

DATABASE MANAGEMENT SYSTEMS

LAB

ETCS-256



**Maharaja Agrasen Institute of Technology, PSP
area,
Sector – 22, Rohini, New Delhi – 110085
(Affiliated to Guru Gobind Singh Indraprastha
University, New Delhi)**

Submitted by- SAHIL ARYA
06214802719
4C4

INDEX

S.No.	DATE	EXPERIEMENT	REMARKS
1.	22/03/2021	To study E-R diagram considering two different scenarios (e.g., bank, college) and explain somebasic properties of E-R diagram.	
2.	12/04/2021	Creation of the databases/tables and insertion of data.	
3.	19/04/2021	Perform the queries for retrieving the appropriate data from the created tables using SELECT Command and WHERE Clause.	
4.	03/05/2021	Write SQL commands for implementing ALTER, UPDATE and DELETE.	
5.	17/05/2021	Write the queries to implement the concept of Integrity constraints like Primary Key, Foreign key, NOT NULL to the tables.	
6.	24/05/2021	To write the queries for implementing the following functions: MAX(), MIN(), AVG(), COUNT() and other logical and pattern matching operations.	

Experiment-1

Aim: Draw E-R diagram considering two different scenarios (e.g., bank, college) and explain some basic properties of E-R diagram.

Theory:

- What is ER Diagram?

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships. ER Diagram is a visual representation of data that describes how data is related to each other using different ERD Symbols and Notations.

Following are the main components and its symbols in ER Diagrams:

- Rectangles: This Entity Relationship Diagram symbol represents entity types
- Ellipses: Symbol represent attributes
- Diamonds: This symbol represents relationship types
- Lines: It links attributes to entity types and entity types with other relationship types
- Primary key: attributes are underlined
- Double Ellipses: Represent multi-valued attributes



- Components of the ER Diagram

This model is based on three basic concepts:

- Entities
- Attributes
- Relationships

WHAT IS ENTITY?

A real-world thing either living or non-living that is easily recognizable and nonrecognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world.

An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.

Examples of entities:

- Person: Employee, Student, Patient
- Place: Store, Building
- Object: Machine, product, and Car
- Event: Sale, Registration, Renewal
- Concept: Account, Course

Relationship

Relationship is nothing but an association among two or more entities. E.g., Tom works in the Chemistry department.



Entities take part in relationships. We can often identify relationships with verbs or verb phrases.

For example:

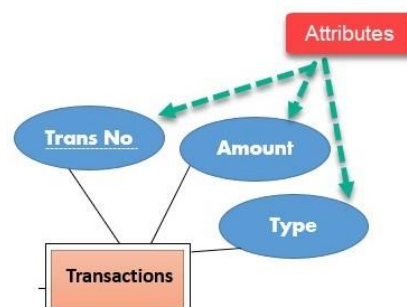
- You are attending this lecture
- I am giving the lecture
- Just like entities, we can classify relationships according to relationship-types:
- A student attends a lecture
- A lecturer is giving a lecture.

Attributes

It is a single-valued property of either an entity-type or a relationship-type.

For example, a lecture might have attributes: time, date, duration, place, etc.

An attribute in ER Diagram examples, is represented by an Ellipse



Types of Attributes	Description
Simple attribute	Simple attributes can't be divided any further. For example, a student's contact number. It is also called an atomic value.
Composite attribute	It is possible to break down composite attribute. For example, a student's full name

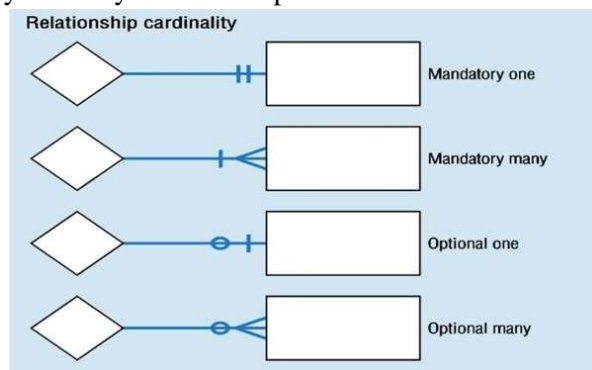
	may be further divided into first name, second name, and last name.
Derived attribute	This type of attribute does not include in the physical database. However, their values are derived from other attributes present in the database. For example, age should not be stored directly. Instead, it should be derived from the DOB of that employee.
Multivalued attribute	Multivalued attributes can have more than one values. For example, a student can have more than one mobile number, email address, etc.

Cardinality

Defines the numerical attributes of the relationship between two entities or entity sets.

Different types of cardinal relationships are:

- One-to-One Relationships
- One-to-Many Relationships
- May to One Relationships
- Many-to-Many Relationships



1. One-to-one:

One entity from entity set X can be associated with at most one entity of entity set Y and vice versa.

Example: One student can register for numerous courses. However, all those courses have a single line back to that one student.

2. One-to-many:

One entity from entity set X can be associated with multiple entities of entity set Y, but an entity from entity set Y can be associated with at least one entity.

For example, one class is consisting of multiple students.

3. Many to One:

More than one entity from entity set X can be associated with at most one entity of entity set Y. However, an entity from entity set Y may or may not be associated with more than one entity from entity set X.

For example, many students belong to the same class.

4. Many to Many:

One entity from X can be associated with more than one entity from Y and vice versa.

For example, Students as a group are associated with multiple faculty members, and faculty members can be associated with multiple students.

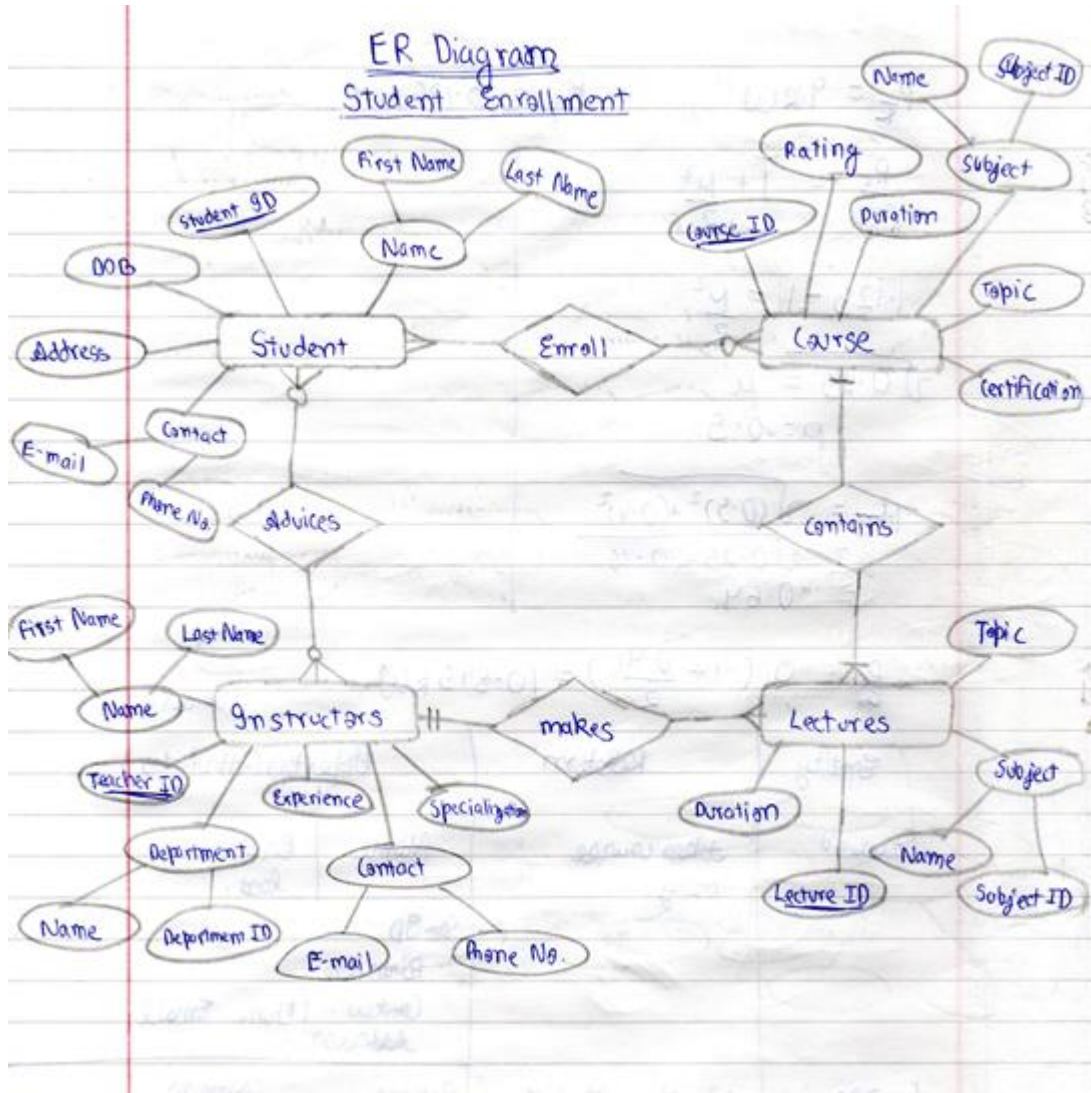
- Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

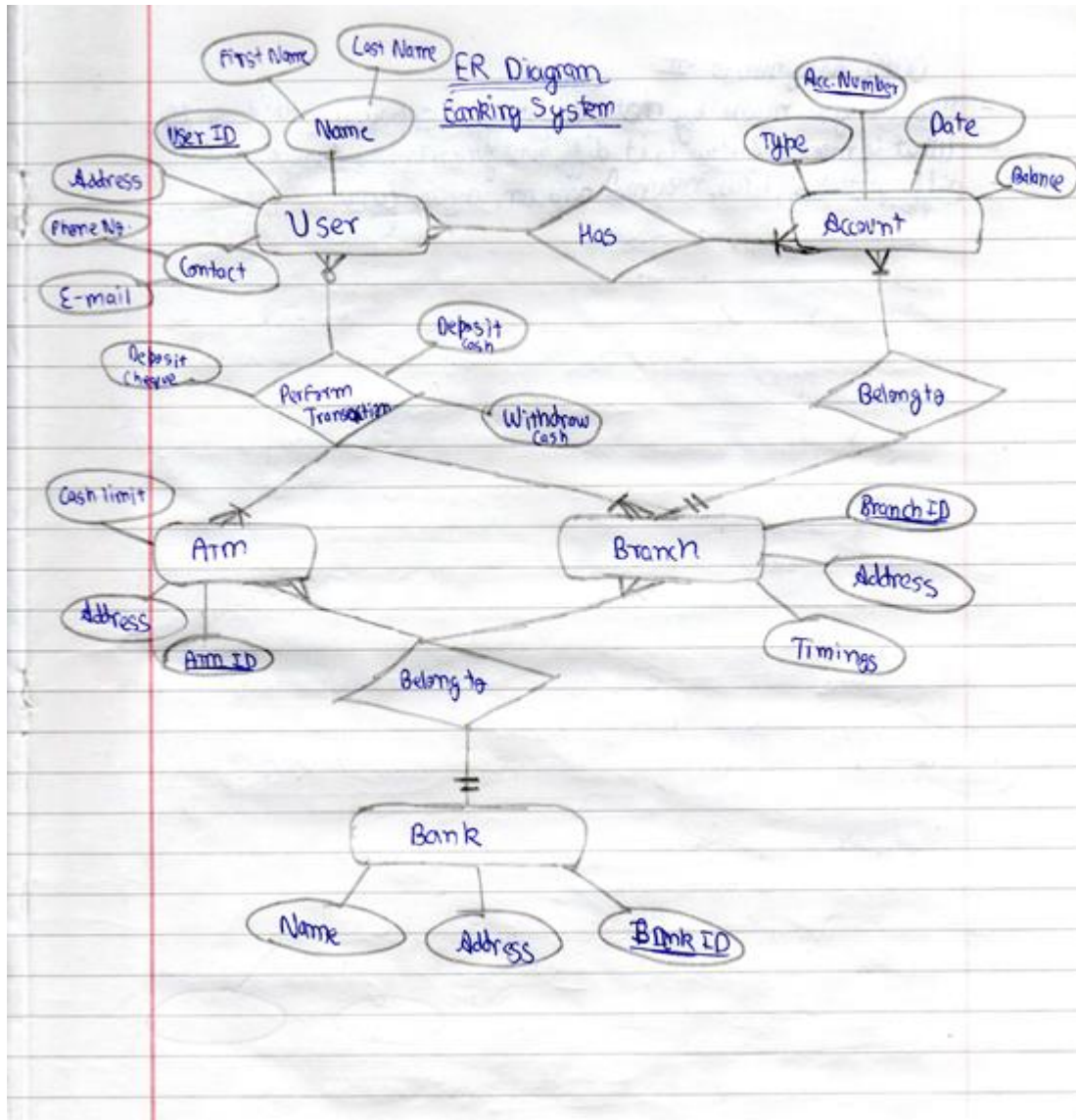
- Helps you to define terms related to entity relationship modelling
- Provide a preview of how all your tables should connect, what fields are going to be on each table
- Helps to describe entities, attributes, relationships
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
- ERD Diagram allows you to communicate with the logical structure of the database to users

