

Oracle Database 12c New Features for Developers

Oracle Database 12c introduces a new multitenant architecture that makes it easy to deploy and manage database clouds. Oracle 12c is a pluggable database environment, where we can plug multiple databases into single database container. All these databases then share same background processes and memory. This helps in reducing the overhead of managing multiple databases.

I have tried to compile some of the important new features of Oracle Database 12c. Below are the top 15 new features of Oracle Database 12c for Oracle Developer & professional.

1. Sequence as Default Value

With Oracle Database 12c, we can directly assign sequence nextval as a default value for a column, So you no longer need to create a trigger to populate the column with the next value of sequence, you just need to declare it with table definition.

Example:

```
create sequence test_seq start with 1 increment by 1 nocycle;

create table test_tab
(
    id number default test_seq.nextval primary key
);
```

2. Invisible column:

Oracle Database 12c provides you the Invisible column feature. A Column defined as invisible, will not appear in generic queries (select * from). An Invisible Column need to be explicitly referred to in the SQL statement or condition. Also invisible column must be explicitly referred in INSERT statement to insert the database into invisible columns.

Example:

```
SQL> create table my_table
 2  (
 3  id number,
 4  name varchar2(100),
 5  email varchar2(100),
 6  password varchar2(100) INVISIBLE
 7  );

SQL> ALTER TABLE my_table MODIFY (password visible);
```

3. Multiple indexes on the same column

Before Oracle Database 12c, we could not have multiple indexes on a single column. In Oracle Database 12c a column may have multiple indexes but all should be of different types. Like a column may have B-Tree and BitMap Index both. But, only one index will be used at a given time.

4. VARCHAR2 length up to 32767

From Oracle Database 12c, a varchar2 column can be sized upto 32767, which was earlier 4000. The maximum size of the VARCHAR2, NVARCHAR2, and RAW data types has been increased from 4,000 to 32,767 bytes. Increasing the allotted size for these data types allows users to store more information in character data types before switching to large objects (LOBs).

5. Top-N feature

A Top-N query is used to retrieve the top or bottom N rows from an ordered set. Combining two Top-N queries gives you the ability to page through an ordered set

Example:

```
SQL> SELECT value
  2 FROM mytable
  3 ORDER BY value DESC
  4 FETCH FIRST 10 ROWS ONLY;
```

6. IDENTITY Columns

In Oracle Database 12c, We can define Table columns with SQL keyword IDENTITY which is a American National Standards Institute (ANSI) SQL keyword. Which are auto-incremented at the time of insertion (like in MySQL).

Example:

```
SQL> create table my_table
  2 (
  3 id number generated as identity,
  4 name varchar2(100),
  5 email varchar2(100),
  6 password varchar2(100) INVISIBLE
  7 );
```

7. With Clause improvement

In Oracle 12c, we can declare PL/SQL functions in the WITH Clause of a select statement and use it as an ordinary function. Using this construct results in better performance as compared with schema-level functions

Example:

```
SQL> WITH
  2 FUNCTION f_test(n IN NUMBER) RETURN NUMBER IS
  3 BEGIN
  4 RETURN n+1;
  5 END;
  6 SELECT f_test(1)
  7 FROM dual
  8 ;
```

8. Cascade for TRUNCATE and EXCHANGE partition.

With Oracle Database 12c, The TRUNCATE can be executed with CASCADE option which will also delete the child records.

9. Online RENAME/MOVE of Datafiles

Oracle Database 12c has provided a simple way to online renamed or moved data files by simply "ALTER DATABASE MOVE DATAFILE" command. Data files can also be migrated online from ASM to NON-ASM and NON-ASM to ASM easily now.

