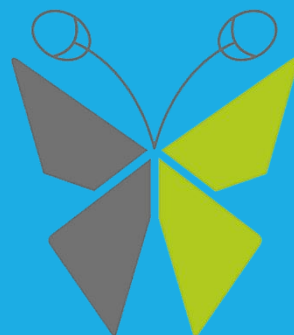


STORE DATABASE

SQL Project

I.T. VEDANT

- Swapnil Sawant





Introduction:

The SQL Store Database Project is designed to create a database system that can manage the inventory, sales, and customer information for a hypothetical retail store. This project aims to provide an efficient, user-friendly, and secure solution for managing store data. The Store Database is designed to streamline the operations of the store by automating many of the manual processes involved in managing inventory and sales.

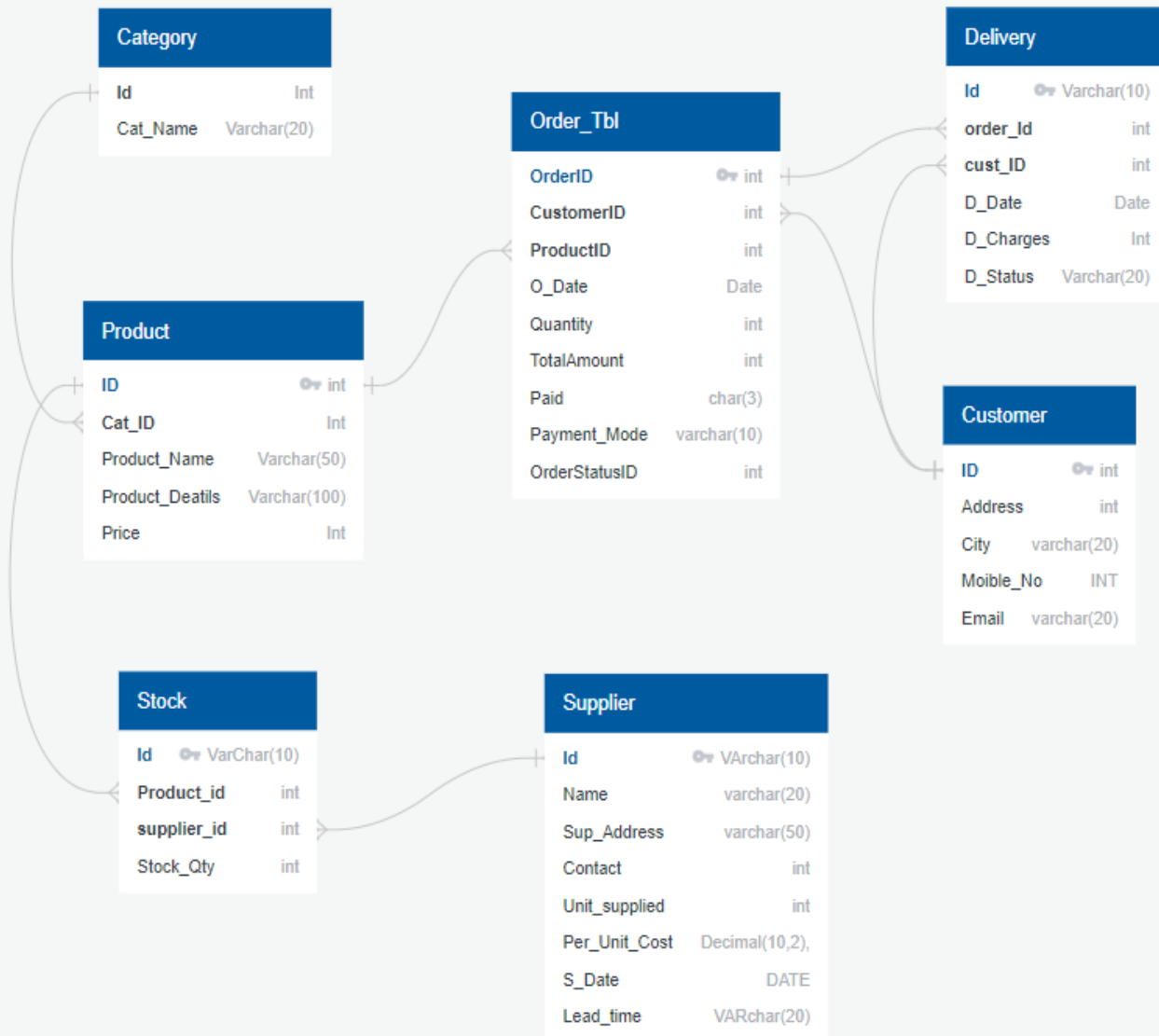
The project consists of several components, including a database schema, Entity Relationship Diagram, Tables, SQL different queries. The database schema includes tables, attributes, and relationships that are designed to store and retrieve data efficiently. The data modeling process involves identifying data entities, attributes, and relationships and normalizing the data to ensure data integrity and security.

