**DATA MANIPULATION COMMANDS FOR INSERTING, DELETING, UPDATING AND RETRIEVING TABLES**

**EX.NO:2**

**DATE:**

**AIM:**

To create a database using Data Manipulation Commands for inserting, deleting, updating and retrieving tables and Transaction Control statements.

**DESCRIPTION:**

**Data Manipulation Language:**

Data manipulation language (DML) statements access and manipulate data in existing tables. DML commands are the most frequently used SQL commands and is used to query and manipulate the existing database objects. Some of the commands are Insert, Select, Update, Delete.

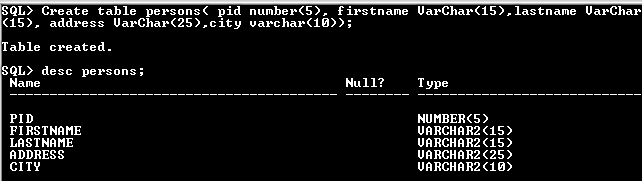
**Examples of DML:**

1. **Insert Command:** This is used to add one or more rows to a table. The values are separated by commas and the data types char and date are enclosed in apostrophes. The values must be entered in the same order as they are defined.
2. **Select Commands:** It is used to retrieve information from the table. It is generally referred to as querying the table. We can either display all columns in a table or only specify column from the table.
3. **Update Command:** It is used to alter the column values in a table. A single column may be updated or more than one column could be updated.
4. **Delete command:** After inserting row in a table we can also delete them if required. The delete command consists of a from clause followed by an optional where clause.

**INSERTING VALUES INTO TABLE**

**Create table:**

SQL>Create table persons( pid number(5), firstnameVarChar(15),lastnameVarChar(15), address VarChar(25),city varchar(10));



**INSERT COMMAND**

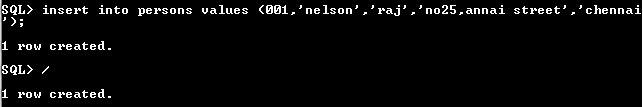
Insert command is used to insert values into table.

**Insert a single record into table.**

**Syntax:**SQL>insert into <table name> values (value list)

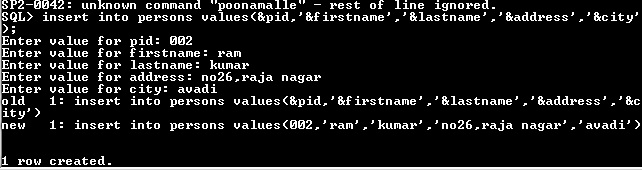
SQL>insert into persons values (001,'nelson','raj','no25,annai street','chennai');

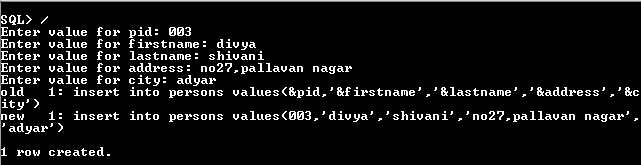
1 row created.

****

**Insert more than a record into persons table using a single insert command.**

SQL> insert into persons values(&pid,'&firstname','&lastname','&address','&city');





**Skipping the fields while inserting:**

SQL> insert into persons(pid,firstname) values(500,'prabhu');



**SELECT COMMAND**

It is used to retrieve information from the table. It is generally referred to as querying the

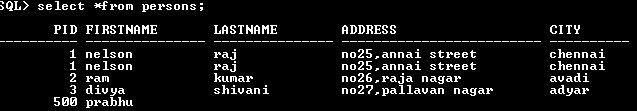
table. We can either display all columns in a table or only specify column from the table.

**Syntax:**

SQL> Select \* from tablename; // This query selects all rows from the table.

**Example:**

SQL>Select \* from persons;



**THE RETRIEVAL OF SPECIFIC COLUMNS FROM A TABLE:**

It retrieves the specified columns from the table

**Syntax:** SQL>Select column\_name1, …..,column\_name N from table name;

**Example:** SQL>Select pid, firstname from persons;



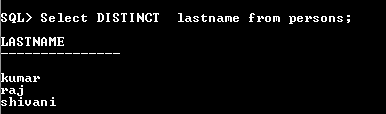
**Elimination of duplicates from the select clause:**

It prevents retrieving the duplicated values .Distinct keyword is to be used.