Examples:

Rename datafile:

```
SQL> ALTER DATABASE MOVE DATAFILE '/u01/oradata/indx.dbf' TO '/u01/oradata/indx_01.dbf';
```

Move Datafile:

```
SQL> ALTER DATABASE MOVE DATAFILE '/u01/oradata/indx.dbf' TO '/u01/oradata/orcl/indx.dbf';
```

NON-ASM to ASM:

```
SQL> ALTER DATABASE MOVE DATAFILE '/u01/oradata/indx.dbf' TO '+DISKGROUP_DATA01';
```

10. Move table partition to different Tablespace online

From Oracle 12c, it become very easy to move Table Partition to different tablespace and does not require complex steps

Example:

```
SQL> ALTER TABLE MY_LARGE_TABLE MOVE PARTITION MY_LARGE_TABLE_PART1 TO TABLESPACE USERS_NEW;
```

11. Temporary Undo

Before Oracle Database 12c, undo records of temporary tables used to be stored in undo tablespace. With the temporary undo feature in Oracle Database 12c, the undo records of temporary tables can now be stored in a temporary table instead of stored in undo tablespace. The main benefits of temporary undo are 1) Low undo tablespace usages 2) less redo data generation. For using this feature Compatibility parameter must be set to 12.0.0 or higher and TEMP_UNDO_ENABLED initialization parameter must be Enabled.

12. DDL logging

By using the ENABLE_DDL_LOGGING initiation parameter in Oracle Database 12c, we can now log the DDL action into xml and log files to capture when the drop or create command was executed and by whom under the \$ORACLE_BASE/diag/rdbms/DBNAME/log/ddl location. The parameter can be set at the database or session levels.

Example:

```
SQL> ALTER SYSTEM SET ENABLE_DDL_LOGGING=TRUE;
```

13. PGA_AGGREGATE_LIMIT parameter

Oracle Database 12c has provided us a way to limit PGA by PGA_AGGREGATE_LIMIT parameter. Before Oracle Database 12c there was no option to limit and control the PGA size. Oracle will automatically abort the session that holds the most untenable PGA memory when PGA limits exceeds the defined value.

14. SQL statement in RMAN

From Oracle Database 12c, we can execute any SQL and PL/SQL commands in RMAN without SQL prefix

Example:

```
RMAN> SELECT username,machine FROM v$session;
```

15. Turning off redo for Data Pump the import

The new TRANSFORM option, `DISABLE_ARCHIVE_LOGGING`, to the `impdp` command line causes Oracle Data Pump to disable redo logging when loading data into tables and when creating indexes. This feature provides a great relief when importing large tables, and reduces the excessive redo generation, which results in quicker imports. This attribute applies to tables and indexes.

Example:

```
impdp directory=mydir dumpfile=mydmp.dmp logfile=mydmp.log TRANSFORM=DISABLE_ARCHIVE_LOGGING:Y
```

New Features for Developers in Oracle 11g

Oracle 11g has introduced many new features in SQL & PL/SQL, which are very useful for a developer. I have compiled some of the most important features, which a Oracle developer should know.

Reference: <http://www.oracle.com/technetwork/articles/grid/index-099021.html>

1. Compound Triggers: A compound trigger is actually four different triggers defined as one. it not only it saves lot of code writing, but also resolves the issue where same set of session variables share common data

2. Trigger Follows/Precedes Clause: The Oracle 11g trigger syntax now includes the Follows/Precedes clause to

guarantee execution order for triggers defined with the same timing point. Follows clause in a trigger specifies that the current trigger would follow the execution of specified triggers.

3. Creating Trigger in DISABLED mode: Oracle 11g provides flexibility to create a Trigger in DISABLED mode also. They remain deactivated until they are enabled. Prior to Oracle 11g, a trigger can be created in ENABLED mode only.

4. Read Only Tables: In Oracle 11g, a table can be set READ ONLY mode to restrict write operations on the table. A table can be altered to toggle over READ ONLY and READ WRITE modes.

5. Invisible Indexes: An invisible index is ignored by the optimizer unless you explicitly set the OPTIMIZER_USE_INVISIBLE_INDEXES initialization parameter to TRUE or using INDEX hint. Making an index invisible is an alternative to making it unusable. It can be converted to VISIBLE mode for auto consideration by the optimizer

6. Virtual Columns: Oracle 11g allows you to create a "virtual column", an empty column that contains a function upon other table columns (the function itself is stored in the data dictionary). They show same behavior as other columns in the table in terms of indexing and statistics.

