

### Task:1. Database Design:

1. Create the database named "TechShop"
2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.

```
mysql> create database TechShop;
Query OK, 1 row affected (0.02 sec)

mysql> use TechShop;
Database changed
mysql> CREATE TABLE Customers (
->     CustomerID INT PRIMARY KEY,
->     FirstName VARCHAR(255),
->     LastName VARCHAR(255),
->     Email VARCHAR(255),
->     Phone VARCHAR(15),
->     Address VARCHAR(255)
-> );
Query OK, 0 rows affected (0.11 sec)

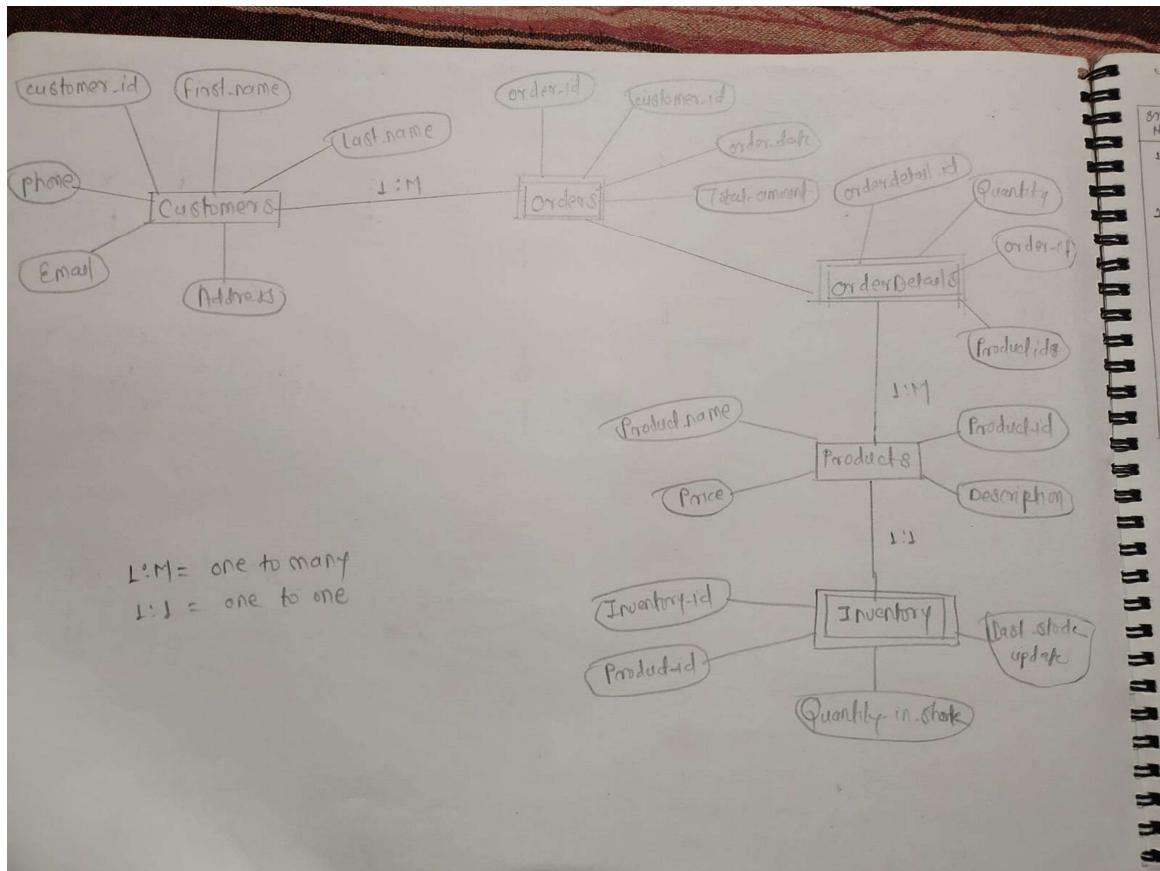
mysql> CREATE TABLE Products (
->     ProductID INT PRIMARY KEY,
->     ProductName VARCHAR(255),
->     Description VARCHAR(255),
->     Price DECIMAL(10, 2)
-> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Orders (
->     OrderID INT PRIMARY KEY,
->     CustomerID INT,
->     OrderDate DATE,
->     TotalAmount DECIMAL(10, 2),
->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
-> );
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE OrderDetails(
->     OrderDetailID INT PRIMARY KEY,
->     OrderID INT,
->     ProductID INT,
->     Qunatity INT,
->     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
-> );
Query OK, 0 rows affected (0.11 sec)
```

```
mysql> CREATE TABLE Inventory (
->     InventoryID INT PRIMARY KEY,
->     ProductID INT,
->     QuantityInStock INT,
->     LastStockUpdate DATE,
->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
-> );
Query OK, 0 rows affected (0.09 sec)
```