The implementation of the Store Database includes creating the database schema, populating the database with sample data, and implementing the necessary queries and reports. Using this Database schema allowing store managers and employees to access and update the data in the database quickly.

The SQL Store Database Project is an essential tool for managing store operations, including inventory management, sales, and customer relationship management. By automating many of the manual processes involved in managing store data, the Store Database frees up valuable time and resources that can be used to improve other aspects of the business.

Overall, the SQL Store Database Project is an innovative and efficient solution for managing store data. It is an excellent example of how technology can be used to streamline operations and improve the overall performance of a business.

1. Entity Relationship Diagram:-



2. Database Design:

Database: Store

Tables:

- a. Order_Tbl*
- b. Customer*
- c. Product*
- d. Category*
- e. Stock*
- f. Supplier*
- g. Delivery*

3.Data Definition Language (DDL)

A. Create Database

```
MariaDB [(none)]> create Database Store;  
Query OK, 1 row affected (0.009 sec)
```

```
MariaDB [Store]> show databases;  
+-----+  
| Database  
+-----+  
| cat_app  
| information_schema  
| mysql  
| performance_schema  
| phpmyadmin  
| sql2to5  
| store  
| store_inventory  
| test  
+-----+
```

```
MariaDB [(none)]> use Store;  
Database changed
```

B. Create Tables

a. Customer

```
MariaDB [store]> CREATE TABLE Customer (
->   ID INT PRIMARY KEY,
->   First_Name Varchar(20) NOT NULL ,
->   Last_Name Varchar(20) NOT NULL ,
->   Gender varchar(10) ,
->   Address Varchar(100) NOT NULL ,
->   City varchar(20) ,
->   Mobile_No INT NOT NULL ,
->   Email varchar(20) default 'abc@gmail.com'
-> );
```

```
MariaDB [store]> desc customer;
```

| Field | Type | Null | Key | Default | Extra |
|------------|--------------|------|-----|---------------|-------|
| ID | int(11) | NO | PRI | NULL | |
| First_Name | varchar(20) | NO | | NULL | |
| Last_Name | varchar(20) | NO | | NULL | |
| Gender | varchar(10) | YES | | NULL | |
| Address | varchar(100) | NO | | NULL | |
| City | varchar(20) | YES | | NULL | |
| Mobile_NO | varchar(11) | NO | | NULL | |
| Email | varchar(20) | YES | | abc@gmail.com | |

b. Order_Tbl

```
MariaDB [Store]> CREATE TABLE OrderData (
->   OrderID int PRIMARY KEY Auto_increment ,
->   CustomerID int ,
->   ProductID int ,
->   O_Date Date DEFAULT Curdate() ,
->   Quantity int ,
->   TotalAmount int NOT NULL ,
->   Paid char(3) default 'NO',
->   Mode varchar(10) NOT NULL ,
->   FOREIGN KEY(CustomerID) REFERENCES Customer(ID) ON Delete Cascade,
->   FOREIGN KEY(ProductID) REFERENCES Product(ID) ON Delete Cascade
-> );
```

```
MariaDB [store]> desc order_Tbl;
```