Output:

- E-R Diagram describing student's course enrolment.



- E-R diagram describing banking system.



Experiment - 2

AIM: Creation of the databases/tables and insertion of data.

TOOL USED: MariaDB (MySQL Client)

THEORY: Various commands of SQL used in this experiment are:

- I. **Create command:** The CREATE TABLE statement is used to create a new table in a database.

Syntax:

```
CREATE TABLE table name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ...  
);
```

- II. **Insert command:** The INSERT INTO statement is used to insert new records in a table.

Syntax:

```
INSERT INTO table name(column1, column2, column3, ...)  
VALUES(value1, value2, value3,...);
```

QUERIES:

PART-1: CREATION OF TABLES

(i)TABLE NAME: CLIENT MASTER

```
create table CLIENT_MASTER(  
    Client_No varchar(6),  
    Name varchar(20),  
    Address1 varchar(30),  
    Address2 varchar(30),  
    City varchar(15),  
    Pincode int(8),
```

```
State varchar(15),
Bal_Due float(10,2));
```

```
MariaDB [dbms]> create table CLIENT_MASTER(
  -> Client_No varchar(6),
  -> Name varchar(20),
  -> Address1 varchar(30),
  -> Address2 varchar(30),
  -> City varchar(15),
  -> Pincode int(8),
  -> State varchar(15),
  -> Bal_Due float(10,2));
Query OK, 0 rows affected (0.021 sec)
```

```
MariaDB [dbms]> describe CLIENT_MASTER;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Client_No  | varchar(6)    | YES  |     | NULL    |       |
| Name       | varchar(20)   | YES  |     | NULL    |       |
| Address1   | varchar(30)   | YES  |     | NULL    |       |
| Address2   | varchar(30)   | YES  |     | NULL    |       |
| City       | varchar(15)   | YES  |     | NULL    |       |
| Pincode    | int(8)        | YES  |     | NULL    |       |
| State      | varchar(15)   | YES  |     | NULL    |       |
| Bal_Due    | float(10,2)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.020 sec)
```

(ii)TABLE NAME : PRODUCT MASTER

```
create table PRODUCT_MASTER(
Product_No varchar(6),
Description varchar(15),
Profit_Percent float(4,2),
Unit_Measure varchar(10),
Qty_On_Hand int(8),
Reorder_Lvl int(8),
Sell_Price float(8,2),
Cost_Price float(8,2));
```

```
MariaDB [dbms]> create table PRODUCT_MASTER(  
  -> Product_No varchar(6),  
  -> Description varchar(15),  
  -> Profit_Percent float(4,2),  
  -> Unit_Measure varchar(10),  
  -> Qty_On_Hand int(8),  
  -> Reorder_Lvl int(8),  
  -> Sell_Price float(8,2),  
  -> Cost_Price float(8,2));  
Query OK, 0 rows affected (0.013 sec)
```

```
MariaDB [dbms]> describe PRODUCT_MASTER;  
+-----+-----+-----+-----+-----+-----+  
| Field          | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| Product_No     | varchar(6)    | YES  |     | NULL    |       |  
| Description     | varchar(15)   | YES  |     | NULL    |       |  
| Profit_Percent | float(4,2)    | YES  |     | NULL    |       |  
| Unit_Measure   | varchar(10)   | YES  |     | NULL    |       |  
| Qty_On_Hand    | int(8)        | YES  |     | NULL    |       |  
| Reorder_Lvl    | int(8)        | YES  |     | NULL    |       |  
| sell_price     | float(10,2)   | YES  |     | NULL    |       |  
| Cost_Price     | float(8,2)    | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
8 rows in set (0.022 sec)
```

(iii)TABLE NAME: SALESMAN MASTER

```
create table SALESMAN_MASTER(  
  
Salesman_No varchar(6),  
  
Salesman_Name varchar(20),  
  
Address1 varchar(30),  
  
Address2 varchar(30),  
  
City varchar(20),  
  
Pincode int(8),  
  
State varchar(20),  
  
Sal_Amt int,  
  
Tgt_To_Get int,  
  
Ytd_Sales int,  
  
Remarks varchar(10));
```

```

MariaDB [dbms]> create table SALESMAN_MASTER(
  -> Salesman_No varchar(6),
  -> Salesman_Name varchar(20),
  -> Address1 varchar(30),
  -> Address2 varchar(30),
  -> City varchar(20),
  -> Pincode int(8),
  -> State varchar(20),
  -> Sal_Amt int,
  -> Tgt_To_Get int,
  -> Ytd_Sales int,
  -> Remarks varchar(10));
Query OK, 0 rows affected (0.019 sec)

```

```

MariaDB [dbms]> describe SALESMAN_MASTER;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Salesman_No    | varchar(6)    | YES  |     | NULL    |       |
| Salesman_Name  | varchar(20)   | YES  |     | NULL    |       |
| Address1       | varchar(30)   | YES  |     | NULL    |       |
| Address2       | varchar(30)   | YES  |     | NULL    |       |
| City           | varchar(20)   | YES  |     | NULL    |       |
| Pincode        | int(8)        | YES  |     | NULL    |       |
| State          | varchar(20)   | YES  |     | NULL    |       |
| Sal_Amt        | int(11)       | YES  |     | NULL    |       |
| Tgt_To_Get     | int(11)       | YES  |     | NULL    |       |
| Ytd_Sales      | int(11)       | YES  |     | NULL    |       |
| Remarks        | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.020 sec)

```

PART-2: INSERTION OF VALUES INTO TABLES

DATA FOR 'CLIENT_MASTER' TABLE:

insert into CLIENT_MASTER

values('C00001','Ivan',' ',' ','Mumbai',400054,'Maharashtra',15000);

```

MariaDB [dbms]> insert into CLIENT_MASTER values('C00001','Ivan',' ',' ','Mumbai',400054,'Maharashtra',15000);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> select * from CLIENT_MASTER;
+-----+-----+-----+-----+-----+-----+-----+
| Client_No | Name  | Address1 | Address2 | City   | Pincode | State       | Bal_Due |
+-----+-----+-----+-----+-----+-----+-----+
| C00001    | Ivan |          |          | Mumbai | 400054 | Maharashtra | 15000.00 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

```

insert into CLIENT_MASTER

values('C00002','Mamta Muzumdar',' ',' ','Madras',780001,'Tamil Nadu',0);

```
MariaDB [dbms]> insert into CLIENT_MASTER values('C00002','Mamta Muzumdar',' ',' ','Madras',780001,'Tamil Nadu',0);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from CLIENT_MASTER;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Client_No | Name       | Address1 | Address2 | City   | Pincode | State   | Bal_Due |
+-----+-----+-----+-----+-----+-----+-----+-----+
| C00001    | Ivan      |          |          | Mumbai | 400054 | Maharashtra | 15000.00 |
| C00002    | Mamta Muzumdar |          |          | Madras | 780001 | Tamil Nadu | 0.00    |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Insert into CLIENT_MASTER

values('C00003','ChhayaBankar',' ',' ','Mumbai',400057,'Maharashtra',5000);

```
MariaDB [dbms]> insert into CLIENT_MASTER values('C00003','Chhaya Bankar',' ',' ','Mumbai',400057,'Maharashtra',5000);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from CLIENT_MASTER;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Client_No | Name       | Address1 | Address2 | City   | Pincode | State   | Bal_Due |
+-----+-----+-----+-----+-----+-----+-----+-----+
| C00001    | Ivan      |          |          | Mumbai | 400054 | Maharashtra | 15000.00 |
| C00002    | Mamta Muzumdar |          |          | Madras | 780001 | Tamil Nadu | 0.00    |
| C00003    | Chhaya Bankar |          |          | Mumbai | 400057 | Maharashtra | 5000.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

insert into CLIENT_MASTER

values('C00004','Ashwini Joshi',' ',' ','Banglore',560001,'Karnataka',0);

```
MariaDB [dbms]> insert into CLIENT_MASTER values('C00004','Ashwini Joshi',' ',' ','Banglore',560001,'Karnataka',0);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from CLIENT_MASTER;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Client_No | Name       | Address1 | Address2 | City   | Pincode | State   | Bal_Due |
+-----+-----+-----+-----+-----+-----+-----+-----+
| C00001    | Ivan      |          |          | Mumbai | 400054 | Maharashtra | 15000.00 |
| C00002    | Mamta Muzumdar |          |          | Madras | 780001 | Tamil Nadu | 0.00    |
| C00003    | Chhaya Bankar |          |          | Mumbai | 400057 | Maharashtra | 5000.00 |
| C00004    | Ashwini Joshi |          |          | Banglore | 560001 | Karnataka | 0.00    |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

insert into CLIENT_MASTER

values('C00005','Hansel Colaco',' ',' ','Mumbai',400060,'Maharashtra',2000);

```
MariaDB [dbms]> insert into CLIENT_MASTER values('C00005','Hansel Colaco','','','Mumbai',400060,'Maharashtra',2000);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> select * from CLIENT_MASTER;
```

Client_No	Name	Address1	Address2	City	Pincode	State	Bal_Due
C00001	Ivan			Mumbai	400054	Maharashtra	15000.00
C00002	Mamta Muzumdar			Madras	780001	Tamil Nadu	0.00
C00003	Chhaya Bankar			Mumbai	400057	Maharashtra	5000.00
C00004	Ashwini Joshi			Banglore	560001	Karnataka	0.00
C00005	Hansel Colaco			Mumbai	400060	Maharashtra	2000.00

```
5 rows in set (0.001 sec)
```

insert into CLIENT_MASTER

values('C00006','Deepak Sharma','','','Mangalore',560050,'Karnataka',0);