**3. Create an ERD (Entity Relationship Diagram) for the database.**



**4. Insert at least 10 sample records into each of the following tables.**

- Customers
- Products
- Orders
- OrderDetails

## Krushnakumar Homraj Patle

```
mysql> INSERT INTO Customers VALUES
-> (1, 'Krishna', 'Patle', 'krishnapatle128@gmail.com', '9325654953', '100 Shivam Nagar'),
-> (2, 'Kashyap', 'Punyawan', 'kpunyawan@gmail.com', '8698456732', '12 Ambe Nagar'),
-> (3, 'Harshal', 'Meshram', 'harshalmeshram@gmail.com', '9115432565', 'Durga Nagar'),
-> (4, 'Shivam', 'Kale', 'shivamkale@gmail.com', '8765432106', 'Manewada'),
-> (5, 'Vikas', 'Nagpure', 'vikasnagpure@gmail.com', '7865943215', 'Ambe Nagar'),
-> (6, 'Nitin', 'Turkar', 'nitinturkar@gmail.com', '8698454795', 'Gruhalaxmi Society'),
-> (7, 'Ruchika', 'Chafekar', 'ruchikachafekar@gmail.com', '7447509656', 'Saoner'),
-> (8, 'Neha', 'Patle', 'nehapatle@gmail.com', '9823485622', 'Shivam Nagar'),
-> (9, 'Pratiksha', 'Katre', 'pratikshakatre@gmail.com', '8475636214', 'Nageshwar Nagar'),
-> (10, 'Shrutika', 'Kolhe', 'shrutkolhe@gmail.com', '8645281931', 'Saoner');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select *from Customers;
+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address |
+-----+-----+-----+-----+-----+
| 1 | Krishna | Patle | krishnapatle128@gmail.com | 9325654953 | 100 Shivam Nagar |
| 2 | Kashyap | Punyawan | kpunyawan@gmail.com | 8698456732 | 12 Ambe Nagar |
| 3 | Harshal | Meshram | harshalmeshram@gmail.com | 9115432565 | Durga Nagar |
| 4 | Shivam | Kale | shivamkale@gmail.com | 8765432106 | Manewada |
| 5 | Vikas | Nagpure | vikasnagpure@gmail.com | 7865943215 | Ambe Nagar |
| 6 | Nitin | Turkar | nitinturkar@gmail.com | 8698454795 | Gruhalaxmi Society |
| 7 | Ruchika | Chafekar | ruchikachafekar@gmail.com | 7447509656 | Saoner |
| 8 | Neha | Patle | nehapatle@gmail.com | 9823485622 | Shivam Nagar |
| 9 | Pratiksha | Katre | pratikshakatre@gmail.com | 8475636214 | Nageshwar Nagar |
| 10 | Shruti | Kolhe | shrutkolhe@gmail.com | 8645281931 | Saoner |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

```
mysql> insert into Products values
-> (1, 'Laptop', 'MI Notebook 14 i5 10th Gen', 40000.00),
-> (2, 'Laptop', 'MI Notebook 14 i5 10th Gen 512 SSD', 42000.00),
-> (3, 'Laptop', 'Dell i5 10th Gen', 56000.00),
-> (4, 'Nothing Smartphone', '8GB RAM 256GB ROM', 30000.00),
-> (5, 'IQ Z6 Lite', '6GB RAM 128GB ROM', 15000.00),
-> (6, 'IQ Neon', '8GB RAM 256GB ROM', 30000.00),
-> (7, 'HP 360', 'i3 10th GEN', 45000.00),
-> (8, 'POCO Mobile', '4GB RAM 64GB ROM', 12000.00),
-> (9, 'Iphone 15 pro', '256GB ROM', 90000.00),
-> (10, 'ASUS Laptop', 'i5 10th GEN', 50000.00);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select *from Products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 1 | Laptop | MI Notebook 14 i5 10th Gen | 40000.00 |
| 2 | Laptop | MI Notebook 14 i5 10th Gen 512 SSD | 42000.00 |
| 3 | Laptop | Dell i5 10th Gen | 56000.00 |
| 4 | Nothing Smartphone | 8GB RAM 256GB ROM | 30000.00 |
| 5 | IQ Z6 Lite | 6GB RAM 128GB ROM | 15000.00 |
| 6 | IQ Neon | 8GB RAM 256GB ROM | 30000.00 |
| 7 | HP 360 | i3 10th GEN | 45000.00 |
| 8 | POCO Mobile | 4GB RAM 64GB ROM | 12000.00 |
| 9 | Iphone 15 pro | 256GB ROM | 90000.00 |
| 10 | ASUS Laptop | i5 10th GEN | 50000.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

```
mysql> insert into Orders values
-> (1,1,'2020-06-15',42000.00),
-> (2,2,'2020-09-25',45000.00),
-> (3,3,'2021-10-20',55000.00),
-> (4,4,'2018-12-01',32000.00),
-> (5,5,'2023-09-23',15000.00),
-> (6,6,'2023-06-26',31000),
-> (7,7,'2022-08-12',45000.00),
-> (8,8,'2022-12-16',12000.00),
-> (9,9,'2022-01-18',91000.00),
-> (10,10,'2023-02-19',51000.00);
```