| Field | Type | Null | Key | Default | Extra |
|-------------|-------------|------|-----|-----------|----------------|
| OrderID | int(11) | NO | PRI | NULL | auto_increment |
| CustomerID | int(11) | YES | MUL | NULL | |
| ProductID | int(11) | YES | MUL | NULL | |
| O_Date | date | YES | | curdate() | |
| Quantity | int(11) | YES | | NULL | |
| TotalAmount | int(11) | NO | | NULL | |
| Paid | char(3) | YES | | NO | |
| Mode | varchar(10) | NO | | NULL | |

c. Category

```
MariaDB [Store]> CREATE TABLE Category (
->     ID Int PRIMARY KEY Auto_increment ,
->     Catg_Name Varchar(20) UNIQUE );
Query OK, 0 rows affected (0.037 sec)
```

```
MariaDB [store]> desc category;
```

| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|----------------|
| ID | int(11) | NO | PRI | NULL | auto_increment |
| Catg_Name | varchar(20) | YES | UNI | NULL | |

d. Product

```
MariaDB [Store]> CREATE TABLE Product (
->     ID int PRIMARY KEY ,
->     Catg_ID Int NOT NULL ,
->     Product_Name Varchar(50) ,
->     Product_Deatils Varchar(100),
->     Price DECIMAL(10,2),
->     FOREIGN KEY(Catg_ID) REFERENCES Category(ID) ON DELETE CASCADE
-> );
Query OK, 0 rows affected (0.033 sec)
```



```
MariaDB [store]> desc Product;
```

| Field | Type | Null | Key | Default | Extra |
|-----------------|---------------|------|-----|---------|-------|
| ID | int(11) | NO | PRI | NULL | |
| Catg_ID | int(11) | NO | MUL | NULL | |
| Product_Name | varchar(50) | YES | | NULL | |
| Product_Details | varchar(100) | YES | | NULL | |
| Price | decimal(10,2) | YES | | NULL | |

e. Stock

```
MariaDB [Store]> CREATE TABLE Stock (
  ->   Id VarChar(10) PRIMARY KEY,
  ->   Product_ID Int ,
  ->   Supplier_ID VarChar(10),
  ->   Stock_Qty_Left int NOT NULL,
  ->   FOREIGN KEY(Product_ID) REFERENCES Product(ID),
  ->   FOREIGN KEY(Supplier_ID) REFERENCES Supplier(ID)
  -> );
Query OK, 0 rows affected (0.028 sec)
```

```
MariaDB [store]> desc stock
```

```
-> ;
```

| Field | Type | Null | Key | Default | Extra |
|----------------|-------------|------|-----|---------|-------|
| Id | varchar(10) | NO | PRI | NULL | |
| Product_ID | int(11) | YES | MUL | NULL | |
| Supplier_ID | varchar(10) | YES | MUL | NULL | |
| Stock_Qty_Left | int(11) | NO | | NULL | |

f. Supplier

```

MariaDB [Store]> CREATE TABLE Supplier (
  ->   Id Varchar(10) PRIMARY KEY,
  ->   Name Varchar(20) NOT NULL ,
  ->   Sup_Address Varchar(100) NOT NULL ,
  ->   Contact Varchar(20) NOT NULL ,
  ->   Unit_supplied int NOT NULL ,
  ->   Per_Unit_Cost decimal(10,2) ,
  ->   S_Date DATE ,
  ->   Lead_time Varchar(20)
  -> );
Query OK, 0 rows affected (0.022 sec)

```

```

MariaDB [store]> desc Supplier;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Id             | varchar(10)   | NO   | PRI | NULL     |       |
| Name           | varchar(20)   | NO   |     | NULL     |       |
| Sup_Address    | varchar(100)  | NO   |     | NULL     |       |
| Contact        | varchar(20)   | NO   |     | NULL     |       |
| Unit_supplied  | int(11)       | NO   |     | NULL     |       |
| Per_Unit_Cost  | decimal(10,2) | YES  |     | NULL     |       |
| S_Date         | date          | YES  |     | NULL     |       |
| Lead_time      | varchar(20)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+