```
MariaDB [dbms]> insert into CLIENT_MASTER values('C00006','Deepak Sharma','','','Mangalore',560050,'Karnataka',0);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> select * from CLIENT_MASTER;
```

Client_No	Name	Address1	Address2	City	Pincode	State	Bal_Due
C00001	Ivan			Mumbai	400054	Maharashtra	15000.00
C00002	Mamta Muzumdar			Madras	780001	Tamil Nadu	0.00
C00003	Chhaya Bankar			Mumbai	400057	Maharashtra	5000.00
C00004	Ashwini Joshi			Banglore	560001	Karnataka	0.00
C00005	Hansel Colaco			Mumbai	400060	Maharashtra	2000.00
C00006	Deepak Sharma			Mangalore	560050	Karnataka	0.00

```
6 rows in set (0.001 sec)
```

DATA FOR 'PRODUCT_MASTER' TABLE:

insert into PRODUCT_MASTER values('P00001','T-Shirts',5,'Piece',200,50,350,250);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P00001','T-Shirts',5,'Piece',200,50,350,250);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00

```
1 row in set (0.001 sec)
```

insert into PRODUCT_MASTER values('P0345','Shirts',6,'Piece',150,50,500,350);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P0345','Shirts',6,'Piece',150,50,500,350);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00

```
2 rows in set (0.000 sec)
```

insert into PRODUCT_MASTER values('P06734','Cotton Jeans',5,'Piece',100,20,600,450);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P06734','Cotton Jeans',5,'Piece',100,20,600,450);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00

3 rows in set (0.000 sec)

insert into PRODUCT_MASTER values('P07865','Jeans',5,'Piece',100,20,750,500);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P07865','Jeans',5,'Piece',100,20,750,500);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00

4 rows in set (0.000 sec)

insert into PRODUCT_MASTER values('P07868','Trousers',2,'Piece',150,50,850,550);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P07868','Trousers',2,'Piece',150,50,850,550);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00
P07868	Trousers	2.00	Piece	150	50	850.00	550.00

5 rows in set (0.000 sec)

insert into PRODUCT_MASTER values('P07885','Pull Overs',2.5,'Piece',80,30,700,450);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P07885','Pull Overs',2.5,'Piece',80,30,700,450);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00
P07868	Trousers	2.00	Piece	150	50	850.00	550.00
P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00

6 rows in set (0.000 sec)

insert into PRODUCT_MASTER values('P07965','Denim Shirts',4,'Piece',100,40,350,250);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P07965','Denim Shirts',4,'Piece',100,40,350,250);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00
P07868	Trousers	2.00	Piece	150	50	850.00	550.00
P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
P07965	Denim Shirts	4.00	Piece	100	40	350.00	250.00

7 rows in set (0.000 sec)

insert into PRODUCT_MASTER values('P07975','Lycra Tops',5,'Piece',70,30,300,175);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P07975','Lycra Tops',5,'Piece',70,30,300,175);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00
P07868	Trousers	2.00	Piece	150	50	850.00	550.00
P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
P07965	Denim Shirts	4.00	Piece	100	40	350.00	250.00
P07975	Lycra Tops	5.00	Piece	70	30	300.00	175.00

8 rows in set (0.001 sec)

insert into PRODUCT_MASTER values('P08865','Skirts',5,'Piece',75,30,450,300);

```
MariaDB [dbms]> insert into PRODUCT_MASTER values('P08865','Skirts',5,'Piece',75,30,450,300);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
```

Product_No	Description	Profit_Percent	Unit_Measure	Qty_On_Hand	Reorder_Lvl	Sell_Price	Cost_Price
P00001	T-Shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00
P07868	Trousers	2.00	Piece	150	50	850.00	550.00
P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
P07965	Denim Shirts	4.00	Piece	100	40	350.00	250.00
P07975	Lycra Tops	5.00	Piece	70	30	300.00	175.00
P08865	Skirts	5.00	Piece	75	30	450.00	300.00

9 rows in set (0.000 sec)

insert into SALESMAN_MASTER

```
MariaDB [dbsms]> insert into SALESMAN_MASTER values('S00001','Aman','A/14','Worli','Mumbai',400002,'Maharashtra',3000,100,50,'Good');
Query OK, 1 row affected (0.002 sec)
```

Salesman_No	Salesman_Name	Address1	Address2	City	Pincode	State	Sal_Amt	Tgt_To_Get	Ytd_Sales	Remarks
500001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000	100	50	Good

1 row in set (0.001 sec)

```
values('S00002','Omkar','65','Nariman','Mumbai',400001,'Maharashtra',3000,200,100,'Good');
```

Salesman_No	Salesman_Name	Address1	Address2	City	Pincode	State	Sal_Amt	Tgt_To_Get	Ytd_Sales	Remarks
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000	100	50	Good
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra	3000	200	100	Good

2 rows in set (0.001 sec)

```
values('S00003','Raj','P-7','Bandra','Mumbai',400032,'Maharashtra',3000,200,100,'Good');
```

Salesman_No	Salesman_Name	Address1	Address2	City	Pincode	State	Sal_Amt	Tgt_To_Get	Ytd_Sales	Remarks
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000	100	50	Good
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra	3000	200	100	Good
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra	3000	200	100	Good

3 rows in set (0.001 sec)

```
values('S00004','Ashish','A/5','Juhu','Mumbai',400044,'Maharashtra',3000,200,150,'Good');
```

Salesman_No	Salesman_Name	Address1	Address2	City	Pincode	State	Sal_Amt	Tgt_To_Get	Ytd_Sales	Remarks
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000	100	50	Good
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra	3000	200	100	Good
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra	3000	200	100	Good
S00004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtra	3000	200	150	Good

4 rows in set (0.001 sec)

EXPERIMENT-3

AIM: Perform the queries for retrieving the appropriate data from the created tables using **SELECT** Command and **WHERE** Clause.

TOOL USED : Maria DB

THEORY:

Data Query Language (DQL) is part of the base grouping of SQL sub-languages. DQL statements are used for performing queries on the data within schema objects. The purpose of DQL commands is to get the schema relation based on the query passed to it. The SQL **SELECT** statement is an example of DQL.

Commands used in this experiment are:

- **SELECT** – It is used to retrieve data from the database.

Syntax:

SELECT *column1, column2*

FROM *table name*

Queries:

1.Find out the names of all clients.

select name from CLIENT_MASTER;

```
MariaDB [(none)]> use dbms
Database changed
MariaDB [dbms]> select name from CLIENT_MASTER;
+-----+
| name |
+-----+
| Ivan |
| Mamta Muzumdar |
| Chhaya Bankar |
| Ashwini Joshi |
| Hansel Colaco |
| Deepak Sharma |
+-----+
6 rows in set (0.001 sec)
```

2.Retrieve the entire contents of client_master table.

select * from CLIENT_MASTER;

```
MariaDB [dbms]> select * from CLIENT_MASTER;
```

Client_No	Name	Address1	Address2	City	Pincode	State	Bal_Due
C00001	Ivan			Mumbai	400054	Maharashtra	15000.00
C00002	Mamta Muzumdar			Madras	780001	Tamil Nadu	0.00
C00003	Chhaya Bankar			Mumbai	400057	Maharashtra	5000.00
C00004	Ashwini Joshi			Banglore	560001	Karnataka	0.00
C00005	Hansel Colaco			Mumbai	400060	Maharashtra	2000.00
C00006	Deepak Sharma			Mangalore	560050	Karnataka	0.00

6 rows in set (0.001 sec)

3.Retrieve the list of names, cities and the states of all clients.

select name, city, state from CLIENT_MASTER;

```
MariaDB [dbms]> select name, city, state from CLIENT_MASTER;
```

name	city	state
Ivan	Mumbai	Maharashtra
Mamta Muzumdar	Madras	Tamil Nadu
Chhaya Bankar	Mumbai	Maharashtra
Ashwini Joshi	Banglore	Karnataka
Hansel Colaco	Mumbai	Maharashtra
Deepak Sharma	Mangalore	Karnataka

6 rows in set (0.001 sec)

4.Find the names of clients from client_master where bal_due is 0.

select name from CLIENT_MASTER where bal_due=0;

```
MariaDB [dbms]> select name from CLIENT_MASTER where bal_due=0;
```

name
Mamta Muzumdar
Ashwini Joshi
Deepak Sharma

3 rows in set (0.006 sec)

5. List all the clients who are located in Mumbai.

select name from CLIENT_MASTER where city='Mumbai';

```
MariaDB [dbms]> select name from CLIENT_MASTER where city='Mumbai';
+-----+
| name |
+-----+
| Ivan |
| Chhaya Bankar |
| Hansel Colaco |
+-----+
3 rows in set (0.003 sec)
```

6. List the various products available from product_master table.

select description from PRODUCT_MASTER;

```
MariaDB [dbms]> select description from PRODUCT_MASTER;
+-----+
| description |
+-----+
| T-Shirts |
| Shirts |
| Trousers |
| Pull Overs |
| Lycra Tops |
| Skirts |
+-----+
6 rows in set (0.022 sec)
```

7. Find all the names of salesmen who have a salary equal to rupees 3000.

select salesman_name from SALESMAN_MASTER where sal_amt=3000;

```
MariaDB [dbms]> select salesman_name from SALESMAN_MASTER where sal_amt=3000;
+-----+
| salesman_name |
+-----+
| Aman |
| Omkar |
| Raj |
| Ashish |
+-----+
4 rows in set (0.001 sec)
```

8. List all salesman number who live in city Mumbai.

select salesman_no from SALESMAN_MASTER where city='Mumbai';

```
MariaDB [dbms]> select salesman_no from SALESMAN_MASTER where city='Mumbai';
+-----+
| salesman_no |
+-----+
| S00001 |
| S00002 |
| S00003 |
| S00004 |
+-----+
4 rows in set (0.001 sec)
```

9. Find the product number whose price is equal to 150 rupees.

select product_no from PRODUCT_MASTER where sell_price=150;

```
MariaDB [dbms]> select product_no from PRODUCT_MASTER where sell_price=150;
Empty set (0.001 sec)
```

10. Find product_no of t-shirts in product_master table.

select product_no from PRODUCT_MASTER where description='T-Shirts';

```
MariaDB [dbms]> select product_no from PRODUCT_MASTER where description='T-Shirts';
+-----+
| product_no |
+-----+
| P00001     |
+-----+
1 row in set (0.001 sec)
```

EXPERIMENT-4

AIM: Write SQL commands for implementing ALTER, UPDATE and DELETE.

TOOL USED: MariaDB

THEORY:

DDL(Data Definition Language) : DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

DDL commands used in this experiment are:

DROP – is used to delete objects from the database.

Syntax:

DROP TABLE *table_name*;

ALTER-is used to alter the structure of the database.

Syntax:

ALTER TABLE *table_name*

ADD (*Column_name datatype*);

RENAME –is used to rename an object existing in the database.

Syntax:

ALTER TABLE *table_name*

RENAME TO *new_table_name*;

DML(Data Manipulation Language): The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. DML commands used in this experiment are:

UPDATE – is used to update existing data within a table.

Syntax:

UPDATE *table_name* SET *column1* = *value1*, *column2* = *value2*,
WHERE *condition*;

DELETE – is used to delete records from a database table.

Syntax:

DELETE FROM table_name WHERE condition;

QUERIES:

1. Change the city of Client_no 'C00005' to 'Bangalore'.

update table client_master

set city='Bangalore' where client_no='C00005';

ca MySQL Client (MariaDB 10.5 (x64)) - "C:\Program Files\MariaDB 10.5\bin\mysql.exe" "--defaults-file=C:\Program Files\MariaDB 10.5\data\my.ini" -uroot -p

