**Data Manipulation Language (DML) Commands in Oracle:**

DML stands for Data Manipulation Language. DML commands are basically used to INSERT, UPDATE, and DELETE data in a database table. That means DML statements affect the records in a table

Syntax1 :

In this syntax values for all the columns are given in the same sequence they are defined in the table

INSERT INTO tablename VALUES ( VALUE1 , VALUE2, VALUE3…..);

**Syntax2:** In this syntax we have specified the column names explicitly and we need to provide the values for the specified columns only. The missing columns will take “null” by default.

INSERT INTO tablename (REQUIRED COLUMN NAMES ) VALUES ( VALUE1, VALUE2…);

##### ****to insert NULLs into a table****

##### **syntax:**

##### **INSERT INTO tablename VALUES ( NULL , NULL , NULL) ;**

##### **INSERT INTO tablename ( column1, column2 , column3 ) VALUES ( NULL, NULL , NULL);**

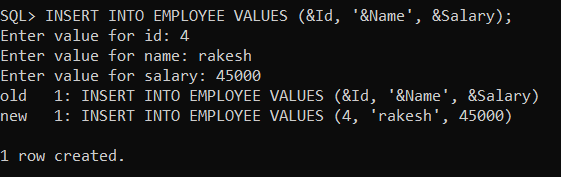
##### ****to insert multiple rows into a table****

##### **there are 2 ways to perform this operation:**

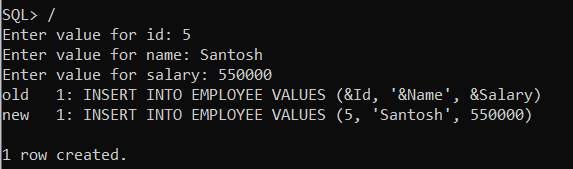
1. &: to insert values to columns dynamically (we can change values)

##### **Syntax1(&): For all columns**

INSERT INTO tablename VALUES ( &col1, &col2 , &col3 );



The / is used to re-execute the last executed SQL query in SQL Plus editor. Once you type the / button and press the enter key then again it will ask you to enter the Id, Name, and salary column values one by one



##### ****Syntax2(&): For required columns****

##### **INSERT INTO tablename ( col1 , col2 , col3 ) VALUES (&col1, &col2 , &col3 );**

##### Data Manipulation Language Commands in Oracle with Examples

##### DML Commands in Oracle with Examples

##### 

1. &&: to insert fixed values to columns (we cannot change values). We can insert values to columns in a fixed manner. If we want to change a fixed value of a column then we should “exit” from the oracle database.

##### ****Syntax1(&&): For all Columns****

##### **INSERT INTO TABLENAME VALUES ( &&col1 , &&col2 , &&col3….);**

##### 

##### Data Manipulation Language Commands in Oracle with Examples

##### 

##### DML Commands in Oracle with Examples

**UPDATE DML statement in Oracle:**

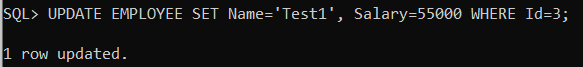
The UPDATE DML Statement in Oracle is basically used to

1. To update all rows data in a table at a time.
2. To update a specific row data in a table by using the “where” condition.

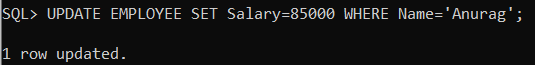
Syntax :

UPDATE TABLENAME SET colname1 = value1, colname2 = val2 WHERE CONDITION ;

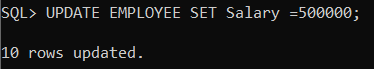
**UPDATE EMPLOYEE SET Name=’Test1′, Salary=55000 WHERE Id=3;**



**UPDATE EMPLOYEE SET Salary=85000 WHERE Name=’Anurag’;**



**UPDATE EMPLOYEE SET Salary =500000;**



We cannot use null operator with where condition

Eg:

Syntax :

UPDATE TABLENAME SET col1 = value1 , col2 = value2 WHERE colname = NULL;