```

g. Delivery

```

MariaDB [Store]> CREATE TABLE Delivery (
  ->   ID Varchar(10) PRIMARY KEY ,
  ->   order_ID int ,
  ->   Cust_ID int ,
  ->   D_Date Date DEFAULT DATE_ADD(CURDATE(), INTERVAL 15 DAY),
  ->   D_Charges INT ,
  ->   D_Status Varchar(20) NOT NULL ,
  ->   FOREIGN KEY(order_ID) REFERENCES Order_Tbl(orderID) ON DELETE Cascade,
  ->   FOREIGN KEY(Cust_ID) REFERENCES Customer(ID) ON DELETE Cascade
  -> );

```

```
MariaDB [store]> desc delivery;
```

| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|-------------------------------|-------|
| ID | varchar(10) | NO | PRI | NULL | |
| order_ID | int(11) | YES | MUL | NULL | |
| Cust_ID | int(11) | YES | MUL | NULL | |
| D_Date | date | YES | | (curdate() + interval 15 day) | |
| D_Charges | int(11) | YES | | NULL | |
| D_Status | varchar(20) | NO | | NULL | |

C. Alter Table:

a. Alter Column Name

```
[Store]> Alter Table Customer change Moible_No Mobile_No INT;  
Query OK, 0 rows affected (0.033 sec)
```

b. Alter Add Column

```
Alter Table Customer add Mobile_NO varchar(11) NOT NULL after City;  
-----+-----+-----+-----+
```

c. Alter Change Datatype

```
MariaDB [Store]> Alter Table Customer modify Mobile_No Varchar(10);  
Query OK, 6 rows affected (0.056 sec)
```

d. Alter Delete Column

```
MariaDB [Store]> Alter Table Customer drop Mobile_no;  
Query OK, 0 rows affected (0.018 sec)
```

e. Alter Table Name

```
MariaDB [Store]> Alter Table OrderData RENAME to Order_Tbl;  
Query OK, 0 rows affected (0.011 sec)
```

4.Data Manipulation language (DML)

A. Insert

```
MariaDB [Store]> INSERT INTO Customer (Id, First_Name, Last_Name,Gender,Address, City, Mobile_No ,Email) VALUES  
-> (1001, 'John', 'Done', 'Male','12, Juhu Road, Maharashtra - 400049','Mumbai',2525896329, 'John@gmail.com'),
```

```
MariaDB [Store]> INSERT INTO Order_Tbl  
-> (CustomerID,ProductID,O_Date,Quantity,TotalAmount,Paid,Mode)  
-> VALUES(1001,101,'2022-12-24',1,30200,'YES','Online');  
Query OK, 1 row affected (0.011 sec)
```

```
MariaDB [Store]> INSERT INTO Category (Catg_Name) VALUES ('TV'),('Smart_Phones'),('Refrigerator');  
Query OK, 3 rows affected (0.012 sec)
```

```
MariaDB [Store]> INSERT INTO Product (ID,Catg_ID,Product_Name,Product_Details,Price)  
-> VALUES  
-> (101,1,'Samsung Crystal 4K ','108 cm (43 inch) Ultra HD(4K)LED Smart Tizen TV',30000);
```

```
MariaDB [Store]> INSERT INTO Stock (Id,Product_ID,Supplier_ID,Stock_Qty_Left)  
-> VALUES('S101',101,'SAM-1C',10);  
Query OK, 1 row affected (0.010 sec)
```

```
MariaDB [Store]> INSERT INTO Delivery(ID,Order_ID,Cust_ID,D_Date,D_Charges,D_Status)  
-> values('D-1',1,1001,'2022-12-25',200,'Delivered');  
Query OK, 1 row affected (0.010 sec)
```

```
MariaDB [Store]> INSERT INTO Supplier(ID,Name,Sup_Address,Contact,Unit_Supplied,Per_Unit_Cost,S_Date,Lead_Time)  
-> VALUES  
-> ('SAM-1C','SAMSUNG_TV Pvt.Ltd.','123, Carter Road, Bandra West, Mumbai, Maharashtra 400050','998579632',15,25000,'2022-10-11','05 Days');  
Query OK, 1 row affected (0.011 sec)
```

