|  |
| --- |
| , RD Dep. |
| Introduction  **Oracle Data Manipulation Language** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial Version | Siarhei Kandrashevich | 15-Jan-2014 |  |  |
|  |  |  |  |  |  |

Contents

[1. Oracle Data Manipulation Language (DML) 3](#_Toc380562025)

[1.1. Data Manipulation Language (DML) Statements 3](#_Toc380562026)

[2. The INSERT statement 3](#_Toc380562027)

[2.1. Single table INSERT 3](#_Toc380562028)

[2.1.1. Insert one row 4](#_Toc380562029)

[2.1.2. Insert multiple rows 4](#_Toc380562030)

[2.2. Multi-table INSERT 4](#_Toc380562031)

[2.2.1. Unconditional 4](#_Toc380562032)

[2.2.2. Conditional 4](#_Toc380562033)

[2.3. Direct-Path INSERT (also known as APPEND) 5](#_Toc380562034)

[2.4. Using the WITH CHECK OPTION Clause 5](#_Toc380562035)

[3. The UPDATE statement 5](#_Toc380562036)

[3.1. Update with subquery 5](#_Toc380562037)

[3.2. Setting DEFAULT value for column 6](#_Toc380562038)

[3.3. Subquery restriction clause 6](#_Toc380562039)

[4. DELETE Statement 6](#_Toc380562040)

[4.1. DELETE Statement purpose 6](#_Toc380562041)

[4.2. Deleting data from updatable (inline) views 6](#_Toc380562042)

[5. External Sources 7](#_Toc380562043)

# Oracle Data Manipulation Language (DML)

## Data Manipulation Language (DML) Statements

Data manipulation language (DML) statements access and manipulate data in existing schema objects. These statements *do not implicitly commit* the current transaction. The data manipulation language statements are [1]:

* CALL
* DELETE
* EXPLAIN PLAN
* INSERT
* LOCK TABLE
* MERGE
* SELECT
* UPDATE

The SELECT statement is a limited form of DML statement in that it can only access data in the database [1]. It cannot manipulate data stored in the database, although it can manipulate the accessed data before returning the results of the query.

DML statements are the most frequently used SQL statements and enable you to:

* Retrieve or fetch data from one or more tables or views (SELECT).
* Add new rows of data into a table or view (INSERT) by specifying a list of column values or using a subquery to select and manipulate existing data.
* Change column values in existing rows of a table or view (UPDATE).
* Update or insert rows conditionally into a table or view (MERGE).
* Remove rows from tables or views (DELETE).
* View the execution plan for a SQL statement (EXPLAIN PLAN).
* Lock a table or view, temporarily limiting access by other users (LOCK TABLE).

Further readings are [1] (chapter 10) and [2] (chapter 7).

# The INSERT statement

The INSERT statement is used to add rows to a table, the base table of a view, a partition of a partitioned table or a subpartition of a composite-partitioned table, or an object table or the base table of an object view.

**Prerequisites**

For you to insert rows into a table, the table must be in your own schema or you must have the INSERT object privilege on the table. For you to insert rows into the base table of a view, the owner of the schema containing the view must have the INSERT object privilege on the base table. Also, if the view is in a schema other than your own, then you must have the INSERT object privilege on the view.

If you have the INSERT ANY TABLE system privilege, then you can also insert rows into any table or the base table of any view. You must also have the SELECT object privilege on the table into which you want to insert rows if the table is on a remote database.

## Single table INSERT

In a single-table insert, you insert values into one row of a table, view, or materialized view by specifying values explicitly or by retrieving the values through a subquery.

### Insert one row

For a single-table insert operation, specify a row of values to be inserted into the table or view. You must specify a value in the values clause for each column in the column list. If you omit the column list, then the values clause must provide values for every column in the table.

**INSERT** **INTO** departments

**VALUES** **(**280**,** 'Business Intelligence'**,** 121**,** 1700**);**

**INSERT** **INTO** departments

**VALUES** **((SELECT** **MAX(**department\_id**)** **FROM** departments**)+**10**,**

'Business Intelligence'**,** **DEFAULT,** 1700**);**

**INSERT** **INTO** **(**

**SELECT** employee\_id**,** last\_name**,** email**,**

hire\_date**,** job\_id**,** salary**,** commission\_pct

**FROM** employees

**)**

**VALUES** **(**208**,** 'Doe'**,** 'john\_doe@test.com'**,** **sysdate,** 'PU\_CLERK'**,** 1.2E3**,** **NULL);**

### Insert multiple rows

If you retrieve values for insert operation through a subquery, then the select list of the subquery must have the same number of columns as the column list of the INSERT statement. If you omit the column list, then the subquery must provide values for every column in the table.

**INSERT** **INTO** bonuses

**SELECT** employee\_id**,** salary**\***1.1

**FROM** employees

**WHERE** commission\_pct **>** 0.25**;**

## Multi-table INSERT

In a multi-table insert, you insert computed rows derived from the rows returned from the evaluation of a subquery into one or more tables.

### Unconditional

Specify ALL followed by multiple “insert into” clauses to perform an unconditional multi-table insert. Oracle Database executes each “insert into” clause once for each row returned by the subquery.

### Conditional

Oracle Database filters each “insert into” clause through the corresponding WHEN condition, which determines whether that “insert into” clause is executed. Each expression in the WHEN condition must refer to columns returned by the select list of the subquery. A single multi-table insert statement can contain up to 127 WHEN clauses.