```
MariaDB [lab_1]> update client_master
-> set city='Bangalore' where client_no='C00005';
Query OK, 1 row affected (0.009 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [lab_1]> select* from client_master;
```

client_no	name	address1	address2	city	pincode	state	bal_due
C00001	Ivan	A/21	maurya nagar bandra	Mumbai	40054	Maharashtra	15000.00
C00002	Mamta Mazumdar	B-14	Anna Enclave	Madras	780001	Tamil Nadu	1200.00
C00003	Chhaya Bankar	C-59	Ambedkar Enclave,Dadar	Mumbai	400057	Mahrashtra	5000.00
C00004	Ashwini Joshi	D-22	Cybertech city	Bangalore	560001	Karnataka	0.00
C00005	Hansel Colaco	H-20	Gokuldham Society,Goregaon	Bangalore	400060	Maharashtra	2000.00
C00006	Deepak Sharma	F/69	Model Town phase-1	Mangalore	560050	Karnataka	0.00

6 rows in set (0.001 sec)

2. Change the Bal_due of client_no 'C00001' to Rs.1000.

update

client_master set

bal_due=1000

where client_no='C00001';

```
MariaDB [lab_1]> select* from client_master;
```

client_no	name	address1	address2	city	pincode	state	bal_due
C00001	Ivan	A/21	maurya nagar bandra	Mumbai	40054	Maharashtra	15000.00
C00002	Mamta Mazumdar	B-14	Anna Enclave	Madras	780001	Tamil Nadu	1200.00
C00003	Chhaya Bankar	C-59	Ambedkar Enclave,Dadar	Mumbai	400057	Mahrashtra	5000.00
C00004	Ashwini Joshi	D-22	Cybertech city	Bangalore	560001	Karnataka	0.00
C00005	Hansel Colaco	H-20	Gokuldharm Society,Goregaon	Bangalore	400060	Maharashtra	2000.00
C00006	Deepak Sharma	F/69	Model Town phase-1	Mangalore	560050	Karnataka	0.00

```
6 rows in set (0.001 sec)
```

```
MariaDB [lab_1]> update client_master
```

```
-> set bal_due=1000
```

```
-> where client_no='C00001';
```

```
Query OK, 1 row affected (0.002 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [lab_1]> select* from client_master;
```

client_no	name	address1	address2	city	pincode	state	bal_due
C00001	Ivan	A/21	maurya nagar bandra	Mumbai	40054	Maharashtra	1000.00
C00002	Mamta Mazumdar	B-14	Anna Enclave	Madras	780001	Tamil Nadu	1200.00
C00003	Chhaya Bankar	C-59	Ambedkar Enclave,Dadar	Mumbai	400057	Mahrashtra	5000.00
C00004	Ashwini Joshi	D-22	Cybertech city	Bangalore	560001	Karnataka	0.00
C00005	Hansel Colaco	H-20	Gokuldharm Society,Goregaon	Bangalore	400060	Maharashtra	2000.00
C00006	Deepak Sharma	F/69	Model Town phase-1	Mangalore	560050	Karnataka	0.00

```
6 rows in set (0.001 sec)
```


3.Change the cost price of 'Trousers' to Rs.950.

update

product_master set

costprice=950

where description='Trousers';

```
MariaDB [lab_1]> select* from product_master;
```

PRODUCT_No	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDER_VL	SELLPRICE	COSTPRICE
P00001	T-shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00
P07868	Trousers	2.00	Piece	150	50	850.00	550.00
P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
P07965	Denim Shirts	4.00	Piece	100	40	350.00	250.00
P07975	Lycra Tops	5.00	Piece	70	30	300.00	175.00
P08865	Skirts	5.00	Piece	75	30	450.00	300.00

9 rows in set (0.001 sec)

```
MariaDB [lab_1]> update product_master
```

```
-> set costprice=950
```

```
-> where description='Trousers';
```

```
Query OK, 1 row affected (0.003 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [lab_1]> select* from product_master;
```

PRODUCT_No	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDER_VL	SELLPRICE	COSTPRICE
P00001	T-shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06734	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00
P07868	Trousers	2.00	Piece	150	50	850.00	950.00
P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
P07965	Denim Shirts	4.00	Piece	100	40	350.00	250.00
P07975	Lycra Tops	5.00	Piece	70	30	300.00	175.00
P08865	Skirts	5.00	Piece	75	30	450.00	300.00

9 rows in set (0.002 sec)

4.Change the city of salesman name to 'Pune'.

update

salesman_master set

city='Pune';

MySQL Client (MariaDB 10.5 (x64)) - "C:\Program Files\MariaDB 10.5\bin\mysql.exe" "--defaults-file=C:\Program Files\MariaDB 10.5\data\my.ini" -uroot -p

MariaDB [lab_1]> select* from salesman_master;

salesmanno	salesmanname	address1	address2	city	pincode	state	salamt	TgtToGet	YtdSales	Remarks
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000.00	100	50	good
S00002	Omkar	B-5	Nariman	Mumbai	400001	Maharashtra	3000.00	200	100	good
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra	3000.00	200	100	good
S00004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtra	3500.00	200	150	good

4 rows in set (0.001 sec)

MariaDB [lab_1]> update salesman_master

-> set city='Pune';

Query OK, 4 rows affected (0.006 sec)

Rows matched: 4 Changed: 4 Warnings: 0

MariaDB [lab_1]> select* from salesman_master;

salesmanno	salesmanname	address1	address2	city	pincode	state	salamt	TgtToGet	YtdSales	Remarks
S00001	Aman	A/14	Worli	Pune	400002	Maharashtra	3000.00	100	50	good
S00002	Omkar	B-5	Nariman	Pune	400001	Maharashtra	3000.00	200	100	good
S00003	Raj	P-7	Bandra	Pune	400032	Maharashtra	3000.00	200	100	good
S00004	Ashish	A/5	Juhu	Pune	400044	Maharashtra	3500.00	200	150	good

4 rows in set (0.001 sec)

5.Delete all salesman from the salesman_master whose salaries are equal to 3500.

delete from salesman_master where salamt=3500;

```
MariaDB [lab_1]> select* from salesman_master;
```

salesmanno	salesmanname	address1	address2	city	pincode	state	salamt	TgtToGet	YtdSales	Remarks
S00001	Aman	A/14	Worli	Pune	400002	Maharashtra	3000.00	100	50	good
S00002	Omkar	B-5	Nariman	Pune	400001	Maharashtra	3000.00	200	100	good
S00003	Raj	P-7	Bandra	Pune	400032	Maharashtra	3000.00	200	100	good
S00004	Ashish	A/5	Juhu	Pune	400044	Maharashtra	3500.00	200	150	good

```
4 rows in set (0.001 sec)

MariaDB [lab_1]> delete from salesman_master
-> where salamt=3500;
Query OK, 1 row affected (0.002 sec)

MariaDB [lab_1]> select* from salesman_master;
```

salesmanno	salesmanname	address1	address2	city	pincode	state	salamt	TgtToGet	YtdSales	Remarks
S00001	Aman	A/14	Worli	Pune	400002	Maharashtra	3000.00	100	50	good
S00002	Omkar	B-5	Nariman	Pune	400001	Maharashtra	3000.00	200	100	good
S00003	Raj	P-7	Bandra	Pune	400032	Maharashtra	3000.00	200	100	good

```
3 rows in set (0.001 sec)
```

6. Delete all products from product_master where QtyOnHand=100.

delete from product_master where qtyonhand=100;

```
MySQL Client (MariaDB 10.5 (x64)) - "C:\Program Files\MariaDB 10.5\bin\mysql.exe" "--defaults-file=C:\Program Files\MariaDB 10.5\data\my.ini" -uroot -p
Query OK, 1 row affected (0.003 sec)

MariaDB [lab_1]> select* from product_master;
```

PRODUCT_No	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDERLVL	sellprice	COSTPRICE
P00001	T-shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P06374	Cotton Jeans	5.00	Piece	100	20	600.00	450.00
P07865	Jeans	5.00	Piece	100	20	750.00	500.00
P07868	Trousers	2.00	Piece	150	50	850.00	550.00
P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
P07965	Denim Shirts	4.00	Piece	100	40	350.00	250.00
P07975	Lycra Tops	5.00	Piece	70	30	300.00	175.00
P08865	Skirts	5.00	Piece	75	30	450.00	300.00

```
9 rows in set (0.002 sec)

MariaDB [lab_1]> delete from product_master where qtyonhand=100;
Query OK, 3 rows affected (0.002 sec)

MariaDB [lab_1]> select* from product_master;
```

PRODUCT_No	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDERLVL	sellprice	COSTPRICE
P00001	T-shirts	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P07868	Trousers	2.00	Piece	150	50	850.00	550.00
P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
P07975	Lycra Tops	5.00	Piece	70	30	300.00	175.00
P08865	Skirts	5.00	Piece	75	30	450.00	300.00

```
6 rows in set (0.002 sec)
```

7.Delete from client_master where the column state holds the value='Tamil Nadu'.

delete from
client_master where

ca MySQL Client (MariaDB 10.5 (x64)) - "C:\Program Files\MariaDB 10.5\bin\mysql.exe" "--defaults-file=C:\Program Files\MariaDB 10.5\data\my.ini" -uroot -p

```
-> \c
MariaDB [lab_1]> select* from client_master;
```

client_no	name	address1	address2	city	pincode	state	bal_due
C00001	Ivan	A/21	maurya nagar bandra	Mumbai	40054	Maharashtra	1000.00
C00002	Mamta Mazumdar	B-14	Anna Enclave	Madras	780001	Tamil Nadu	1200.00
C00003	Chhaya Bankar	C-59	Ambedkar Enclave,Dadar	Mumbai	400057	Mahrashtra	5000.00
C00004	Ashwini Joshi	D-22	Cybertech city	Bangalore	560001	Karnataka	0.00
C00005	Hansel Colaco	H-20	Gokuldharm Society,Goregaon	Bangalore	400060	Maharashtra	2000.00
C00006	Deepak Sharma	F/69	Model Town phase-1	Mangalore	560050	Karnataka	0.00

```
6 rows in set (0.001 sec)

MariaDB [lab_1]> delete from client_master
-> where state='Tamil Nadu';
Query OK, 1 row affected (0.003 sec)

MariaDB [lab_1]> select* from client_master;
```

client_no	name	address1	address2	city	pincode	state	bal_due
C00001	Ivan	A/21	maurya nagar bandra	Mumbai	40054	Maharashtra	1000.00
C00003	Chhaya Bankar	C-59	Ambedkar Enclave,Dadar	Mumbai	400057	Mahrashtra	5000.00
C00004	Ashwini Joshi	D-22	Cybertech city	Bangalore	560001	Karnataka	0.00
C00005	Hansel Colaco	H-20	Gokuldharm Society,Goregaon	Bangalore	400060	Maharashtra	2000.00
C00006	Deepak Sharma	F/69	Model Town phase-1	Mangalore	560050	Karnataka	0.00

```
5 rows in set (0.049 sec)
```

state='Tamil Nadu';

8.Add a column called 'Telephone' of datatype 'Numeric' and size=10 to client_master table.