B. Update

```
MariaDB [store]> update order_tbl set paid='YES' where orderID=5;
Query OK, 1 row affected (0.018 sec)
```

```
MariaDB [Store]> Update Customer set Mobile_NO='2525896329' where ID=1001;
Query OK, 1 row affected (0.011 sec)
```

C. Delete

```
MariaDB [store]> Delete from customer where Id=1010;
Query OK, 1 row affected (0.011 sec)
```

5. Data Query Language (DQL)

A. Select Query

```
MariaDB [store]> select * from order_Tbl;
```

| OrderID | CustomerID | ProductID | O_Date | Quantity | TotalAmount | Paid | Mode |
|---------|------------|-----------|------------|----------|-------------|------|--------|
| 1 | 1001 | 101 | 2022-12-24 | 1 | 30200 | YES | Online |
| 2 | 1001 | 202 | 2022-12-24 | 1 | 80200 | YES | Online |
| 3 | 1002 | 102 | 2023-01-02 | 1 | 59200 | YES | Online |
| 4 | 1003 | 301 | 2023-01-15 | 1 | 25400 | YES | Online |
| 5 | 1004 | 201 | 2023-02-01 | 1 | 27000 | YES | COD |
| 6 | 1005 | 203 | 2023-02-24 | 2 | 38200 | NO | COD |
| 7 | 1006 | 104 | 2023-02-24 | 1 | 20000 | NO | COD |

```
MariaDB [store]> select * from category;
```

| ID | Catg_Name |
|----|--------------|
| 3 | Refrigerator |
| 2 | Smart_Phones |
| 1 | TV |

B. Order By Query ASC

```
MariaDB [store]> select Id, Product_Name , Price from Product order by Price Asc;
```

| Id | Product_Name | Price |
|-----|---------------------------|----------|
| 302 | SAMSUNG 189 L | 18000.00 |
| 104 | OnePlus Y1 | 18999.00 |
| 203 | OnePlus Nord CE 2 Lite 5G | 19000.00 |
| 301 | SAMSUNG 253 L | 24990.00 |
| 201 | POCO X5 Pro | 25000.00 |
| 204 | OnePlus 10T | 30000.00 |
| 101 | Samsung Crystal 4K | 30000.00 |
| 103 | Mi X Series | 37999.00 |
| 102 | OnePlus U1S | 59000.00 |
| 202 | SAMSUNG Galaxy S23 5G | 79999.00 |

C. Order By Query DESC

```
MariaDB [store]> select Id, Product_Name , Price from Product order by Price desc;
```

| Id | Product_Name | Price |
|-----|---------------------------|----------|
| 202 | SAMSUNG Galaxy S23 5G | 79999.00 |
| 102 | OnePlus U1S | 59000.00 |
| 103 | Mi X Series | 37999.00 |
| 204 | OnePlus 10T | 30000.00 |
| 101 | Samsung Crystal 4K | 30000.00 |
| 201 | POCO X5 Pro | 25000.00 |
| 301 | SAMSUNG 253 L | 24990.00 |
| 203 | OnePlus Nord CE 2 Lite 5G | 19000.00 |
| 104 | OnePlus Y1 | 18999.00 |
| 302 | SAMSUNG 189 L | 18000.00 |

D. Order By Two Column

```
MariaDB [store]> select ID, Product_Name, Price
-> from product Order by price desc, Product_name asc;
```

| ID | Product_Name | Price |
|-----|---------------------------|----------|
| 202 | SAMSUNG Galaxy S23 5G | 79999.00 |
| 102 | OnePlus U1S | 59000.00 |
| 103 | Mi X Series | 37999.00 |
| 204 | OnePlus 10T | 30000.00 |
| 101 | Samsung Crystal 4K | 30000.00 |
| 201 | POCO X5 Pro | 25000.00 |
| 301 | SAMSUNG 253 L | 24990.00 |
| 203 | OnePlus Nord CE 2 Lite 5G | 19000.00 |
| 104 | OnePlus Y1 | 18999.00 |
| 302 | SAMSUNG 189 L | 18000.00 |

