LAB MANUAL

FOR

Data Base Management System

Ms. Sonia Raoof

Lab 4

Objective: - To Manipulate the Operations on the table.
DML (Data Manipulation Language) Data manipulation is

- ! The retrieval of information stored in the database.
- ! The insertion of new information into the database.
- ! The deletion of information from the database.
- ! The modification of information stored by the appropriate data model. There are two types.

(i) **Procedural DML**: This requires a user to specify what data are needed and how to obtain those data.

(ii) **Non Procedural DML:** requires a user to specify what data are neededwithout specifying how to get those data.

Updating the content of a table:
In a creation situation, we may wish to change a value in a table without changing all valuesin the tuple. For this purpose, the update statement can be used.

Update table name
Set columnname = expression, columnname =expression……
Where columnname = expression;

Query: Update client_master Set state='Tamilnadu' where client_no = '0005';

Deletion Operation: -
A delete query is expressed in much the same way as a Query. We can delete whole tuple(rows) we can delete values on only particular attributes.

Deletion of all rows

Syntax:
Delete from tablename;

Deletion of a specified number of rows
Syntax:

Delete from table name
Where search condition;

Query: Delete from client_master Where client_no = '0005';

Arithmetic computation in expression/columnname lists used to select data

+	Addition	- Subtraction	
*	multiplication	** exponentiation	
/ Division		() Enclosed operation	

Examples:

Apply these queries on the Product_master table created during Lab -3.

1. SELECT sell_price + tax AS total_price FROM product_master;
2. SELECT sell_price - tax AS cost FROM product_master;
3. SELECT cost_price* profit_percent AS new_price FROM product_master;
4. SELECT sell_price/profit_percent AS new_cost FROM producte_master;
5. SELECT POWER(columnname, 2) AS new_price from product_master;
6. SELECT (sell_price + tax) * quantity AS total FROM product_master;
7. SELECT (sell_price * quantity) + tax AS total FROM product_master WHERE product_no = P00001 ;

Rename Column Names:

Renaming columns used with Expression Lists: - The default output column names can be renamed by the user if required

(**AS** keyword is used to rename a columnname)

Syntax:

Select column name As Columnname From table name;

Query: SELECT sell_price AS new_price FROM product_master;

Logical Operators:

The logical operators that can be used in SQL sentences are

AND	all must be included
OR	any of them may be included,
NOT	none of them could be included

Range Searching: Between operations is used for range searching.

Query: SELECT * FROM product_master WHERE sell_price BETWEEN 2000 AND 5000;

Pattern Searching:

The most used operation on string is pattern matching using the operation 'like' we describe patterns by using two special characters.

- Percent (%); the % character matches any substring we consider in the following examples.
- 'Perry %' matches any string beginning with "Perry" (e.g., "Perry", "Perry Smith", "Perry123").
- '% idge % matches any string containing' "idge" anywhere (e.g., "bridge", "ridgeway", "fridge").
- '_' represents a single character.

Example:

- '_ _ _' matches any string that is exactly three characters long (e.g., "cat", "dog", "123").
- '- - - % ' matches any string of at least three characters.
- **SQL Queries:**

  - SELECT name FROM client_master WHERE name like 'I%' ;
  - SELECT name FROM client_master WHERE name like 'I_ _ _' ;
  - SELECT name FROM client_master WHERE name like '_an_ _ _ _' ;
  - SELECT name FROM client_master WHERE name like '%a%' ;

Oracle functions:

Functions are used to manipulate data items and return results. Functions follow the format of function _ name (argument1, argument2...). An arrangement is a user-defined variable or constant. The structure of a function is such that it accepts zero or more arguments.

Examples:

Average ():

- **Avg** return average value of n
- **Syntax:** Avg ([distinct/all]n)
- **Query:** SELECT Avg(sell_price) FROM product_master ;

Minimum ():

- **Min** return minimum value of expr/column.
- **Syntax:** MIN((distinct/all)expr)
- **Query:** SELECT Min(sell_price) FROM product_master ;

Count ():

- **Count** Returns the no of rows where expr is not null
- **Syntax:** Count ([distinct/all)expr]
- Count (*) Returns the no rows in the table, including duplicates and those with nulls.
- **Query:** SELECT Count(sell_price) FROM product_master ;

Maximum ():

- **Max** Return max value of the column
- Syntax: Max ([distinct/all]expr)
- **Query:** SELECT Max(sell_price) FROM product_master ;

Sum ():

- **Sum** Returns the sum of values of the column
- **Syntax:** Sum ([distinct/all]n)
- **Query:** SELECT Sum(sell_price) FROM product_master ;

Sorting of data in the table

Syntax:
Select columnname, columnname
From table
Order by columnname;
Example:

- **Query:** SELECT* FROM client_master order by name;

Assignment Task

Question.1 Using the table client master and product master answer the following Questionnaires.

i. Change the selling price of '1.44 floppy drive to Rs.1150.00
ii. Delete the record with client 0001 from the client master table.
iii. Change the city of client_no'0002' to Bombay.
iv. Change the bal_due of client_no '0001, to 1000.
v. Find the products whose selling price is more than 1500 and also find the new selling price as original selling price *15.
vi. Find out the clients who stay in a city whose second letter is a.
vii. Find out the name of all clients having 'a' as the second letter in their names.
viii. List the products in sorted order of their description.
ix. Count the total number of orders
x. Calculate the average price of all the products.
xi. Calculate the minimum price of products.
xii. Determine the maximum and minimum prices. Rename the tittle as 'max_price' and min_price respectively.
xiii. Count the number of products having price greater than or equal to 150.