```
alter table client_master
add telephone
numeric(10);
```

MySQL Client (MariaDB 10.5 (x64)) - "C:\Program Files\MariaDB 10.5\bin\mysql.exe" "--defaults-file=C:\Program Files\MariaDB 10.5\data\my.ini" -uroot -p

```
MariaDB [lab_1]> describe client_master;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| client_no | char(6) | YES | | NULL | |
| name | varchar(20) | YES | | NULL | |
| address1 | varchar(30) | YES | | NULL | |
| address2 | varchar(30) | YES | | NULL | |
| city | varchar(15) | YES | | NULL | |
| pincode | decimal(8,0) | YES | | NULL | |
| state | varchar(15) | YES | | NULL | |
| bal_due | decimal(10,2) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.009 sec)

MariaDB [lab_1]> alter table client_master
-> add telephone numeric(10);
Query OK, 0 rows affected (0.009 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [lab_1]> describe client_master;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| client_no | char(6) | YES | | NULL | |
| name | varchar(20) | YES | | NULL | |
| address1 | varchar(30) | YES | | NULL | |
| address2 | varchar(30) | YES | | NULL | |
| city | varchar(15) | YES | | NULL | |
| pincode | decimal(8,0) | YES | | NULL | |
| state | varchar(15) | YES | | NULL | |
| bal_due | decimal(10,2) | YES | | NULL | |
| telephone | decimal(10,0) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.014 sec)
```

9. Change the size of 'Sellprice' column to '(10,2)' in product_master table.

```
alter table product_master
```

```
modify column sellprice numeric(10,2);
```

```
MariaDB [lab_1]> describe product_master;
```

Field	Type	Null	Key	Default	Extra
PRODUCT_No	char(6)	YES		NULL	
DESCRIPTION	varchar(20)	YES		NULL	
PROFITPERCENT	decimal(4,2)	YES		NULL	
UNITMEASURE	varchar(10)	YES		NULL	
QTYONHAND	decimal(8,0)	YES		NULL	
REORDERLVL	decimal(8,0)	YES		NULL	
sellprice	decimal(8,2)	YES		NULL	
COSTPRICE	decimal(8,2)	YES		NULL	

```
8 rows in set (0.009 sec)

MariaDB [lab_1]> alter table product_master
-> modify column sellprice numeric(10,2);
Query OK, 6 rows affected (0.047 sec)
Records: 6 Duplicates: 0 Warnings: 0

MariaDB [lab_1]> describe product_master;
```

Field	Type	Null	Key	Default	Extra
PRODUCT_No	char(6)	YES		NULL	
DESCRIPTION	varchar(20)	YES		NULL	
PROFITPERCENT	decimal(4,2)	YES		NULL	
UNITMEASURE	varchar(10)	YES		NULL	
QTYONHAND	decimal(8,0)	YES		NULL	
REORDERLVL	decimal(8,0)	YES		NULL	
sellprice	decimal(10,2)	YES		NULL	
COSTPRICE	decimal(8,2)	YES		NULL	

```
8 rows in set (0.014 sec)
```

10. Destroy the table client_master along with its data.

```
drop table client_master
```

```
ca: MySQL Client (MariaDB 10.5 (x64)) - "C:\Program Files\MariaDB 10.5\bin\mysql.exe" "--defaults-file=C:\Program Files\MariaDB 10.5\data\my.ini" -uroot -p
er' at line 1
MariaDB [lab_1]> show tables;
```

Tables_in_lab_1
client_master
product_master
sman_mast

```
3 rows in set (0.001 sec)

MariaDB [lab_1]> drop table client_master;
Query OK, 0 rows affected (0.011 sec)

MariaDB [lab_1]> show tables;
```

Tables_in_lab_1
product_master
sman_mast

```
2 rows in set (0.002 sec)
```

11. Change the name of salesman_master table to 'SMAN_MAST'.

```
alter table salesman_master  
rename to SMAN_MAST;
```

MySQL Client (MariaDB 10.5 (x64)) - "C:\Program Files\MariaDB 10.5\bin\mysql.exe" "--defaults-file=C:\Program Files\MariaDB 10.5\data\my.ini" -uroot -p

MariaDB [lab_1]> show tables;

```
+-----+  
| Tables_in_lab_1 |  
+-----+  
| product_master |  
| salesman_master |  
+-----+
```

2 rows in set (0.001 sec)

MariaDB [lab_1]> alter table salesman_master
-> rename to SMAN_MAST;

Query OK, 0 rows affected (0.012 sec)

MariaDB [lab_1]> show tables;

```
+-----+  
| Tables_in_lab_1 |  
+-----+  
| product_master |  
| sman_mast      |  
+-----+
```

2 rows in set (0.002 sec)

MariaDB [lab_1]> select* from sman_mast;

salesmanno	salesmanname	address1	address2	city	pincode	state	salamt	TgtToGet	YtdSales	Remarks
S00001	Aman	A/14	Worli	Pune	400002	Maharashtra	3000.00	100	50	good
S00002	Omkar	B-5	Nariman	Pune	400001	Maharashtra	3000.00	200	100	good
S00003	Raj	P-7	Bandra	Pune	400032	Maharashtra	3000.00	200	100	good

3 rows in set (0.005 sec)

VIVA QUESTIONS:

Q1. What are different DML commands?

Ans. DML (Data Manipulation Language): The SQL commands that deal with the manipulation of data present in the database belong to DML. The different DML commands are:

- **INSERT**
 - it is used to insert data into a table
- **UPDATE**
 - it is used to update existing data within a table
- **DELETE**
 - it is used to delete records from a database table
- **SELECT**
 - it is used to fetch data records from the database table

Q2. What is the purpose of ALTER command? What is the syntax?

Ans. The SQL **ALTER** command is used to add, delete or modify columns in an existing table.

SYNTAX

- To **add** a new column

```
ALTER TABLE table_name ADD column_name datatype;
```

- To **drop** a column

```
ALTER TABLE table_name DROP COLUMN column_name;
```

- To **change data type** of a column

```
ALTER TABLE table_name MODIFY COLUMN column_name datatype;
```

Q3. How do we add a column in a table?

Ans. We can add a new column to an existing table by making use of the **ALTER TABLE** command which is a part of DDL language.

```
ALTER TABLE table_name  
ADD column_name data_type column_constraint;
```

Q4. What is the purpose of DELETE command?

Ans. In the database structured query language (SQL), the **DELETE** statement is used to remove one or more records from a table. A subset may be defined for deletion using a condition, otherwise all records are removed.

Q5. What is the difference between ALTER and UPDATE commands?

Ans. The difference between ALTER and UPDATE commands are:

S.No.	ALTER	UPDATE
1.	It falls in the Data Definition Language (DDL).	It falls in the Data Manipulation Language (DML).
2.	It is used to add, delete or modify the attributes of the tables in the database.	It is used to update the existing records in a database.
3.	SYNTAX: <ul style="list-style-type: none"> Add a column <code>ALTER TABLE table_name ADD column_name datatype;</code> Drop a column <code>ALTER TABLE table_name DROP COLUMN column_name;</code> Modify a column <code>ALTER TABLE table_name MODIFY column_name data_type;</code> 	SYNTAX: <pre>UPDATE table_name SET column1=value1, column2=value2, ... WHERE condition;</pre>
4.	It will perform the action on structure level and not on the data level.	It will perform the action on the data level.
5.	By default, it initializes values of all the tuples as NULL.	It sets the specified values in the command to the tuples.
6.	It makes changes with the table structure.	It makes changes with the data that is inside the table.
7.	EX: Table structure, Table Name, SP, functions, etc.	EX: Change data in the table in rows or in column, etc.

EXPERIMENT-5

AIM: Write the queries to implement the concept of Integrity constraints like Primary Key, Foreign key, NOT NULL to the tables.

TOOLS USED: MariaDB

THEORY AND PROCEDURE:

In this experiment, we will see what constraints for a table are and how these constraints can be applied to each column/attributes of a table with their role for an attribute.

SQL Constraints:

SQL constraints are used to specify rules for the data in a table. Constraints can be used to limit the type of data that can go into table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column and Table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

- **NOT NULL:** Ensures that a column cannot have a NULL value.
- **UNIQUE:** Ensures that all values in a column are different.
- **PRIMARY KEY:** A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table.
- **FOREIGN KEY:** Uniquely identifies a row/record in another table.
- **CHECK:** Ensures that all values in a column satisfy a specific condition.
- **DEFAULT:** Sets a default value for a column when no value is specified.
- **INDEX:** Used to create and retrieve data from the database very quickly.

SQL NOT NULL Constraint:

By default, a column can hold NULL values. The NOT NULL constraint enforces a column to not accept NULL values. This enforces a field to always contain a value, which means you cannot insert a new record, or update a record without adding a value to this field.

Example:

```
Create table persons(  
    ID int NOT NULL,  
    LastName varchar(200) NOT NULL,  
    FirstName varchar(200) NOT NULL );
```

SOL UNIQUE Constraint:

The UNIQUE constraint ensures that all values in a column are different. Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns. A primary key constraint automatically has a unique constraint.

Example:

```
Create table Persons(  
    ID int NOT NULL UNIQUE,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);
```

SOL PRIMARY KEY Constraint:

The PRIMARY KEY constraint uniquely identifies each record in a database table.

Primary keys must contain UNIQUE values and cannot contain NULL values.

A table can have only one primary key, which may consist of single or multiple fields.

Example:

```
Create table Persons(  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY(ID)  
);
```

SOL FOREIGN KEY Constraint:

A FOREIGN KEY is a key used to link two tables together. A Foreign Key is a field in one table that refers to the Primary Key in another table. The table containing Foreign Key is called child table and table containing the candidate key is called parent or referenced table.

Example:

```
Create table Orders(  
    ID int NOT NULL PRIMARY KEY,  
    OrderNumber int NOT NULL,  
    PersonID int FOREIGN KEY REFERENCES Persons(PersonID)
```

v

);

SOL CHECK Constraint:

The CHECK Constraint can be used to limit the value range that can be placed in a column. If check constraint is defined on single column, it only allows certain values for this column. If it is defined on a table it can limit the values in certain columns based on values in other columns in the row.

Example:

```
Create table Persons(  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CHECK (Age>=18)  
);
```

SOL DEFAULT Constraint:

The DEFAULT constraint is used to provide a default value for a column. The default value will be added to all new records if no other value is specified.

Example:

```
Create table Persons(  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255) DEFAULT 'Sandnes'  
);
```

SOL CREATE INDEX Constraint:

The CREATE INDEX statement is used to create indexes in a table. Indexes are used to receive data from the database very fast.

Example:

```
CREATE INDEX index_name  
ON table_name (column1,column2,.....);
```

CREATE UNIQUE INDEX index_name

ON table_name (column1,column2,.....);

QUERIES

PART-1: CREATION OF TABLES:

(i) TABLE NAME: CLIENT_MASTER

```
create table CLIENT_MASTER(  
    Client_No varchar(6) primary key,  
    Name varchar(20) not null,  
    Address1 varchar(30),  
    Address2 varchar(30),  
    City varchar(15),  
    Pincode int(8),  
    State varchar(15),  
    Bal_Due float(10,2));
```

```
MariaDB [dbms]> create table CLIENT_MASTER(  
    -> Client_No varchar(6) primary key,  
    -> Name varchar(20) not null,  
    -> Address1 varchar(30),  
    -> Address2 varchar(30),  
    -> City varchar(15),  
    -> Pincode int(8),  
    -> State varchar(15),  
    -> Bal_Due float(10,2));
```

Query OK, 0 rows affected (0.037 sec)

```
MariaDB [dbms]> describe CLIENT_MASTER;
```

Field	Type	Null	Key	Default	Extra
Client_No	varchar(6)	NO	PRI	NULL	
Name	varchar(20)	NO		NULL	
Address1	varchar(30)	YES		NULL	
Address2	varchar(30)	YES		NULL	
City	varchar(15)	YES		NULL	
Pincode	int(8)	YES		NULL	
State	varchar(15)	YES		NULL	
Bal_Due	float(10,2)	YES		NULL	

8 rows in set (0.041 sec)

(ii) TABLE NAME: PRODUCT_MASTER:

```
create table PRODUCT_MASTER(  
    Product_No varchar(6) primary key,  
    Description varchar(15) NOT NULL,  
    Profit_Percent float(4,2) NOT NULL,  
    Unit_Measure varchar(10) NOT NULL,  
    Qty_On_Hand int(8) NOT NULL,  
    Reorder_Lvl int(8) NOT NULL,  
    Sell_Price float(8,2) NOT NULL,  
    Cost_Price float(8,2) NOT NULL);
```

```
MariaDB [dbms]> create table PRODUCT_MASTER(  
-> Product_No varchar(6) primary key,  
-> Description varchar(15) NOT NULL,  
-> Profit_Percent float(4,2) NOT NULL,  
-> Unit_Measure varchar(10) NOT NULL,  
-> Qty_On_Hand int(8) NOT NULL,  
-> Reorder_Lvl int(8) NOT NULL,  
-> Sell_Price float(8,2) NOT NULL,  
-> Cost_Price float(8,2) NOT NULL);  
Query OK, 0 rows affected (0.013 sec)
```

```
MariaDB [dbms]> describe PRODUCT_MASTER;
```

Field	Type	Null	Key	Default	Extra
Product_No	varchar(6)	NO	PRI	NULL	
Description	varchar(15)	NO		NULL	
Profit_Percent	float(4,2)	NO		NULL	
Unit_Measure	varchar(10)	NO		NULL	
Qty_On_Hand	int(8)	NO		NULL	
Reorder_Lvl	int(8)	NO		NULL	
Sell_Price	float(8,2)	NO		NULL	
Cost_Price	float(8,2)	NO		NULL	

8 rows in set (0.037 sec)

(iii) TABLE NAME :SALESMAN_MASTER

```
create table SALESMAN_MASTER(
  Salesman_No varchar(6) PRIMARY KEY,
  Salesman_Name varchar(20) NOT NULL,
  Address1 varchar(30) NOT NULL,
  Address2 varchar(30),
  City varchar(20),
  Pincode int(8),
  State varchar(20),
  Sal_Amt float(8,2) NOT NULL,
  Tgt_To_Get float(6,2) NOT NULL,
  Ytd_Sales float(6,2) NOT NULL,
  Remarks varchar(60));
```

```
MariaDB [dbms]> create table SALESMAN_MASTER(
  -> Salesman_No varchar(6) PRIMARY KEY,
  -> Salesman_Name varchar(20) NOT NULL,
  -> Address1 varchar(30) NOT NULL,
  -> Address2 varchar(30),
  -> City varchar(20),
  -> Pincode int(8),
  -> State varchar(20),
  -> Sal_Amt float(8,2) NOT NULL,
  -> Tgt_To_Get float(6,2) NOT NULL,
  -> Ytd_Sales float(6,2) NOT NULL,
  -> Remarks varchar(60));
```

Query OK, 0 rows affected (0.014 sec)

```
MariaDB [dbms]> describe SALESMAN_MASTER;
```

Field	Type	Null	Key	Default	Extra
Salesman_No	varchar(6)	NO	PRI	NULL	
Salesman_Name	varchar(20)	NO		NULL	
Address1	varchar(30)	NO		NULL	
Address2	varchar(30)	YES		NULL	
City	varchar(20)	YES		NULL	
Pincode	int(8)	YES		NULL	
State	varchar(20)	YES		NULL	
Sal_Amt	float(8,2)	NO		NULL	
Tgt_To_Get	float(6,2)	NO		NULL	
Ytd_Sales	float(6,2)	NO		NULL	
Remarks	varchar(60)	YES		NULL	

11 rows in set (0.038 sec)

(iv) TABLE NAME: SALES_ORDER

```
create table SALES_ORDER(  
    Order_No varchar(6) primary key,  
    Client_No varchar(6) references CLIENT_MASTER(Client_No),  
    Orderdate date,  
    Salesman_No varchar(6) references SALESMAN_MASTER(Salesman_No),  
    Delivtype char(1) default 'F',  
    Billyn char(1),  
    Delivdate date,  
    Orderstatus varchar(10),  
    constraint ck_Delivtype check(Delivtype in('P','F')),  
    constraint ck_Delivdate check(Delivdate>Orderdate),  
    constraint ck_Orderstatus check(Orderstatus in('In Process','Fulfilled','Backorder','Cancelled')));
```

```
MariaDB [dbms]> create table SALES_ORDER(  
-> Order_No varchar(6) primary key,  
-> Client_No varchar(6) references CLIENT_MASTER(Client_No),  
-> Orderdate date,  
-> Salesman_No varchar(6) references SALESMAN_MASTER(Salesman_No),  
-> Delivtype char(1) default 'F',  
-> Billyn char(1),  
-> Delivdate date,  
-> Orderstatus varchar(10),  
-> constraint ck_Delivtype check(Delivtype in('P','F')),  
-> constraint ck_Delivdate check(Delivdate>Orderdate),  
-> constraint ck_Orderstatus check(Orderstatus in('In Process','Fulfilled','Backorder','Cancelled')));  
Query OK, 0 rows affected (0.015 sec)
```

```
MariaDB [dbms]> describe SALES_ORDER;
```

Field	Type	Null	Key	Default	Extra
Order_No	varchar(6)	NO	PRI	NULL	
Client_No	varchar(6)	YES	MUL	NULL	
Orderdate	date	YES		NULL	
Salesman_No	varchar(6)	YES	MUL	NULL	
Delivtype	char(1)	YES		F	
Billyn	char(1)	YES		NULL	
Delivdate	date	YES		NULL	
Orderstatus	varchar(10)	YES		NULL	

```
8 rows in set (0.041 sec)
```


(v) TABLE NAME: SALES_ORDER_DETAILS

```
create table SALES_ORDER_DETAILS(
    Order_No varchar(6) references SALES_ORDER(Order_No),
    Product_No varchar(6) references PRODUCT_MASTER(Product_No),
    Qty_Ordered int(8),
    Qty_Dispatch int(8),
    Product_Rate float(10,2));
```

```
MariaDB [dbms]> create table SALES_ORDER_DETAILS(
-> Order_No varchar(6) references SALES_ORDER(Order_No),
-> Product_No varchar(6) references PRODUCT_MASTER(Product_No),
-> Qty_Ordered int(8),
-> Qty_Dispatch int(8),
-> Product_Rate float(10,2));
Query OK, 0 rows affected (0.015 sec)
```

```
MariaDB [dbms]> describe SALES_ORDER_DETAILS;
```

Field	Type	Null	Key	Default	Extra
Order_No	varchar(6)	YES	MUL	NULL	
Product_No	varchar(6)	YES	MUL	NULL	
Qty_Ordered	int(8)	YES		NULL	
Qty_Dispatch	int(8)	YES		NULL	
Product_Rate	float(10,2)	YES		NULL	

5 rows in set (0.035 sec)

PART-2: INSERTION OF DATA IN TABLES**DATA FOR 'CLIENT_MASTER' TABLE:**

```
insert into CLIENT_MASTER values('C00001','Ivan',' ','Mumbai',400054,'Maharashtra',15000);
insert into CLIENT_MASTER values('C00002','Mamta Muzumdar',' ','Madras',780001,'Tamil Nadu',0);
insert into CLIENT_MASTER values('C00003','Chhaya Bankar',' ','Mumbai',400057,'Maharashtra',5000);
insert into CLIENT_MASTER values('C00004','Ashwini Joshi',' ','Bangalore',560001,'Karnataka',0);
insert into CLIENT_MASTER values('C00005','Hansel Colaco',' ','Mumbai',400060,'Maharashtra',2000);
insert into CLIENT_MASTER values('C00006','Deepak Sharma',' ','Mangalore',560050,'Karnataka',0);
```

```
MariaDB [dbms]> insert into CLIENT_MASTER values('C00001','Ivan','','','Mumbai',400054,'Maharashtra',15000);
Query OK, 1 row affected (0.005 sec)

MariaDB [dbms]> insert into CLIENT_MASTER values('C00002','Mamta Muzumdar','','','Madras',780001,'Tamil Nadu',0);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into CLIENT_MASTER values('C00003','Chhaya Bankar','','','Mumbai',400057,'Maharashtra',5000);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into CLIENT_MASTER values('C00004','Ashwini Joshi','','','Banglore',560001,'Karnataka',0);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms]> insert into CLIENT_MASTER values('C00005','Hansel Colaco','','','Mumbai',400060,'Maharashtra',2000);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into CLIENT_MASTER values('C00006','Deepak Sharma','','','Mangalore',560050,'Karnataka',0);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> select * from CLIENT_MASTER;
+-----+-----+-----+-----+-----+-----+-----+
| Client_No | Name       | Address1 | Address2 | City      | Pincode | State      | Bal_Due |
+-----+-----+-----+-----+-----+-----+-----+
| C00001    | Ivan       |          |          | Mumbai    | 400054  | Maharashtra | 15000.00 |
| C00002    | Mamta Muzumdar |          |          | Madras     | 780001  | Tamil Nadu  | 0.00    |
| C00003    | Chhaya Bankar |          |          | Mumbai     | 400057  | Maharashtra | 5000.00 |
| C00004    | Ashwini Joshi |          |          | Banglore   | 560001  | Karnataka   | 0.00    |
| C00005    | Hansel Colaco |          |          | Mumbai     | 400060  | Maharashtra | 2000.00 |
| C00006    | Deepak Sharma |          |          | Mangalore  | 560050  | Karnataka   | 0.00    |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.003 sec)
```

DATA FOR 'PRODUCT_MASTER' TABLE:

```
insert into PRODUCT_MASTER values('P00001','T-Shirts',5,'Piece',200,50,350,250);

insert into PRODUCT_MASTER values('P0345','Shirts',6,'Piece',150,50,500,350);

insert into PRODUCT_MASTER values('P06734','Cotton Jeans',5,'Piece',100,20,600,450);

insert into PRODUCT_MASTER values('P07865','Jeans',5,'Piece',100,20,750,500);

insert into PRODUCT_MASTER values('P07868','Trousers',2,'Piece',150,50,850,550);

insert into PRODUCT_MASTER values('P07885','Pull Overs',2.5,'Piece',80,30,700,450);

insert into PRODUCT_MASTER values('P07965','Denim Shirts',4,'Piece',100,40,350,250);

insert into PRODUCT_MASTER values('P07975','Lycra Tops',5,'Piece',70,30,300,175);

insert into PRODUCT_MASTER values('P08865','Skirts',5,'Piece',75,30,450,300);
```

```

MariaDB [dbms]> insert into PRODUCT_MASTER values('P00001','T-Shirts',5,'Piece',200,50,350,250);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into PRODUCT_MASTER values('P0345','Shirts',6,'Piece',150,50,500,350);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into PRODUCT_MASTER values('P06734','Cotton Jeans',5,'Piece',100,20,600,450);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into PRODUCT_MASTER values('P07865','Jeans',5,'Piece',100,20,750,500);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into PRODUCT_MASTER values('P07868','Trousers',2,'Piece',150,50,850,550);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into PRODUCT_MASTER values('P07885','Pull Overs',2.5,'Piece',80,30,700,450);
Query OK, 1 row affected (0.005 sec)