E. Limit Query

```
MariaDB [store]> select Id, Product_Name , Price from Product
-> where Product_name like 'S_s%' limit 1,2;
```

| Id | Product_Name | Price |
|-----|-----------------------|----------|
| 202 | SAMSUNG Galaxy S23 5G | 79999.00 |
| 301 | SAMSUNG 253 L | 24990.00 |

F. Select Query with Specific Column

```
MariaDB [store]> select ID, Name, Sup_Address,Contact from Supplier;
```

| ID | Name | Sup_Address | Contact |
|------------|----------------------|---|-------------|
| OPL-1U | OnePlus_TV | 456, SV Road, Andheri West, Mumbai,Maharashtra 400058 | 9687851298 |
| REDMI-1X | REDMI_TV | 1213, MG Road, Goregaon West, Mumbai, Maharashtra 400062 | 3569874526 |
| REDMI-2PX | REDMI_SM | 1214, MG Road, Goregaon West, Mumbai, Maharashtra 400062 | 65478521569 |
| SAM-1C | SAMSUNG_TV Pvt.Ltd. | 123, Carter Road, Bandra West, Mumbai, Maharashtra 400050 | 998579632 |
| SAM-2S23 | SAMSUNG_SM Pvt.Ltd. | 124, Carter Road, Bandra West, Mumbai, Maharashtra 400050 | 78569845326 |
| SAM-3-253L | SAMSUNG_REF Pvt.Ltd. | 125, Carter Road, Bandra West, Mumbai, Maharashtra 400050 | 54785147859 |

G. Distinct Query

```
MariaDB [store]> select distinct Name from Supplier;
```

| Name |
|----------------------|
| OnePlus_TV |
| REDMI_TV |
| REDMI_SM |
| SAMSUNG_TV Pvt.Ltd. |
| SAMSUNG_SM Pvt.Ltd. |
| SAMSUNG_REF Pvt.Ltd. |

6. Where Clause with Comparison Operator

```
MariaDB [store]> select order_id , D_charges,D_status from Delivery where D_status = 'NULL';
```

| order_id | D_charges | D_status |
|----------|-----------|----------|
| 6 | 200 | NULL |
| 7 | 100 | NULL |

```
MariaDB [store]> select * from delivery where D_Charges > 200;
```

| ID | order_ID | Cust_ID | D_Date | D_Charges | D_Status |
|-----|----------|---------|------------|-----------|-----------|
| D-4 | 4 | 1003 | 2023-01-18 | 500 | Delivered |

```
MariaDB [store]> select order_id , D_charges from Delivery where D_charges <= 500;
```

| order_id | D_charges |
|----------|-----------|
| 1 | 200 |
| 2 | 100 |
| 3 | 200 |
| 4 | 500 |
| 5 | 200 |
| 6 | 200 |
| 7 | 100 |

7. Logical Operator

A. AND

```
MariaDB [store]> select First_Name , Last_Name, gender,City from Customer where Gender='Male' and City='Mumbai';
```

| First_Name | Last_Name | gender | City |
|------------|-----------|--------|--------|
| John | Done | Male | Mumbai |
| Karan | Kapoor | Male | Mumbai |

B. OR

```
MariaDB [store]> select First_Name , Last_Name, gender,City from Customer where Gender='Male' or City='Thane';
```

| First_Name | Last_Name | gender | City |
|------------|-----------|--------|--------|
| John | Done | Male | Mumbai |
| James | Brown | Male | Thane |
| Hina | Sharma | Female | Thane |
| Karan | Kapoor | Male | Mumbai |

C. NOT

```
MariaDB [store]> select ID, First_Name, Last_Name, Gender  
-> from customer where not gender = 'Male';
```

| ID | First_Name | Last_Name | Gender |
|------|------------|-----------|--------|
| 1002 | Mary | Smith | Female |
| 1004 | Linda | Jonas | Female |
| 1005 | Hina | Sharma | Female |