The above command is syntactically right but logic is wrong , the right way for working with null is

Syntax :

UPDATE TABLENAME SET col1 = value1 , col2 = value2 WHERE colname IS NULL ;

**DELETE DML statement in Oracle:**

The DELETE DML Statement in Oracle is basically used to

1. To delete all rows from a table at a time.
2. To delete a specific row from a table by using the “where” condition.

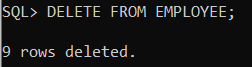
Syntax :

DELETE FROM TABLENAME WHERE CONDITION ;

**DELETE FROM EMPLOYEE WHERE Id=5;**

DELETE DML statement in Oracle

**DELETE FROM EMPLOYEE;**



##### ****Delete VS Truncate in Oracle:****

###### **Delete**

1. It is a DML command
2. It can delete a specific row from a table.
3. It supports the “WHERE” clause condition.
4. It is a temporary data deletion.
5. We can restore deleted data by using the “ROLLBACK” command.
6. Execution speed is slow as deleting operation is performed row by row / one by one manner.

###### **Truncate**

1. It is a DDL command
2. It is not possible to delete a specific row from a table.
3. It does not support the “WHERE” clause condition.
4. It is a permanent data deletion.
5. We cannot restore deleted data by using the “ROLLBACK” command.
6. Execution speed is fast as deleting a group of rows at a time.

INSERTALL /\* first go through the dql (next concept) and then learn this concept

##### ****INSERT ALL Command in Oracle:****

The INSERT ALL statement in Oracle is basically used to add multiple rows with a single INSERT statement. The important point that you need to remember is, the rows can be inserted into one table or multiple tables using only one SQL command.

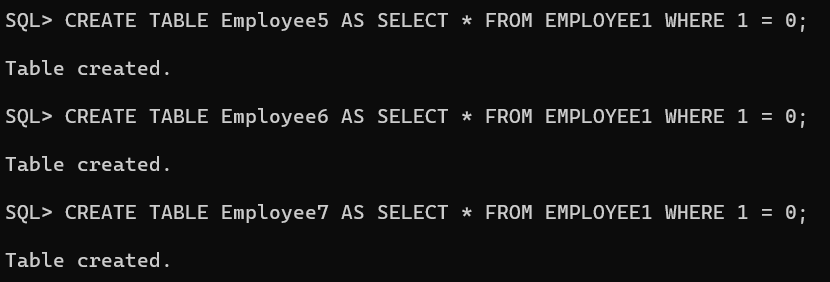
Syntax 1 : inserting single predefined table rows data to different new tables at a time.

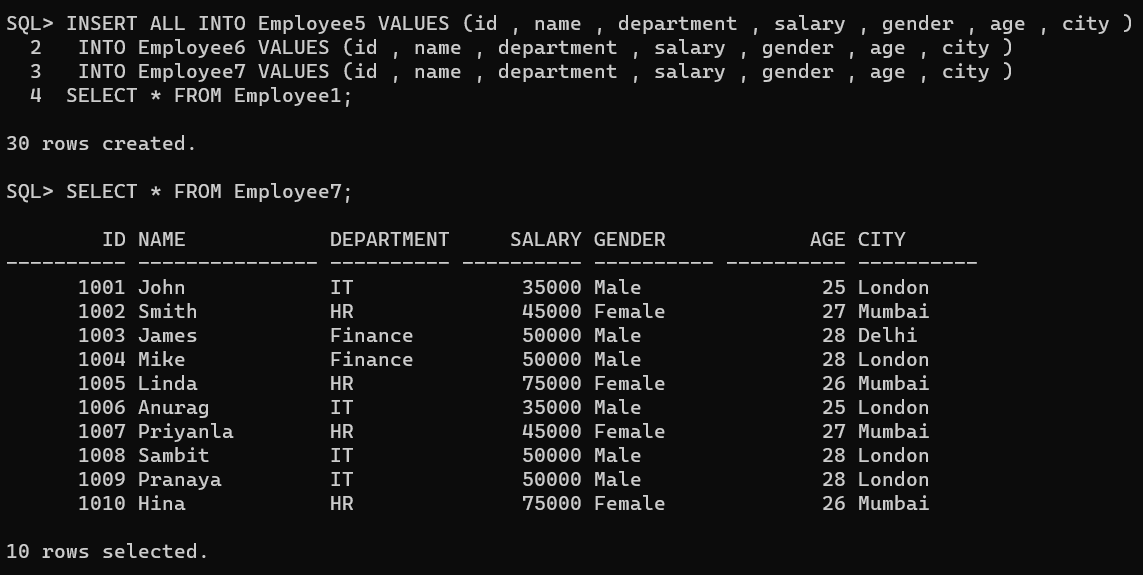
INSERT ALL INTO tablename1 VALUES ( col1,col2,col3)