MariaDB [dbms]> insert into PRODUCT_MASTER values('P07965','Denim Shirts',4,'Piece',100,40,350,250);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> insert into PRODUCT_MASTER values('P07975','Lycra Tops',5,'Piece',70,30,300,175);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into PRODUCT_MASTER values('P08865','Skirts',5,'Piece',75,30,450,300);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from PRODUCT_MASTER;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Product_No | Description | Profit_Percent | Unit_Measure | Qty_On_Hand | Reorder_Lvl | Sell_Price | Cost_Price |
+-----+-----+-----+-----+-----+-----+-----+-----+
| P00001 | T-Shirts | 5.00 | Piece | 200 | 50 | 350.00 | 250.00 |
| P0345 | Shirts | 6.00 | Piece | 150 | 50 | 500.00 | 350.00 |
| P06734 | Cotton Jeans | 5.00 | Piece | 100 | 20 | 600.00 | 450.00 |
| P07865 | Jeans | 5.00 | Piece | 100 | 20 | 750.00 | 500.00 |
| P07868 | Trousers | 2.00 | Piece | 150 | 50 | 850.00 | 550.00 |
| P07885 | Pull Overs | 2.50 | Piece | 80 | 30 | 700.00 | 450.00 |
| P07965 | Denim Shirts | 4.00 | Piece | 100 | 40 | 350.00 | 250.00 |
| P07975 | Lycra Tops | 5.00 | Piece | 70 | 30 | 300.00 | 175.00 |
| P08865 | Skirts | 5.00 | Piece | 75 | 30 | 450.00 | 300.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.001 sec)

```

DATA FOR 'SALESMAN_MASTER' TABLE:

```

insert into SALESMAN_MASTER values('S001','Kiran','A/14','Worli','Mumbai',400002,'Maharashtra',3000,100,50,'Good');

insert into SALESMAN_MASTER values('S002','Manish','65','Nariman','Mumbai',400001,'Maharashtra',3000,200,100,'Good');

insert into SALESMAN_MASTER values('S003','Ravi','P-7','Bandra','Mumbai',400032,'Maharashtra',3000,200,100,'Good');

insert into SALESMAN_MASTER values('S004','Ashish','A/5','Juhu','Mumbai',400044,'Maharashtra',3000,200,150,'Good');

```

```
MariaDB [dbms]> insert into SALESMAN_MASTER values('S001','Kiran','A/14','Worli','Mumbai',400002,'Maharashtra',3000,100,50,'Good');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into SALESMAN_MASTER values('S002','Manish','65','Nariman','Mumbai',400001,'Maharashtra',3000,200,100,'Good');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into SALESMAN_MASTER values('S003','Ravi','P-7','Bandra','Mumbai',400032,'Maharashtra',3000,200,100,'Good');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into SALESMAN_MASTER values('S004','Ashish','A/5','Juhu','Mumbai',400044,'Maharashtra',3000,200,150,'Good');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from SALESMAN_MASTER;
```

Salesman_No	Salesman_Name	Address1	Address2	City	Pincode	State	Sal_Amt	Tgt_To_Get	Ytd_Sales	Remarks
S001	Kiran	A/14	Worli	Mumbai	400002	Maharashtra	3000.00	100.00	50.00	Good
S002	Manish	65	Nariman	Mumbai	400001	Maharashtra	3000.00	200.00	100.00	Good
S003	Ravi	P-7	Bandra	Mumbai	400032	Maharashtra	3000.00	200.00	100.00	Good
S004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtra	3000.00	200.00	150.00	Good

```
4 rows in set (0.001 sec)
```

DATA FOR 'SALES_ORDER' TABLE:

```
insert into SALES_ORDER values('O19001','C00001','2010-01-12','S001','F','N','2010-01-20','In Process');
```

```
insert into SALES_ORDER values('O19002','C00002','2010-01-25','S002','P','N','2010-01-27','Cancelled');
```

```
insert into SALES_ORDER values('O46865','C00003','2010-02-18','S003','F','Y','2010-02-20','Fulfilled');
```

```
insert into SALES_ORDER values('O19003','C00001','2010-04-03','S001','F','Y','2010-04-07','Fulfilled');
```

```
insert into SALES_ORDER values('O46866','C00004','2010-05-20','S002','P','N','2010-05-22','Cancelled');
```

```
insert into SALES_ORDER values('O19008','C00005','2010-05-24','S004','F','N','2010-05-26','In Process');
```

```

MariaDB [dbms]> describe SALES_ORDER;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Order_No   | varchar(6) | NO   | PRI | NULL    |       |
| Client_No  | varchar(6) | YES  | MUL | NULL    |       |
| Orderdate  | date       | YES  |     | NULL    |       |
| Salesman_No | varchar(6) | YES  | MUL | NULL    |       |
| Delivtype  | char(1)    | YES  |     | F       |       |
| Billyn     | char(1)    | YES  |     | NULL    |       |
| Delivdate  | date       | YES  |     | NULL    |       |
| Orderstatus | varchar(10) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.059 sec)

MariaDB [dbms]> insert into SALES_ORDER values('019001','C00001','2010-01-12','S001','F','N','2010-01-20','In Process');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into SALES_ORDER values('019002','C00002','2010-01-25','S002','P','N','2010-01-27','Cancelled');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into SALES_ORDER values('046865','C00003','2010-02-18','S003','F','Y','2010-02-20','Fulfilled');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into SALES_ORDER values('019003','C00001','2010-04-03','S001','F','Y','2010-04-07','Fulfilled');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into SALES_ORDER values('046866','C00004','2010-05-20','S002','P','N','2010-05-22','Cancelled');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into SALES_ORDER values('019008','C00005','2010-05-24','S004','F','N','2010-05-26','In Process');
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> select * from SALES_ORDER;
+-----+-----+-----+-----+-----+-----+-----+
| Order_No | Client_No | Orderdate | Salesman_No | Delivtype | Billyn | Delivdate | Orderstatus |
+-----+-----+-----+-----+-----+-----+-----+
| 019001   | C00001    | 2010-01-12 | S001        | F         | N      | 2010-01-20 | In Process  |
| 019002   | C00002    | 2010-01-25 | S002        | P         | N      | 2010-01-27 | Cancelled   |
| 019003   | C00001    | 2010-04-03 | S001        | F         | Y      | 2010-04-07 | Fulfilled   |
| 019008   | C00005    | 2010-05-24 | S004        | F         | N      | 2010-05-26 | In Process  |
| 046865   | C00003    | 2010-02-18 | S003        | F         | Y      | 2010-02-20 | Fulfilled   |
| 046866   | C00004    | 2010-05-20 | S002        | P         | N      | 2010-05-22 | Cancelled   |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

```

DATA FOR 'SALES_ORDER_DETAILS' TABLE:

```

insert into sales_order_details values('O19001','P00001',4,4,525);
insert into sales_order_details values('O19001','P07965',2,1,8400);
insert into sales_order_details values('O19001','P07885',2,1,5250);
insert into sales_order_details values('O19002','P00001',10,0,525);
insert into sales_order_details values('O46865','P07868',3,3,3150);
insert into sales_order_details values('O46865','P07885',3,1,5250);
insert into sales_order_details values('O46865','P00001',10,10,525);
insert into sales_order_details values('O46865','P0345',4,4,1050);
insert into sales_order_details values('O19003','P0345',2,2,1050);

```

```
insert into sales_order_details values('O19003','P06734',1,1,12000);
insert into sales_order_details values('O46866','P07965',1,0,8400);
insert into sales_order_details values('O46866','P07975',1,0,1050);
insert into sales_order_details values('O19008','P00001',10,5,525);
insert into sales_order_details values('O19008','P07975',5,3,1050);
```

```
MariaDB [dbms]> insert into sales_order_details values('O19001','P00001',4,4,525);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into sales_order_details values('O19001','P07965',2,1,8400);
Query OK, 1 row affected (0.002 sec)

MariaDB [dbms]> insert into sales_order_details values('O19001','P07885',2,1,5250);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> insert into sales_order_details values('O19002','P00001',10,0,525);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> insert into sales_order_details values('O46865','P07868',3,3,3150);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into sales_order_details values('O46865','P07885',3,1,5250);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> insert into sales_order_details values('O46865','P00001',10,10,525);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into sales_order_details values('O46865','P0345',4,4,1050);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into sales_order_details values('O19003','P0345',2,2,1050);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into sales_order_details values('O19003','P06734',1,1,12000);
Query OK, 1 row affected (0.004 sec)

MariaDB [dbms]> insert into sales_order_details values('O46866','P07965',1,0,8400);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into sales_order_details values('O46866','P07975',1,0,1050);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into sales_order_details values('O19008','P00001',10,5,525);
Query OK, 1 row affected (0.003 sec)

MariaDB [dbms]> insert into sales_order_details values('O19008','P07975',5,3,1050);
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [dbms]> select * from SALES_ORDER_DETAILS;
```

Order_No	Product_No	Qty_Ordered	Qty_Disb	Product_Rate
019001	P00001	4	4	525.00
019001	P07965	2	1	8400.00
019001	P07885	2	1	5250.00
019002	P00001	10	0	525.00
046865	P07868	3	3	3150.00
046865	P07885	3	1	5250.00
046865	P00001	10	10	525.00
046865	P0345	4	4	1050.00
019003	P0345	2	2	1050.00
019003	P06734	1	1	12000.00
046866	P07965	1	0	8400.00
046866	P07975	1	0	1050.00
019008	P00001	10	5	525.00
019008	P07975	5	3	1050.00

```
14 rows in set (0.001 sec)
```


VIVA QUESTIONS

Q1. What are different Constraints in SQL?

Ans. The different constraints in SQL are:

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT
- INDEX

Q2. What is the purpose of Null Constraint?

Ans. By default, a column can hold NULL values. It enforces a column to NOT accept NULL values. It always contains a value, which means we cannot insert a value to that field.

NOT NULL constraint
If a column has a NOT NULL constraint, this enforces the field to contain a value. It means we cannot insert a record without adding a value to that field.

Q3. What is the function of Reference Constraint?

Ans. A Foreign Key is a key used to link two tables together. A Foreign Key is a field or a collection of fields in one table that refers to the Primary Key in another table. The table containing the Foreign Key is called the child table, and the table containing the Candidate Key is called the parent table or referenced table.

Q4. What is Index Constraint?

Ans. The Index Constraint is used to create indexes in tables. Indexes are used to retrieve data from the database very fast. It speeds up the searches and queries. The users cannot see these indexes.

Q5. What is the purpose of Default Constraint?

Ans. The Default Constraint is used to provide a default value for a column. The default value will be added to all the new records IF no other value is specified.

Experiment – 6

Aim: To write the queries for implementing the following functions: MAX(), MIN(), AVG(), COUNT() and other logical and pattern matching operations.

Tool Used: MariaDB

Theory: SQL has many built-in functions for performing calculations on data. SQL aggregate functions return a single value, calculated from values in a column. Useful aggregate functions:

- **AVG()**
 - The AVG () function returns the average value of a numeric column.
 - Syntax:
`SELECT AVG(column_name)`
`FROM table_name`
`WHERE condition;`
- **COUNT()**
 - The COUNT () function returns the number of rows that matches a specified criterion.
 - Syntax:
`SELECT COUNT(column_name)`
`FROM table_name`
`WHERE condition;`
- **MAX()**
 - The MAX () function returns the largest value of the selected column.
 - Syntax:
`SELECT MAX(column_name)`
`FROM table_name`
`WHERE condition;`
- **MIN()**
 - The MIN () function returns the smallest value of the selected column.

- Syntax:
`SELECT MIN(column_name)`
`FROM table_name`
`WHERE condition;`

- **SUM()**

- The SUM () function returns the total sum of a numeric column.
- Syntax:
`SELECT SUM(column_name)`
`FROM table_name`
`WHERE condition;`

In SQL, we sometimes need to filter our resultset based on some pattern matching techniques. SQL has a standard pattern matching technique using the '**LIKE**' operator. But, it also supports the regular expression pattern matching for better functionality. Pattern matching operator:

- **LIKE**

- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
- There are two wildcards often used in conjunction with the LIKE operator:
 - The percent sign (%) represents zero, one, or multiple characters
 - The underscore sign (_) represents one, single character
- Syntax:
`SELECT column1, column2, ...`
`FROM table_name`
`WHERE columnN LIKE pattern;`

QUERIES

Q1. List the names of all clients having “a” as the second letter in their names.

select name from CLIENT_MASTER

-> where name like "_a%";