8. Aggregate Function

A. Count

```
MariaDB [store]> select Count(Price) 'Total Count of Product' from product;
+-----+
| Total Count of Product |
+-----+
|                10     |
+-----+
```

B. Average

```
MariaDB [store]> select Avg(Price) 'Average Product Price' from product;
+-----+
| Average Product Price |
+-----+
|          34298.700000 |
+-----+
1 row in set (0.000 sec)
```

C. Sum

```
MariaDB [store]> select Sum(Price) 'Total Product Price' from product;
+-----+
| Total Product Price |
+-----+
|          342987.00 |
+-----+
```

D. Min

```
MariaDB [store]> select min(Price) 'Minimum Price' from product;
+-----+
| Minimum Price |
+-----+
|          18000.00 |
+-----+
```

E. Max

```
MariaDB [store]> select Max(Price) 'Maximum Price' from product;
+-----+
| Maximum Price |
+-----+
|      79999.00 |
+-----+
```

9. Group By Clause

A. With Aggregate Function

```
MariaDB [Store]> select Gender, count(*) from Customer Group BY Gender;
+-----+-----+
| Gender | count(*) |
+-----+-----+
| Female |         3 |
| Male   |         3 |
+-----+-----+
```

B. With Join

```
MariaDB [store]> Select Catg_Name, Count(Catg_ID) from Product
-> inner join Category
-> on product.Catg_ID = Category.ID group by Catg_ID;
+-----+-----+
| Catg_Name | Count(Catg_ID) |
+-----+-----+
| TV        |         4      |
| Smart_Phones |         4      |
| Refrigerator |         2      |
+-----+-----+
3 rows in set (0.003 sec)
```

C. With Having Clause

```
MariaDB [store]> Select Catg_Name, Count(Catg_ID) as Quantity from Product
-> inner join Category
-> on product.Catg_ID = Category.ID group by Catg_ID
-> Having Count(Catg_ID)=4;
```

| Catg_Name | Quantity |
|--------------|----------|
| TV | 4 |
| Smart_Phones | 4 |

10. Like Operator

```
MariaDB [store]> select Id, Product_Name , Price from Product where Product_name like 'S_s%';
```

| Id | Product_Name | Price |
|-----|-----------------------|----------|
| 101 | Samsung Crystal 4K | 30000.00 |
| 202 | SAMSUNG Galaxy S23 5G | 79999.00 |
| 301 | SAMSUNG 253 L | 24990.00 |
| 302 | SAMSUNG 189 L | 18000.00 |

11. Union

```
MariaDB [store]> select product_name from product
-> union
-> select name from supplier;
```

| product_name |
|---------------------------|
| Samsung Crystal 4K |
| OnePlus U1S |
| Mi X Series |
| OnePlus Y1 |
| POCO X5 Pro |
| SAMSUNG Galaxy S23 5G |
| OnePlus Nord CE 2 Lite 5G |
| OnePLUS 10T |
| SAMSUNG 253 L |
| SAMSUNG 189 L |
| OnePlus_TV |
| REDMI_TV |
| REDMI_SM |
| SAMSUNG_TV Pvt.Ltd. |
| SAMSUNG_SM Pvt.Ltd. |
| SAMSUNG_REF Pvt.Ltd. |

12. Joins

```
MariaDB [store]> select Product_Name, Catg_Name, Price from Product inner Join Category ON product.Catg_ID = Category.ID;
```