Query OK, 10 rows affected (0.02 sec)

Records: 10 Duplicates: 0 Warnings: 0

```
mysql> select *from Orders;
```

OrderID	CustomerID	OrderDate	TotalAmount
1	1	2020-06-15	42000.00
2	2	2020-09-25	45000.00
3	3	2021-10-20	55000.00
4	4	2018-12-01	32000.00
5	5	2023-09-23	15000.00
6	6	2023-06-26	31000.00
7	7	2022-08-12	45000.00
8	8	2022-12-16	12000.00
9	9	2022-01-18	91000.00
10	10	2023-02-19	51000.00

10 rows in set (0.00 sec)

Krushnakumar Homraj Patle

```
mysql> insert into OrderDetails values
-> (1,1,1,1)
->
-> (9,2,3,4),
-> (2,1,3,1),
-> (4,2,1,5),
-> (3,4,1,1),
-> (5,7,9,1),
-> (6,10,7,1),
-> (7,5,2,1),
-> (8,10,9,1),
-> (10,7,10,1);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select *from OrderDetails;
+-----+-----+-----+-----+
| OrderDetailID | OrderID | ProductID | Qunatity |
+-----+-----+-----+-----+
|      1 |      1 |        1 |       1 |
|      2 |      1 |        3 |       1 |
|      3 |      4 |        1 |       1 |
|      4 |      2 |        1 |       5 |
|      5 |      7 |        9 |       1 |
|      6 |     10 |        7 |       1 |
|      7 |      5 |        2 |       1 |
|      8 |     10 |        9 |       1 |
|      9 |      2 |        3 |       4 |
|     10 |      7 |       10 |       1 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

```
mysql> insert into Inventory values
-> (1,1,10,'2023-10-12'),
-> (2,2,15,'2023-01-02'),
-> (3,3,9,'2022-12-09'),
-> (4,4,8,'2023-10-05'),
-> (5,5,12,'2022-11-20'),
-> (6,6,23,'2023-02-09'),
-> (7,7,30,'2023-03-10'),
-> (8,8,25,'2022-10-19'),
-> (9,9,5,'2023-05-04'),
-> (10,10,10,'2023-10-10');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select *from Inventory;
+-----+-----+-----+-----+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-----+-----+-----+-----+
|      1 |        1 |          10 | 2023-10-12 |
|      2 |        2 |          15 | 2023-01-02 |
|      3 |        3 |           9 | 2022-12-09 |
|      4 |        4 |           8 | 2023-10-05 |
|      5 |        5 |          12 | 2022-11-20 |
|      6 |        6 |          23 | 2023-02-09 |
|      7 |        7 |          30 | 2023-03-10 |
|      8 |        8 |          25 | 2022-10-19 |
|      9 |        9 |           5 | 2023-05-04 |
|     10 |       10 |          10 | 2023-10-10 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

**Task 2:**

- 1. Write an SQL query to retrieve the names and emails of all customers.**

```
mysql> select FirstName,LastName,Email from Customers;
+-----+-----+-----+
| FirstName | LastName | Email
+-----+-----+-----+
| Krishna   | Patle    | krishnapatle128@gmail.com |
| Kashyap   | Punyawan  | kpunyawan@gmail.com      |
| Harshal   | Meshram   | harshalmeshram@gmail.com |
| Shivam    | Kale      | shivamkale@gmail.com     |
| Vikas     | Nagpure   | vikasnagpure@gmail.com   |
| Nitin     | Turkar    | nitinturkar@gmail.com    |
| Ruchika   | Chafekar  | ruchikachafekar@gmail.com |
| Neha      | Patle    | nehapatle@gmail.com      |
| Pratiksha | Katre    | pratikshakatre@gmail.com |
| ShrutiKa  | Kolhe    | shrutkolhe@gmail.com     |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

- 2. Write an SQL query to list all orders with their order dates and corresponding customer names.**