```
MariaDB [dbms]> select name from CLIENT_MASTER
-> where name like "_a%";
+-----+
| name      |
+-----+
| Mamta Muzumdar |
| Hansel Colaco  |
+-----+
2 rows in set (0.097 sec)
```

Q2. List the clients who stay in the city whose first letter in “M”.

select name, city from CLIENT_MASTER

-> where city like "M%";

```
MariaDB [dbms]> select name, city from CLIENT_MASTER
-> where city like "M%";
+-----+-----+
| name      | city      |
+-----+-----+
| Ivan      | Mumbai    |
| Mamta Muzumdar | Madras    |
| Chhaya Bankar | Mumbai    |
| Hansel Colaco | Mumbai    |
| Deepak Sharma | Mangalore |
+-----+-----+
5 rows in set (0.001 sec)
```

Q3. List all the clients who stay in “Bangalore” or “Mangalore”.

select name, city from CLIENT_MASTER

-> where city in("Bangalore","Mangalore");

```
MariaDB [dbms]> select name, city from CLIENT_MASTER
-> where city in("Bangalore","Mangalore");
+-----+-----+
| name      | city      |
+-----+-----+
| Deepak Sharma | Mangalore |
+-----+-----+
1 row in set (0.001 sec)
```

Q4. List all the clients whose BAL_DUE = 10,000.

```
select name from CLIENT_MASTER
```

```
-> where bal_due=10000;
```

```
MariaDB [dbms]> select name from CLIENT_MASTER
-> where bal_due=10000;
Empty set (0.004 sec)
```

Q5. List all the information from the SALES_ORDER for orders placed in the month of June.

```
select * from SALES_ORDER
```

```
-> where month(Orderdate)=06;
```

```
MariaDB [dbms]> select * from SALES_ORDER
-> where month(Orderdate)=06;
Empty set (0.003 sec)
```

Q6. List the order information for the client number "C00001" and "C00002".

```
select * from SALES_ORDER
```

```
-> where Client_No="C00001" or Client_No="C00002";
```

```
MariaDB [dbms]> select * from SALES_ORDER
-> where Client_No="C00001" or Client_No="C00002";
+-----+-----+-----+-----+-----+-----+-----+
| Order_No | Client_No | Orderdate | Salesman_No | Delivtype | Billyn | Delivdate | Orderstatus |
+-----+-----+-----+-----+-----+-----+-----+
| 019001 | C00001 | 2010-01-12 | S001 | F | N | 2010-01-20 | In Process |
| 019003 | C00001 | 2010-04-03 | S001 | F | Y | 2010-04-07 | Fulfilled |
| 019002 | C00002 | 2010-01-25 | S002 | P | N | 2010-01-27 | Cancelled |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.004 sec)
```

Q7. List the products whose selling price is greater than 500 and less than or equal to 750.

select description, sell_price from PRODUCT_MASTER

-> where sell_price>500 and sell_price<=750;

```
MariaDB [dbms]> select description, sell_price from PRODUCT_MASTER
-> where sell_price>500 and sell_price<=750;
+-----+-----+
| description | sell_price |
+-----+-----+
| Cotton Jeans | 600.00 |
| Jeans | 750.00 |
| Pull Overs | 700.00 |
+-----+-----+
3 rows in set (0.001 sec)
```

Q8. List the products whose selling price is more than 500. Calculate a new selling price as original selling price * 0.15.

select description, sell_price, sell_price*0.15

-> from PRODUCT_MASTER

-> where sell_price>500;

```
MariaDB [dbms]> select description, sell_price, sell_price*0.15
-> from PRODUCT_MASTER
-> where sell_price>500;
+-----+-----+-----+
| description | sell_price | sell_price*0.15 |
+-----+-----+-----+
| Cotton Jeans | 600.00 | 90.00 |
| Jeans | 750.00 | 112.50 |
| Trousers | 850.00 | 127.50 |
| Pull Overs | 700.00 | 105.00 |
+-----+-----+-----+
4 rows in set (0.004 sec)
```

Q9. Rename the new column in the output of above query as NEW_PRICE.

select description, sell_price, sell_price*0.15 as New_price

-> from PRODUCT_MASTER

-> where sell_price>500;

```
MariaDB [dbms]> select description, sell_price, sell_price*0.15 as New_price
-> from PRODUCT_MASTER
-> where sell_price>500;
+-----+-----+-----+
| description | sell_price | New_price |
+-----+-----+-----+
| Cotton Jeans | 600.00 | 90.00 |
| Jeans | 750.00 | 112.50 |
| Trousers | 850.00 | 127.50 |
| Pull Overs | 700.00 | 105.00 |
+-----+-----+-----+
4 rows in set (0.001 sec)
```

Q10. List the name, city of clients who are not in the state of “Maharashtra”.

select name, city from CLIENT_MASTER

-> where state!="Maharashtra";

```
MariaDB [dbms]> select name, city from CLIENT_MASTER
-> where state!="Maharashtra";
+-----+-----+
| name | city |
+-----+-----+
| Mamta Muzumdar | Madras |
| Ashwini Joshi | Bangalore |
| Deepak Sharma | Mangalore |
+-----+-----+
3 rows in set (0.002 sec)
```

Q11. Count the total number of orders.

select count(Distinct Order_No) as Total_Orders

-> from SALES_ORDER;

```
MariaDB [dbms]> select count(Distinct Order_No) as Total_Orders
-> from SALES_ORDER;
+-----+
| Total_Orders |
+-----+
| 6 |
+-----+
1 row in set (0.006 sec)
```

Q12. Calculate the average price of all the products.

select AVG(Sell_Price) from PRODUCT_MASTER;

```
MariaDB [dbms]> select AVG(Sell_Price) from PRODUCT_MASTER;
+-----+
| AVG(Sell_Price) |
+-----+
| 538.888889 |
+-----+
1 row in set (0.001 sec)
```

Q13. Determine the maximum and minimum product prices. Rename the output as MAX_PRICE and MIN_PRICE respectively.

```
select MAX(Sell_Price) as MAX_PRICE, MIN(Sell_Price) as MIN_PRICE from PRODUCT_MASTER;
```

```
MariaDB [dbms]> select MAX(Sell_Price) as MAX_PRICE, MIN(Sell_Price) as MIN_PRICE from PRODUCT_MASTER;
+-----+-----+
| MAX_PRICE | MIN_PRICE |
+-----+-----+
|      850.00 |       300.00 |
+-----+-----+
1 row in set (0.001 sec)
```

Q14. Count the number of products having price less than or equal to 500.

```
select COUNT(Product_No) from PRODUCT_MASTER where Sell_Price<=500;
```

```
MariaDB [dbms]> select COUNT(Product_No) from PRODUCT_MASTER where Sell_Price<=500;
+-----+
| COUNT(Product_No) |
+-----+
|                5 |
+-----+
1 row in set (0.001 sec)
```

Q15. List all the products whose QTY_ON_HAND is less than 3 times the REORDER_LVL.

```
select Description, Qty_On_Hand, 3*Reorder_Lvl
```

```
-> from PRODUCT_MASTER
```

```
-> Where Qty_On_Hand<3*Reorder_Lvl;
```

```
MariaDB [dbms]> select Description, Qty_On_Hand, 3*Reorder_Lvl
-> from PRODUCT_MASTER
-> Where Qty_On_Hand<3*Reorder_Lvl;
+-----+-----+-----+
| Description | Qty_On_Hand | 3*Reorder_Lvl |
+-----+-----+-----+
| Pull Overs |          80 |          90 |
| Denim Shirts |         100 |         120 |
| Lycra Tops |          70 |          90 |
| Skirts |          75 |          90 |
+-----+-----+-----+
4 rows in set (0.001 sec)
```

Viva Questions

Q1. What are Pattern Matching Operations?

Ans. In SQL, we sometimes need to filter our result set based on some pattern matching techniques. SQL has a standard pattern matching technique using the '**LIKE**' operator. But, it also supports the regular expression pattern matching for better functionality.

LIKE Operator

The LIKE operator provides standard pattern matching in SQL that is always used after a WHERE clause. It matches any pattern based on some conditions provided using the wildcard characters.

Some of the commonly used wildcard characters in MySQL are as follows:

- '%' represents zero or more characters.
- '_' represents exactly 1 character.

Q2. What are different variations in LIKE command?

Ans. The different variations in LIKE command are:

LIKE operator	Description
WHERE CustomerName LIKE "a%"	Finds any value that starts with "a"
WHERE CustomerName LIKE "%a"	Finds any value that ends with "a"
WHERE CustomerName LIKE "%or%"	Finds any value that has "or" in any position

WHERE CustomerName LIKE “_r%”	Finds any value that has “r” in the second position
WHERE CustomerName LIKE “a_%_%”	Finds any value that starts with “a” and is at least 3 characters in length
WHERE CustomerName LIKE “a%o”	Finds any value that starts with “a” and ends with “o”

Q3. What are Arithmetic Operations?

Ans. Arithmetic operators can perform arithmetical operations on numeric operands involved. Arithmetic operators are:

Operator	Meaning
+ (Add)	Addition
- (Subtract)	Subtraction
* (Multiply)	Multiplication
/ (Divide)	Division
% (Mudulo)	Returns the integer remainder of a division

Q4. What are different Logical Operations?

Ans. Logical operators are those that are true or false. They return either a true or a false value to combine one or more true or false values.

Operator	Description
ALL	TRUE if all of the subquery values meet the condition
AND	TRUE if all the conditions separated by AND is TRUE
ANY	TRUE if any of the subquery values meet the conditions
BETWEEN	TRUE if the operand is within the range of comparisons
EXISTS	TRUE if the subquery returns one or more records
IN	TRUE if the operand is equal to any one of a list of expressions
NOT	Displays a record if the condition(s) is NOT TRUE

OR	TRUE if any of the conditions separated by OR is TRUE
----	---

Q5. What is the difference between BETWEEN and IN commands?

Ans. The differences between BETWEEN and IN commands are:

- **BETWEEN Operator**

- *The BETWEEN operator selects a range of data between two values.*

- *The values can be numbers, text, etc.*

- **Syntax:**

```
SELECT * FROM table_name  
WHERE column_name BETWEEN 'value1' AND 'value2'
```

- **Example:**

```
SELECT * FROM STUDENT  
WHERE marks BETWEEN 50 AND 80;
```

- Here, only those records will be returned in which the value of marks is between 50 and 80 including both 50 and 80.

- **IN Operator**

- *The IN operator allows you to specify multiple values.*

- **Syntax:**

```
SELECT * FROM table_name  
WHERE column_name IN ('value1','value 2')
```

- **Example:**

```
SELECT * FROM STUDENT  
WHERE marks IN (50,80);
```

- Here, only those records will be returned in which the value of marks is either 50 or 80.