| Product_Name | Catg_Name | Price |
|---------------------------|--------------|----------|
| Samsung Crystal 4K | TV | 30000.00 |
| OnePlus U1S | TV | 59000.00 |
| Mi X Series | TV | 37999.00 |
| OnePlus Y1 | TV | 18999.00 |
| POCO X5 Pro | Smart_Phones | 25000.00 |
| SAMSUNG Galaxy S23 5G | Smart_Phones | 79999.00 |
| OnePlus Nord CE 2 Lite 5G | Smart_Phones | 19000.00 |
| OnePLUS 10T | Smart_Phones | 30000.00 |
| SAMSUNG 253 L | Refrigerator | 24990.00 |
| SAMSUNG 189 L | Refrigerator | 18000.00 |

```
MariaDB [store]> select orderID ,Concat(First_Name,' ',Last_Name), O_Date,Product_name,catg_name,totalAmount
-> from order_Tbl o
-> join customer c on o.customerid =c.id
-> join product p on o.productid =p.id
-> join Category c1 on p.Catg_ID = c1.ID;
```

| orderID | Concat(First_Name,' ',Last_Name) | O_Date | Product_name | catg_name | totalAmount |
|---------|----------------------------------|------------|---------------------------|--------------|-------------|
| 4 | James Brown | 2023-01-15 | SAMSUNG 253 L | Refrigerator | 25400 |
| 2 | John Done | 2022-12-24 | SAMSUNG Galaxy S23 5G | Smart_Phones | 80200 |
| 5 | Linda Jonas | 2023-02-01 | POCO X5 Pro | Smart_Phones | 27000 |
| 6 | Hina Sharma | 2023-02-24 | OnePlus Nord CE 2 Lite 5G | Smart_Phones | 38200 |
| 1 | John Done | 2022-12-24 | Samsung Crystal 4K | TV | 30200 |
| 3 | Mary Smith | 2023-01-02 | OnePlus U1S | TV | 59200 |
| 7 | Karan Kapoor | 2023-02-24 | OnePlus Y1 | TV | 20000 |

```
MariaDB [store]> select Concat(First_name,' ',Last_Name), Count(CustomerID) Total_Order, City,Email
-> from Customer
-> Inner Join Order_Tbl
-> on Customer.ID = Order_Tbl.CustomerID
-> Group by ID;
```

| Concat(First_name,' ',Last_Name) | Total_Order | City | Email |
|----------------------------------|-------------|--------|-------------------|
| John Done | 2 | Mumbai | John@gmail.com |
| Mary Smith | 1 | Mumbai | Mary@gmail.com |
| James Brown | 1 | Thane | James@gmail.com |
| Linda Jonas | 1 | Mumbai | Jonas@gmail.com |
| Hina Sharma | 1 | Thane | hinas@gmail.com |
| Karan Kapoor | 1 | Mumbai | Karankk@gmail.com |

13. Subquery

```
MariaDB [store]> select ID, Product_Name, Price from product
-> where price in (select max(price) from product union select min(Price) from Product);
```

| ID | Product_Name | Price |
|-----|-----------------------|----------|
| 202 | SAMSUNG Galaxy S23 5G | 79999.00 |
| 302 | SAMSUNG 189 L | 18000.00 |

14. View

A. Create View

```
MariaDB [store]> Create View VwCust as select ID, First_Name, Gender, City from customer;
Query OK, 0 rows affected (0.009 sec)
```

```
MariaDB [store]> show tables;
```

| Tables_in_store |
|-----------------|
| category |
| customer |
| delivery |
| order_tbl |
| product |
| stock |
| supplier |
| vwcust |

B. Insert Value in view

```
MariaDB [store]> insert into vwcust values(1007, 'Puja', 'Female', 'Navi-Mumbai');
Query OK, 1 row affected, 3 warnings (0.014 sec)
```

C. Update Value in view

```
MariaDB [store]> update vwcust set First_name='Pooja' where ID=1007;  
Query OK, 1 row affected (0.004 sec)
```

D. Delete

```
MariaDB [store]> Delete from Vwcust where id=1007;  
Query OK, 1 row affected (0.011 sec)
```

E. Drop view

```
MariaDB [store]> drop view vwcust;  
Query OK, 0 rows affected (0.001 sec)  
  
MariaDB [store]> show tables;  
+-----+  
| Tables_in_store |  
+-----+  
| category        |  
| customer        |  
| delivery        |  
| order_tbl       |  
| product         |  
| stock           |  
| supplier        |  
+-----+  
7 rows in set (0.001 sec)
```