```
mysql> select Orders.OrderID, Orders.OrderDate,Customers.FirstName,Customers.LastName
-> from Orders
-> join Customers on Orders.CustomerID=Customers.CustomerID;
+-----+-----+-----+-----+
| OrderID | OrderDate | FirstName | LastName
+-----+-----+-----+-----+
| 1 | 2020-06-15 | Krishna | Patle |
| 2 | 2020-09-25 | Kashyap | Punyawan |
| 3 | 2021-10-20 | Harshal | Meshram |
| 4 | 2018-12-01 | Shivam | Kale |
| 5 | 2023-09-23 | Vikas | Nagpure |
| 6 | 2023-06-26 | Nitin | Turkar |
| 7 | 2022-08-12 | Ruchika | Chafekar |
| 8 | 2022-12-16 | Neha | Patle |
| 9 | 2022-01-18 | Pratiksha | Katre |
| 10 | 2023-02-19 | ShrutiKa | Kolhe |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

- 3. Write an SQL query to insert a new customer record into the “Customers” table. Include customer information such as name, email and address.**

```
mysql> insert into Customers(CustomerID,FirstName,LastName,Email,Address)
-> values(11,'Masud','Ansari','masudansari@gmail.com','87 Digori');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select *from Customers;
+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address |
+-----+-----+-----+-----+-----+
| 1 | Krishna | Patle | krishnapatle128@gmail.com | 9325654953 | 100 Shivam Nagar |
| 2 | Kashyap | Punyawan | kpunyawan@gmail.com | 8698456732 | 12 Ambe Nagar |
| 3 | Harshal | Meshram | harshalmeshram@gmail.com | 9115432565 | Durga Nagar |
| 4 | Shivam | Kale | shivamkale@gmail.com | 8765432106 | Manewada |
| 5 | Vikas | Nagpure | vikasnagpure@gmail.com | 7865943215 | Ambe Nagar |
| 6 | Nitin | Turkar | nitinturkar@gmail.com | 8698454795 | Gruhalaxmi Society |
| 7 | Ruchika | Chafekar | ruchikachafekar@gmail.com | 7447509656 | Saoner |
| 8 | Neha | Patle | nehapatle@gmail.com | 9823485622 | Shivam Nagar |
| 9 | Pratiksha | Katre | pratikshakatre@gmail.com | 8475636214 | Nageshwar Nagar |
| 10 | Shruti | Kolhe | shrutkolhe@gmail.com | 8645281931 | Saoner |
| 11 | Masud | Ansari | masudansari@gmail.com | NULL | 87 Digori |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

#### 4. Write an SQL query to update the prices of all electronics gadgets in the “Products” table by increasing them by 10%.

```
mysql> update Products set Price = Price*1.1;
Query OK, 10 rows affected (0.04 sec)
Rows matched: 10  Changed: 10  Warnings: 0
```

```
mysql> select * from Products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 1 | Laptop | MI Notebook 14 i5 10th Gen | 44000.00 |
| 2 | Laptop | MI Notebook 14 i5 10th Gen 512 SSD | 46200.00 |
| 3 | Laptop | Dell i5 10th Gen | 61600.00 |
| 4 | Nothing Smartphone | 8GB RAM 256GB ROM | 33000.00 |
| 5 | IQ Z6 Lite | 6GB RAM 128GB ROM | 16500.00 |
| 6 | IQ Neon | 8GB RAM 256GB ROM | 33000.00 |
| 7 | HP 360 | i3 10th GEN | 49500.00 |
| 8 | POCO Mobile | 4GB RAM 64GB ROM | 13200.00 |
| 9 | Iphone 15 pro | 256GB ROM | 99000.00 |
| 10 | ASUS Laptop | i5 10th GEN | 55000.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

#### 5. Write an SQL query to delete a specific order and its associated order details from the “Orders” and “OrderDetails” table. Allow users to input the order ID as a parameter.

```
mysql> set @OrderIDToDelete=5;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> delete from OrderDetails
-> where OrderID=@OrderIDToDelete;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> delete from Orders
-> where OrderID=@OrderIDToDelete;
Query OK, 1 row affected (0.01 sec)
```

#### 6. Write an SQL query to insert a new order into the “Orders” table. Include the customers ID, order date, and any other necessary information.

```
mysql> insert into Orders(OrderID,CustomerID,OrderDate,TotalAmount)
-> values(5,5,'2023-12-09',120000);
Query OK, 1 row affected (0.03 sec)

mysql> select * from Orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
| 1 | 1 | 2020-06-15 | 42000.00 |
| 2 | 2 | 2020-09-25 | 45000.00 |
| 3 | 3 | 2021-10-20 | 55000.00 |
| 4 | 4 | 2018-12-01 | 32000.00 |
| 5 | 5 | 2023-12-09 | 120000.00 |
| 6 | 6 | 2023-06-26 | 31000.00 |
| 7 | 7 | 2022-08-12 | 45000.00 |
| 8 | 8 | 2022-12-16 | 12000.00 |
| 9 | 9 | 2022-01-18 | 91000.00 |
| 10 | 10 | 2023-02-19 | 51000.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

- 7. Write an SQL query to update the contact information (eg, email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.**