- [Oracle 11g - Virtual Columns](#)

7. Continue Statement: The continue statement passes control to the next iteration. The CONTINUE statement exits the current iteration of a loop, either conditionally or unconditionally, and transfer control to the next iteration.

8. LISTAGG Analytic Functions: Oracle 11g has introduced. LISTAGG analytic function to aggregate the result set in multiple rows into one single column.

- [Oracle 11g New String Aggregation Techniques](#)

9. REGEXP_COUNT: Oracle 11g introduced REGEXP_COUNT in regular expressions, used to count the occurrence of a character or string expression in another string.

10. Pivot and Unpivot: The PIVOT operator takes data in separate rows, aggregates it and converts it into columns. The UNPIVOT operator converts column-based data into separate rows.

- [Oracle 11g Pivot data Rows to Columns](#)

11. Result Cache: Oracle 11g has introduced a new component in SGA - Shared Pool as Result Cache to retain result-sets of SQL queries and PL/SQL functions results. By caching the results of queries, Oracle can avoid having to repeat the potentially time-consuming and intensive operations that generated the resultset.

12. Pragma Inline: A new pragma PRAGMA INLINE has been introduced to specify whether a subprogram call has to be inlines or not. Inlining replaces a subprogram call with a copy of the called subprogram

13. Direct Sequence Assignment: Prior to Oracle 11g, sequence assignment to a number variable could be done through a SELECT statement only, which requires context switching from PL/SQL engine to SQL engine. From oracle 11g we can directly assign a sequence value to a pl/sql variable.

14. SIMPLE_INTEGER Data Type: Oracle 11g has designed a new data type SIMPLE_INTEGER, SIMPLE_FLOAT, and SIMPLE_DOUBLE. They are compatible with the native compilation feature of Oracle 11g, which makes supports their faster implementation.

15. Recursive Subquery Factoring: Version 11g release 2 introduced recursive subquery factoring or the recursive with clause. This is an extension to the SQL syntax with which you can do recursive/hierarchical queries.

- [Oracle 11g - Recursive subquery factoring](#)

16. Calling Functions: In Oracle 11g, functions can now be called using Named, Positional and Mixed notation while calling from SQL SELECT statement.

17. DDL Wait Option: Oracle Database 11g, using ddl_lock_timeout parameter, when a DDL statement in the session does not get the exclusive lock, it will not error out. Instead, it will wait for sepcified seconds in ddl_lock_timeout parameter. In those seconds, Oracle continually re-tries the DDL operation until it's successful or the time expires.

18. Deprecation of SQL Plus: Oracle has deprecated the use of SQL Plus (isqlplus and sqlplusw) since Oracle 11g. Oracle 11g supports SQL plus through command line and recommends the use of SQL Developer.

19. SIMPLE_INTEGER: Oracle 11g has designed a new data type SIMPLE_INTEGER, SIMPLE_FLOAT, and SIMPLE_DOUBLE keeping in view the hardware requirements and expectations with an Integer value. They are compatible with the native compilation feature of Oracle 11g, which makes supports their faster implementation.

20. New PL/SQL Compiler Warning: Oracle Database 11g has a new warning called PLW-06009 to warn you "dangerous practice of leaving the WHEN OTHERS THEN NULL" during compile time. enable this warning you have to set this session parameter plsql_warnings = 'enable:all'

21. Behaviour: Oracle 11g no longer keeps PL/SQL as a pure procedural language. Now PL/SQL is efficient to implement Object Oriented Concepts, thus making it comparable to other programming languages like C++ and JAVA.

22. New Partitions: By using Oracle 11g new feature called Reference Partitioning we can create partitions identical to parent table in child table. Oracle 11g has also provided us a way to define partitions in an intelligent manner by System Partitioning.

- [Oracle 11g - Reference Partitions](#)

- [Oracle 11g - System Partitions](#)