**Syntax:** SQL>Select DISTINCT col1, col2 from table name;

**Example:**

SQL>Select DISTINCT lastname from persons;



**SELECT COMMAND WITH WHERE CLAUSE:**

To select specific rows from a table we include ‘where’ clause in the select command. It

can appear only after the ‘from’ clause.

**Syntax:** SQL>Select column\_name1, …..,column\_name N from table name where condition;

**Example:** SQL>Select firstname, lastname from persons where pid>2;



**Select command with order by clause:**

**Syntax:** SQL>Select column\_name1, …..,column\_namen from table name where condition orderbycolmnname;

**Example:**

SQL>Select firstname, lastname from persons order by pid;



**Select command to create a table:**

**Syntax:**

SQL>create table tablename as select \* from existing\_tablename;

**Example:**

SQL>create table persons1 as select \* from persons;

Table created

**SELECT COMMAND TO INSERT RECORDS:**

**Syntax:**SQL>insert into tablename( select columns from existing\_tablename);

**Example:**SQL>insert into persons1( select \* from persons);

PID FIRSTNAME LASTNAMEADDRESS CITY PHONENO

---------- -------------------- ----------------------- -------------------- ----------- ------------------

001 nelson raj no25,annai street Chennai

100 niranjan kumar 10/25 krishna street Mumbai 999999999

102 arjun kumar 30 sundaram street coimbatore

300 gugan chand 5/10 mettu street Coimbatore

500 prabhu

**SELECT COMMAND USING IN KEYWORD:**

**Syntax:**SQL>Select column\_name1, …..,column\_namen from table name where colmnname IN (value1,value2);

**Example:** SQL>Select \* from persons where pid in (100,500);

(OR)

SQL>Select \* from persons where (pid=100 OR pid=500);

PID FIRSTNAME LASTNAMEADDRESS CITY PHONENO

---------- -------------------- ----------------------- -------------------- ----------- ------------------

100 niranjan kumar 10/25 krishna street Mumbai 999999999

500 prabhu

**SELECT COMMAND USING BETWEEN KEYWORD:**

**Syntax:**SQL>Select column\_name1, …..,column\_namen from table name where colmnname BETWEEN value1 AND value2;

**Example:** SQL>Select \* from persons where pid between 100 and 500;

PID FIRSTNAME LASTNAMEADDRESS CITY PHONENO

---------- -------------------- ----------------------- -------------------- ----------- ------------------

100 niranjan kumar 10/25 krishna street Mumbai 999999999

500 prabhu

**SELECT COMMAND USING PATTERN:**

**Syntax:**SQL>Select column\_name1, …..,column\_namen from table name where colmnname LIKE ‘% or \_‘;

**Example:** SQL>Select \* from persons where firstname like ‘nir\_n%’;

PID FIRSTNAME LASTNAMEADDRESS CITY PHONENO

---------- -------------------- ----------------------- -------------------- ----------- ------------------

100 niranjan kumar 10/25 krishna street Mumbai 999999999

**RENAMING THE FIELDNAME AT THE TIME OF DISPLAY USING SELECT STATEMENT:**

**Syntax:**SQL>Select old\_column\_namenew\_column\_namefrom table name where condition;

**Example:** SQL>Select pidpersonid from persons;

**SELECT COMMAND TO RETRIEVE NULL VALUES:**

**Syntax:**SQL>Select column\_name from table name where column\_name is NULL ;

**Example:** SQL>Select \* from persons where lastname is null;

PID FIRSTNAME LASTNAMEADDRESS CITY PHONENO

---------- -------------------- ----------------------- -------------------- ----------- ------------------

500 prabhu

**UPDATE COMMAND:**

**Syntax:**

UPDATE table\_nameSET column\_name = value [, column\_name = value]...

[ WHERE condition ];

**Example:**SQL>update persons set pid= 5 where firstname=’prabhu’;

Table updated.

**DELETE COMMAND**

**Syntax:** SQL>Delete from table where conditions;

**Example:**SQL>delete from persons where pid=500;

**1 row deleted.**

**RESULT:**

Thus the database has been created and the data has been inserted, deleted, modified, altered, updated and records are viewed based on conditions.