```
mysql> set @CustomerID =4;
Query OK, 0 rows affected (0.00 sec)

mysql> update Customers
-> set Email='shivamkale12@gmail.com',Address='123 Manewada'
-> where CustomerID=@CustomerID;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Customers;
+-----+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address |
+-----+-----+-----+-----+-----+-----+
| 1 | Krishna | Patle | krishnapatle128@gmail.com | 9325654953 | 100 Shivam Nagar |
| 2 | Kashyap | Punyawan | kpunyawan@gmail.com | 8698456732 | 12 Ambe Nagar |
| 3 | Harshal | Meshram | harshalmeshram@gmail.com | 9115432565 | Durga Nagar |
| 4 | Shivam | Kale | shivamkale12@gmail.com | 8765432106 | 123 Manewada |
| 5 | Vikas | Nagpure | vikasnagpure@gmail.com | 7865943215 | Ambe Nagar |
| 6 | Nitin | Turkar | nitinturkar@gmail.com | 8698454795 | Gruhalaxmi Society |
| 7 | Ruchika | Chafekar | ruchikachafekar@gmail.com | 7447509656 | Saoner |
| 8 | Neha | Patle | nehapatle@gmail.com | 9823485622 | Shivam Nagar |
| 9 | Pratiksha | Katre | pratikshakatre@gmail.com | 8475636214 | Nageshwar Nagar |
| 10 | Shruti | Kolhe | shrutkolhe@gmail.com | 8645281931 | Saoner |
| 11 | Masud | Ansari | masudansari@gmail.com | NULL | 87 Digori |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

- 8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.**

```
mysql> update Orders
-> set TotalAmount=
-> select sum(Products.Price * OrderDetails.Qunatity)
-> from OrderDetails
-> join Products on OrderDetails.ProductID=Products.ProductID
-> where OrderDetails.OrderID=Orders.OrderID
-> );
Query OK, 10 rows affected (0.01 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select *from Orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
| 1 | 1 | 2020-06-15 | 105600.00 |
| 2 | 2 | 2020-09-25 | 466400.00 |
| 3 | 3 | 2021-10-20 | NULL |
| 4 | 4 | 2018-12-01 | 44000.00 |
| 5 | 5 | 2023-12-09 | NULL |
| 6 | 6 | 2023-06-26 | NULL |
| 7 | 7 | 2022-08-12 | 154000.00 |
| 8 | 8 | 2022-12-16 | NULL |
| 9 | 9 | 2022-01-18 | NULL |
| 10 | 10 | 2023-02-19 | 148500.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

**9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.**

```
mysql> set @customerID =5;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from OrderDetails where OrderID in(
-> select OrderID from Orders where CustomerID=@customerID);
Query OK, 0 rows affected (0.00 sec)

mysql> delete from Orders where CustomerID=@customerID;
Query OK, 1 row affected (0.01 sec)

mysql> select * from Orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
| 1 | 1 | 2020-06-15 | 105600.00 |
| 2 | 2 | 2020-09-25 | 466400.00 |
| 3 | 3 | 2021-10-20 | NULL |
| 4 | 4 | 2018-12-01 | 44000.00 |
| 6 | 6 | 2023-06-26 | NULL |
| 7 | 7 | 2022-08-12 | 154000.00 |
| 8 | 8 | 2022-12-16 | NULL |
| 9 | 9 | 2022-01-18 | NULL |
| 10 | 10 | 2023-02-19 | 148500.00 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

**10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.**

```
mysql> INSERT INTO Products (ProductID,ProductName, Description, Price)
-> values (11,'Noise SmartWatch','100+ Sports Mode',2000);
Query OK, 1 row affected (0.01 sec)

mysql> select * from Products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 1 | Laptop | MI Notebook 14 i5 10th Gen | 44000.00 |
| 2 | Laptop | MI Notebook 14 i5 10th Gen 512 SSD | 46200.00 |
| 3 | Laptop | Dell i5 10th Gen | 61600.00 |
| 4 | Nothing Smartphone | 8GB RAM 256GB ROM | 33000.00 |
| 5 | IQ Z6 Lite | 6GB RAM 128GB ROM | 16500.00 |
| 6 | IQ Neon | 8GB RAM 256GB ROM | 33000.00 |
| 7 | HP 360 | i3 10th GEN | 49500.00 |
| 8 | POCO Mobile | 4GB RAM 64GB ROM | 13200.00 |
| 9 | Iphone 15 pro | 256GB ROM | 99000.00 |
| 10 | ASUS Laptop | i5 10th GEN | 55000.00 |
| 11 | Noise SmartWatch | 100+ Sports Mode | 2000.00 |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

**11.** Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.