*The ELSE clause*

For a given row, if no WHEN clause evaluates to true, then:

* If you have specified an ELSE clause, then the database executes the INTO clause list associated with the ELSE clause.
* If you did not specify an else clause, then the database takes no action for that row.

#### Full check

If you specify ALL, the default value, then the database evaluates each WHEN clause regardless of the results of the evaluation of any other WHEN clause. For each WHEN clause whose condition evaluates to true, the database executes the corresponding INTO clause list.

#### Short-circuit check

If you specify FIRST, then the database evaluates each WHEN clause in the order in which it appears in the statement. For the first WHEN clause that evaluates to true, the database executes the corresponding INTO clause and skips subsequent WHENclauses for the given row.

## Direct-Path INSERT (also known as APPEND)

Out of course scope, for more information about this please see “APPEND Hint” and “NOAPPEND Hint” in [1].

## Using the WITH CHECK OPTION Clause

The following statement is legal even though the third value inserted violates the condition of the subquery where clause:

**INSERT** **INTO** **(**

**SELECT** department\_id**,** department\_name**,** location\_id

**FROM** departments **WHERE** location\_id **<** 2000

**)**

**VALUES** **(**300**,** 'Business Intelligence'**,** 3000**);**

However, THE FOLLOWING statement is illegal because it contains THE WITH CHECK OPTION clause:

**INSERT** **INTO** **(**

**SELECT** department\_id**,** department\_name**,** location\_id

**FROM** departments **WHERE** location\_id **<** 2000

**WITH** **CHECK** **OPTION**

**)**

**VALUES** **(**300**,** 'Business Intelligence'**,** 3000**);**

Further readings are [1] (chapter 18: INSERT), [2]

# The UPDATE statement

Use the UPDATE statement to change existing values in a table or in the base table of a view or the master table of a materialized view.

For you to update values in a table the table must be in your own schema or you must have the UPDATE object privilege on the table.

For you to update values in the base table of a view:

* You must have the UPDATE object privilege on the view, and
* Whoever owns the schema containing the view must have the UPDATE object privilege on the base table.

The UPDATE ANY TABLE system privilege also allows you to update values in any table or in the base table of any view. You must also have the SELECT object privilege on the object you want to update if:

* The object is on a remote database or
* The SQL92\_SECURITY initialization parameter is set to TRUE and the UPDATE operation references table columns, such as the columns in a WHERE clause.

## Update with subquery

* Specify a subquery that returns exactly one row for each row updated.
* If you specify only one column in the *update\_set\_clause*, then the subquery can return only one value.
* If you specify multiple columns in the *update\_set\_clause*, then the subquery must return as many values as you have specified columns.
* If the subquery returns no rows, then the column is assigned a null.

## Setting DEFAULT value for column

Specify DEFAULT to set the column to the value previously specified as the default value for the column.

If no default value for the corresponding column has been specified, then the database sets the column to null.

You cannot specify DEFAULT if you are updating a view.

## Subquery restriction clause

* **WITH READ ONLY**Specify WITH READ ONLY to indicate that the table or view cannot be updated.
* **WITH CHECK OPTION**Specify WITH CHECK OPTION to indicate that Oracle Database prohibits any changes to the table or view that would produce rows that are not included in the subquery.   
  When used in the subquery of a DML statement, you can specify this clause in a subquery in the FROM clause but not in subquery in the WHERE clause.
* **CONSTRAINT *constraint***Specify the name of the CHECK OPTION constraint. If you omit this identifier, then Oracle automatically assigns the constraint a name of the form SYS\_C*n*, where n is an integer that makes the constraint name unique within the database.

# DELETE Statement

## DELETE Statement purpose

Use the DELETE statement to remove rows from:

* An unpartitioned or partitioned table
* The unpartitioned or partitioned base table of a view
* The unpartitioned or partitioned container table of a writable materialized view
* The unpartitioned or partitioned master table of an updatable materialized view

## Deleting data from updatable (inline) views

To be updatable, the view must not contain any of the following constructs:

1. A set operator
2. A DISTINCT operator
3. An aggregate or analytic function
4. A GROUP BY, ORDER BY, MODEL, CONNECT BY, or START WITH clause
5. A collection expression in a SELECT list
6. A subquery in a SELECT list
7. A subquery designated WITH READ ONLY
8. Joins, with some exceptions, as documented in [5]

If you want a **(inline) join view to be updatable**, then all of the following conditions must be true:

* The DML statement must affect only one table underlying the join.
* For an INSERT statement: all columns into which values are inserted must come from a key-preserved table. *A key-preserved table is one for which every primary key or unique key value in the base table is also unique in the join view.*
* For an UPDATE statement: all columns updated must be extracted from a key-preserved table.
* For a DELETE statement: you can delete from a join view provided there is one and only one key-preserved table in the join. If the join results in more than one key-preserved table, then Oracle Database deletes from the first table named in the FROM clause.

See “DML Statements and Join Views” in [5] for further information.

# External Sources

1. Oracle Database SQL Language Reference 11g Release 2 (11.2) E26088-02
2. Oracle Database Concepts 11gRelease 2 (11.2) E25789-01
3. ISO/IEC 9075-1 Part 1 Framework (SQL/Framework)
4. ISO/IEC 9075-2 Part 2 Foundation (SQL/Foundation)
5. ISO/IEC 9075-11 Part 11 Information and Definition Schemas (SQL/Schemata)
6. Oracle Database Administrator's Guide 11g Release 2 (11.2) E25494-02
7. **Oracle Database SQL Language Reference 12g Release12 (12.1) E17209-15**