INTO tablename2 VALUES ( col1,col2,col3)

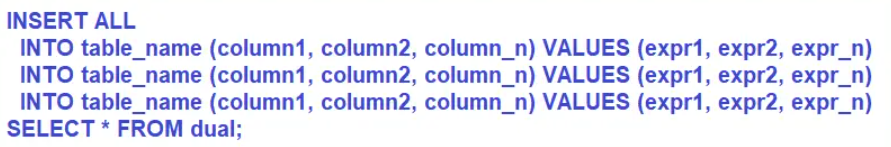
INTO tablename3 VALUES ( col1,col2,col3)

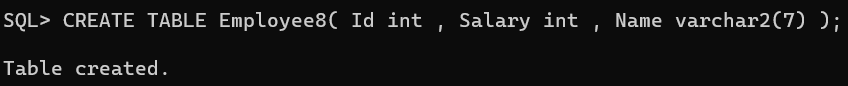
SELECT \* FROM tablename;

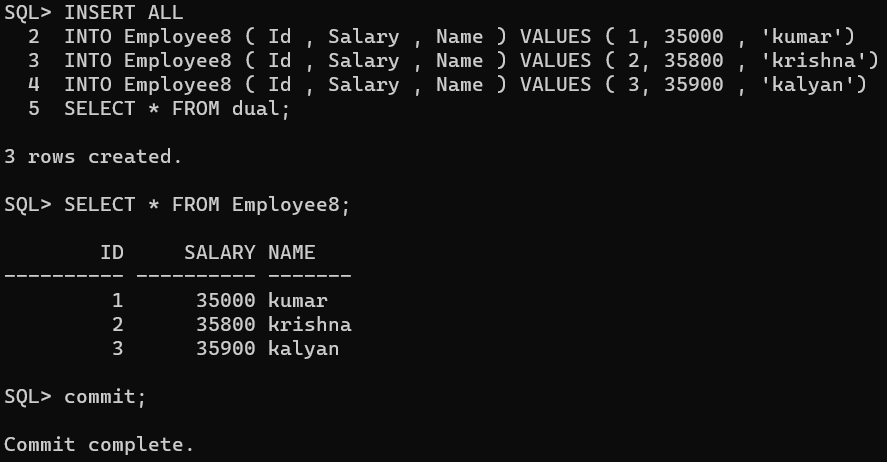




Syntax 2: inserting multiple rows data dynamically to single table at a time , here dual is not a table name it is a predefined keyword .







##### 

Syntax 3: inserting multiple rows data dynamically to multiple tables at a time , here dual is not a table name it is a predefined keyword .

CREATE TABLE Address

(

Id INT ,

EmpId INT ,

Address VARCHAR2(15)

);

INSERT ALL

INTO Employee8 (Id, Salary, Name) VALUES (4, 35000, 'Pranaya')

INTO Employee8 (Id, Salary, Name) VALUES (5, 45000, 'Kalyani')

INTO Address (Id, EmpId, Addrees) VALUES (1, 4, 'BBSR')

INTO Address (Id, EmpId, Addrees) VALUES (2, 5, 'CTC')

SELECT \* FROM dual;