```
mysql> alter table Orders
-> add column OrderStatus varchar(255);
Query OK, 0 rows affected (0.18 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from Orders;
+-----+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount | OrderStatus |
+-----+-----+-----+-----+-----+
| 1 | 1 | 2020-06-15 | 105600.00 | NULL |
| 2 | 2 | 2020-09-25 | 466400.00 | NULL |
| 3 | 3 | 2021-10-20 | NULL | NULL |
| 4 | 4 | 2018-12-01 | 44000.00 | NULL |
| 6 | 6 | 2023-06-26 | NULL | NULL |
| 7 | 7 | 2022-08-12 | 154000.00 | NULL |
| 8 | 8 | 2022-12-16 | NULL | NULL |
| 9 | 9 | 2022-01-18 | NULL | NULL |
| 10 | 10 | 2023-02-19 | 148500.00 | NULL |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql> update Orders set OrderStatus="Shipped" where OrderID=6;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Orders;
+-----+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount | OrderStatus |
+-----+-----+-----+-----+-----+
| 1 | 1 | 2020-06-15 | 105600.00 | NULL |
| 2 | 2 | 2020-09-25 | 466400.00 | NULL |
| 3 | 3 | 2021-10-20 | NULL | NULL |
| 4 | 4 | 2018-12-01 | 44000.00 | NULL |
| 6 | 6 | 2023-06-26 | NULL | Shipped |
| 7 | 7 | 2022-08-12 | 154000.00 | NULL |
| 8 | 8 | 2022-12-16 | NULL | NULL |
| 9 | 9 | 2022-01-18 | NULL | NULL |
| 10 | 10 | 2023-02-19 | 148500.00 | NULL |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

**12.** Write an SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table.

```
mysql> UPDATE Customers
-> SET TotalOrders = (
->     SELECT COUNT(*)
->     FROM Orders
->     WHERE Orders.CustomerID = Customers.CustomerID
-> );
Query OK, 11 rows affected (0.01 sec)
Rows matched: 11  Changed: 11  Warnings: 0

mysql> select * from Customers;
+-----+-----+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address | TotalOrders |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Krishna | Patle | krishnapatle128@gmail.com | 9325654953 | 100 Shivam Nagar | 1 |
| 2 | Kashyap | Punyawan | kpunyawan@gmail.com | 8698456732 | 12 Ambe Nagar | 1 |
| 3 | Harshal | Meshram | harshalmeshram@gmail.com | 9115432565 | Durga Nagar | 1 |
| 4 | Shivam | Kale | shivamkale12@gmail.com | 8765432106 | 123 Manewada | 1 |
| 5 | Vikas | Nagpure | vikasnagpure@gmail.com | 7865943215 | Ambe Nagar | 0 |
| 6 | Nitin | Turkar | nitinturkar@gmail.com | 8698454795 | Gruhalaxmi Society | 1 |
| 7 | Ruchika | Chafekar | ruchikachafekar@gmail.com | 7447509656 | Saoner | 1 |
| 8 | Neha | Patle | nehapatle@gmail.com | 9823485622 | Shivam Nagar | 1 |
| 9 | Pratiksha | Katre | pratikshakatre@gmail.com | 8475636214 | Nageshwar Nagar | 1 |
| 10 | Shruti | Kolhe | shrutkolhe@gmail.com | 8645281931 | Saoner | 1 |
| 11 | Masud | Ansari | masudansari@gmail.com | NULL | 87 Digori | 0 |
+-----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

### Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.

```
mysql> select Orders.OrderID, Customers.CustomerID,Customers.FirstName,Customers.LastName,Orders.OrderDate,Orders.TotalAmount
-> from Orders
-> join Customers on Orders.CustomerID=Customers.CustomerID;
+-----+-----+-----+-----+-----+-----+
| OrderID | CustomerID | FirstName | LastName | OrderDate | TotalAmount |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Krishna | Patle | 2020-06-15 | 105600.00 |
| 2 | 2 | Kashyap | Punyawan | 2020-09-25 | 466400.00 |
| 3 | 3 | Harshal | Meshram | 2021-10-20 | NULL |
| 4 | 4 | Shivam | Kale | 2018-12-01 | 44000.00 |
| 6 | 6 | Nitin | Turkar | 2023-06-26 | NULL |
| 7 | 7 | Ruchika | Chafekar | 2022-08-12 | 154000.00 |
| 8 | 8 | Neha | Patle | 2022-12-16 | NULL |
| 9 | 9 | Pratiksha | Katre | 2022-01-18 | NULL |
| 10 | 10 | Shruti | Kolhe | 2023-02-19 | 148500.00 |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.

```
mysql> select Products.ProductName, sum(OrderDetails.Qunatity*Products.Price) as TotalRevenue
-> from OrderDetails
-> join Products on OrderDetails.ProductID=Products.ProductID
-> group by Products.productName;
+-----+-----+
| ProductName | TotalRevenue |
+-----+-----+
| Laptop | 616000.00 |
| Iphone 15 pro | 198000.00 |
| HP 360 | 49500.00 |
| ASUS Laptop | 55000.00 |
+-----+-----+
4 rows in set (0.03 sec)
```

