technologies, specifically Resource Description Framework (RDF) and a subset of the Web Ontology Language (OWL).
Oracle Spatial GeoRaster Developer's Guide	B28398-01	Describes GeoRaster, a feature of Oracle Spatial that lets you store, index, query, analyze, and deliver GeoRaster data, that is, raster image data and its associated Spatial vector geometry data, plus metadata.
Oracle Spatial Topology and Network Data Models Developer's Guide	B28399-01	Provides usage and reference information about the topology data model and network data model capabilities of Oracle Spatial.
Oracle Spatial Developer's Guide	B28400-01	Provides usage and reference information for indexing and storing spatial data and for developing spatial applications. Includes an appendix about Oracle Locator 10 and its relationship to Oracle Spatial.
Oracle Spatial Java API Reference	B28401-01	Consists of interfaces and classes that support features available with Oracle Spatial, including GeoRaster and the topology and network data models, and with MapView, which is packaged with Oracle Application Server.

2.1.6 Oracle® Database 11g Java Documentation

Table 2–6   Oracle® Database Java Documentation

Documentation	Part Number	Description
Oracle Ultra Search Java API Reference	B28329-01	Contains all Oracle Ultra Search APIs. Oracle Ultra Search provides APIs for performing various tasks including obtaining query results and displaying e-mail messages.
Oracle Database XML Java API Reference	B28391-01	Describes the Java APIs for XML.
Oracle Database 2 Day + Java Developer's Guide	B28765-01	Illustrates how to access and modify data in Oracle Database using a simple Java Database Connectivity (JDBC) application. This guide uses the Oracle JDeveloper integrated development environment (IDE) to create the application.
Oracle Database Java Developer's Guide	B31225-01	Provides a general overview on how to develop, load, and execute Java applications in the database.

2.1.7 Oracle® Database 11g Linux x86 Documentation

Table 2–7   Oracle® Database Linux x86 Documentation

Documentation	Part Number	Description
Oracle Database Gateway Installation and Configuration Guide for AIX 5L Based Systems (64-Bit), HP-UX Itanium, Solaris Operating System (SPARC 64-Bit), Linux x86, and Linux x86-64	B31042-01	Describes how to install and configure Oracle Database Gateway for Sybase, Informix, Teradata, Microsoft SQL Server, and ODBC on UNIX based platforms.
Oracle Database Gateway for IMS, VSAM, and Adabas Installation and Configuration Guide for AIX 5L Based Systems (64-Bit), HP-UX Itanium, Solaris Operating System (SPARC 64-Bit), Linux x86, and Linux x86-64	B32526-01	Describes the installation and configuration of Oracle Transparent Gateway for IMS, VSAM, and Adabas, which enables Oracle client applications to access IMS, VSAM, and Adabas data through Structured Query Language (SQL). The gateway, with the Oracle Database, creates the appearance that all data is located on a local Oracle Database, even though the data can be widely distributed.

2.1.8 Oracle® Database 11g Release Notes Documentation

Table 2–8   Oracle® Database Release Notes Documentation

Documentation	Part Number	Description
Oracle Database Release Notes for Linux	B32001-01	These Release Notes contain important information not included in the Oracle Database documentation for Linux.
Oracle Warehouse Builder Release Notes	B40098-01	Contains important information not included in the Oracle Warehouse Builder documentation.
Oracle Database Gateway Release Notes for Linux	E10398-01	Contains important information that was not included in the platform-specific or product-specific documentation for this release.

2.1.9 Oracle® Database 11g Installation Guide Documentation

Table 2–9   Oracle® Database Installation Guide Documentation

Documentation	Part Number	Description
Oracle Grid Infrastructure Installation Guide for Linux	B28263-01	Provides configuration information for network and system administrators, and database installation information for database administrators (DBAs) who install and configure Oracle Clusterware for Linux.
Oracle Database Installation Guide for Linux	B32002-01	Describes how to install and configure Oracle Database on Linux.
Oracle Database Client Installation Guide for Linux	B32003-01	Describes how to install and configure Oracle Database client on Linux.
Oracle Database Quick Installation Guide for Linux x86	B32281-01	Describes how to quickly install Oracle Database on Linux x86 systems.
Oracle Database Client Quick Installation Guide for Linux x86	B32282-01	Describes how to quickly install Oracle Database client on Linux x86 systems.
Oracle Database Examples Installation Guide	E10402-01	Describes how to install and configure the products available on the Oracle Database Examples installation media.

2.1.10 Oracle® Database 11g Documentation Library

Table 2–10   Oracle® Database Documentation Library

Documentation	Part Number	Description
Oracle Database Readme	B28280-01	Describes important last-minute features and changes not included in Oracle Database Documentation Library.
Oracle Database Licensing Information	B28287-01	Describes Oracle Database licensing.
Oracle Database Documentation Readme	Q15595-01	Describes Oracle Database documentation.