**3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.**

```
mysql> SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName, Customers.Email, Customers.Phone
-> FROM Customers
-> JOIN Orders ON Customers.CustomerID=Orders.CustomerID
-> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName, Customers.Email, Customers.Phone;
+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email           | Phone      |
+-----+-----+-----+-----+-----+
| 1 | Krishna | Patle   | krishnapatle128@gmail.com | 9325654953 |
| 2 | Kashyap  | Punyawan | kpunyawan@gmail.com     | 8698456732 |
| 3 | Harshal   | Meshram  | harshalmeshram@gmail.com | 9115432565 |
| 4 | Shivam    | Kale     | shivamkale12@gmail.com  | 8765432106 |
| 6 | Nitin     | Turkar   | nitinturkar@gmail.com   | 8698454795 |
| 7 | Ruchika   | Chafekar | ruchikachafekar@gmail.com | 7447509656 |
| 8 | Neha      | Patle   | nehapatle@gmail.com     | 9823485622 |
| 9 | Pratiksha | Katre   | pratikshakatre@gmail.com | 8475636214 |
| 10 | ShrutiKa | Kolhe   | shrutkolhe@gmail.com    | 8645281931 |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

**4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.**

```
mysql> SELECT Products.ProductName, SUM(OrderDetails.Qunatity) AS TotalQuantityOrdered
-> FROM OrderDetails
-> JOIN Products ON OrderDetails.ProductID=Products.ProductID
-> GROUP BY Products.ProductName
-> ORDER BY TotalQuantityOrdered DESC
-> LIMIT 1;
+-----+-----+
| ProductName | TotalQuantityOrdered |
+-----+-----+
| Laptop      |          12 |
+-----+-----+
1 row in set (0.00 sec)
```

**5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.**

```
mysql> SELECT Products.ProductName from Products;
+-----+
| ProductName |
+-----+
| Laptop      |
| Laptop      |
| Laptop      |
| Nothing     |
| Smartphone  |
| IQ Z6 Lite  |
| IQ Neon     |
| HP 360      |
| POCO Mobile |
| Iphone 15 pro|
| ASUS Laptop  |
| Noise SmartWatch |
+-----+
11 rows in set (0.00 sec)
```

**6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.**

```
mysql> SELECT Customers.CustomerID, Customers.FirstName,Customers.LastName, AVG(Orders.TotalAmount) AS AverageOrderValue
-> FROM Customers
-> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
-> GROUP BY Customers.CustomerID, Customers.FirstName,Customers.LastName;
+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | AverageOrderValue |
+-----+-----+-----+-----+
| 1 | Krishna | Patle | 105600.000000 |
| 2 | Kashyap | Punyawan | 466400.000000 |
| 3 | Harshal | Meshram | NULL |
| 4 | Shivam | Kale | 44000.000000 |
| 6 | Nitin | Turkar | NULL |
| 7 | Ruchika | Chafekar | 154000.000000 |
| 8 | Neha | Patle | NULL |
| 9 | Pratiksha | Katre | NULL |
| 10 | Shrutiika | Kolhe | 148500.000000 |
+-----+-----+-----+-----+
9 rows in set (0.01 sec)
```

- 7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.**

```
mysql> SELECT Orders.OrderID,Customers.FirstName,Customers.LastName, Orders.TotalAmount
-> FROM Orders
-> JOIN Customers ON Orders.CustomerID = Customers.CustomerID
-> ORDER BY Orders.TotalAmount DESC
-> LIMIT 1;
+-----+-----+-----+-----+
| OrderID | FirstName | LastName | TotalAmount |
+-----+-----+-----+-----+
| 2 | Kashyap | Punyawan | 466400.00 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- 8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.**

```
mysql> SELECT Products.ProductName, COUNT(OrderDetails.OrderID) AS OrderCount
-> FROM Products
-> LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID
-> GROUP BY Products.ProductName;
+-----+-----+
| ProductName | OrderCount |
+-----+-----+
| Laptop | 5 |
| Nothing Smartphone | 0 |
| IQ Z6 Lite | 0 |
| IQ Neon | 0 |
| HP 360 | 1 |
| POCO Mobile | 0 |
| Iphone 15 pro | 2 |
| ASUS Laptop | 1 |
| Noise SmartWatch | 0 |
+-----+-----+
9 rows in set (0.00 sec)
```

- 9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.**

```
mysql> SELECT Customers.CustomerID, Customers.FirstName,Customers.LastName
-> FROM Customers
-> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
-> JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
-> JOIN Products ON OrderDetails.ProductID = Products.ProductID
-> WHERE Products.ProductName ="Laptop";
+-----+-----+-----+
| CustomerID | FirstName | LastName |
+-----+-----+-----+
| 1 | Krishna | Patle |
| 4 | Shivam | Kale |
| 2 | Kashyap | Punyawan |
| 1 | Krishna | Patle |
| 2 | Kashyap | Punyawan |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

**10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.**

```
mysql> SELECT SUM(TotalAmount) AS TotalRevenue
-> FROM Orders
-> WHERE OrderDate BETWEEN '2018-1-1' AND '2023-10-30' ;
+-----+
| TotalRevenue |
+-----+
| 918500.00 |
+-----+
1 row in set (0.00 sec)
```

#### Task 4: Subqueries:

**1. Write an SQL query to find out which customers have not placed any orders.**

```
406 •   SELECT customerid, firstname, lastname
407     FROM customers
408    WHERE customerid NOT IN (SELECT DISTINCT customerid FROM orders);
409
```

	customerid	firstname	lastname
▶	5	Vikas	Nagpure
	11	Masud	Ansari

**2. Write an SQL query to find the total number of products available for sale.**

```
410 •   SELECT COUNT(*) AS total_products  
411      FROM products;  
412
```

Result Grid	
<a href="#">Filter Rows:</a>	<input type="text"/>
total_products	11

**3. Write an SQL query to calculate the total revenue generated by TechShop.**

```
413 •   SELECT SUM(totalamount) AS total_revenue  
414      FROM orders;
```

Result Grid	
<a href="#">Filter Rows:</a>	<input type="text"/>
total_revenue	918500.00

**4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.**

```
416 •   SELECT AVG(qunatity) AS average_quantity_ordered  
417      FROM orderdetails  
418     WHERE productid IN (SELECT productid FROM products WHERE productname = 'Laptop');
```

Result Grid	
<a href="#">Filter Rows:</a>	<input type="text"/>
average_quantity_ordered	2.4000

**5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.**

```
420 •   SELECT customerid,  
421     (SELECT firstname FROM customers WHERE customerid = orders.customerid) AS firstname,  
422     (SELECT lastname FROM customers WHERE customerid = orders.customerid) AS lastname, SUM(totalamount) AS total_spending  
423     FROM orders  
424    WHERE customerid = 2  
425    GROUP BY customerid;
```

Result Grid	
<a href="#">Filter Rows:</a>	<input type="text"/>
customerid	firstname
2	Kashyap
	lastname
	Punyawan
	total_spending
	466400.00

**6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.**

```
427 •  SELECT customerid,
428      (SELECT firstname FROM customers WHERE customerid = orders.customerid) AS firstname,
429      (SELECT lastname FROM customers WHERE customerid = orders.customerid) AS lastname,
430      COUNT(orderid) AS order_count
431  FROM orders
432  GROUP BY customerid, firstname, lastname
433  ORDER BY order_count DESC
434  LIMIT 1;
```

Result Grid				
	customerid	firstname	lastname	order_count
▶	1	Krishna	Patle	1

**7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.**

```
436 •  SELECT productname , SUM(qunatity) AS total_quantity_ordered
437  FROM orderdetails
438  JOIN products ON orderdetails.productid = products.productid
439  GROUP BY productname
440  ORDER BY total_quantity_ordered DESC
441  LIMIT 1;
```

Result Grid		
	productname	total_quantity_ordered
▶	Laptop	12

**8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.**

```
443 •  SELECT customerid,
444      (SELECT firstname FROM customers WHERE customerid = orders.customerid) AS firstname,
445      (SELECT lastname FROM customers WHERE customerid = orders.customerid) AS lastname,
446      SUM(totalamount) AS total_spending
447  FROM orders
448  WHERE customerid IN (SELECT DISTINCT customerid FROM orderdetails WHERE productid IN
449  (SELECT productid FROM products WHERE productname = 'laptop'))
450  GROUP BY customerid, firstname, lastname
451  ORDER BY total_spending DESC
452  LIMIT 1;
```

Result Grid				
	customerid	firstname	lastname	total_spending
▶	2	Kashyap	Punyawan	466400.00

**9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.**

```
454 •   SELECT AVG(totalamount) AS average_order_value  
455      FROM orders;  
456
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
▶ average_order_value				183700.000000