

## Assignment 1 SQL

By- Kunal Sudhir Mishra

### Task:1. Database Design:

1. Create the database named "TechShop".

```
mysql> create database TechShop1;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> use TechShop1;
Database changed
```

2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.

```
mysql> CREATE TABLE Customers (
    -> CustomerID INT PRIMARY KEY,
    -> FirstName VARCHAR(50),
    -> LastName VARCHAR(50),
    -> Email VARCHAR(100),
    -> Phone VARCHAR(20),
    -> Address VARCHAR(100));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE Products (
    -> ProductID INT PRIMARY KEY,
    -> ProductName VARCHAR(100),
    -> Description VARCHAR(500),
    -> Price DECIMAL(10, 2));
Query OK, 0 rows affected (0.03 sec)
```

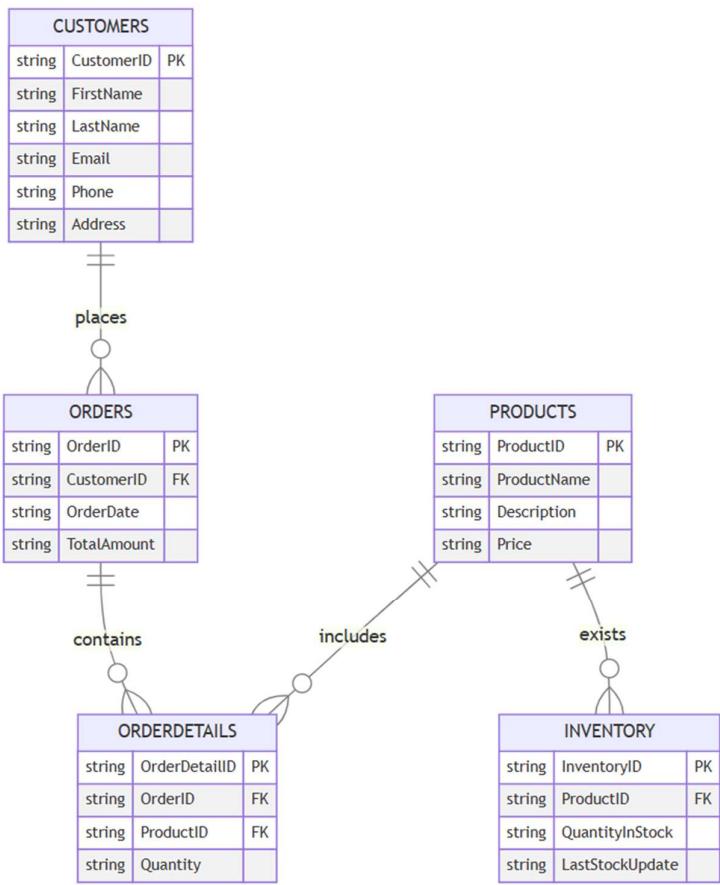
```
mysql> CREATE TABLE Orders (
    -> OrderID INT PRIMARY KEY,
    -> CustomerID INT,
    -> OrderDate DATE,
    -> TotalAmount DECIMAL(10, 2),
    -> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE OrderDetails (
    ->   OrderDetailID INT PRIMARY KEY,
    ->   OrderID INT,
    ->   ProductID INT,
    ->   Quantity INT,
    ->   FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    ->   FOREIGN KEY (ProductID) REFERENCES Products(ProductID));
Query OK, 0 rows affected (0.04 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 (0.03 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_techshop1 |
+-----+
| customers           |
| inventory          |
| orderdetails        |
| orders              |
| products            |
+-----+
5 rows in set (0.01 sec)
```

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



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

```
mysql> desc Customers;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CustomerID | int | NO | PRI | NULL |
| FirstName | varchar(50) | YES | | NULL |
| LastName | varchar(50) | YES | | NULL |
| Email | varchar(100) | YES | | NULL |
| Phone | varchar(20) | YES | | NULL |
| Address | varchar(100) | YES | | NULL |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> desc Inventory;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| InventoryID | int | NO | PRI | NULL |
| ProductID | int | YES | MUL | NULL |
| QuantityInStock | int | YES | | NULL |
| LastStockUpdate | date | YES | | NULL |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> desc OrderDetails;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| OrderDetailID | int | NO | PRI | NULL | 
| OrderID | int | YES | MUL | NULL | 
| ProductID | int | YES | MUL | NULL | 
| Quantity | int | YES | | NULL | 
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc Orders;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| OrderID | int | NO | PRI | NULL | 
| CustomerID | int | YES | MUL | NULL | 
| OrderDate | date | YES | | NULL | 
| TotalAmount | decimal(10,2) | YES | | NULL | 
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc Products;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ProductID | int | NO | PRI | NULL | 
| ProductName | varchar(100) | YES | | NULL | 
| Description | varchar(500) | YES | | NULL | 
| Price | decimal(10,2) | YES | | NULL | 
+-----+-----+-----+-----+-----+
```

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

**a. Customers**

```
mysql> INSERT INTO
->   Customers (
->     CustomerID,
->     FirstName,
->     LastName,
->     Email,
->     Phone,
->     Address
->   )
-> VALUES
->   (
->     1,
->     'Rahul',
->     'Sharma',
->     'rahul@gmail.com',
->     '9876543210',
->     '123 ABC Street'
->   ),
->   (
->     2,
->     'Amit',
->     'Patel',
->     'amit@gmail.com',
->     '9876543211',
->     '456 XYZ Street'
->   ),
```

```
-->      3,
-->      'Priya',
-->      'Singh',
-->      'priya@gmail.com',
-->      '9876543212',
-->      '789 PQR Street'
-->  ),
-->  (
-->    4,
-->    'Rajesh',
-->    'Gupta',
-->    'rajesh@gmail.com',
-->    '9876543213',
-->    '321 LMN Street'
-->  ),
-->  (
-->    5,
-->    'Sneha',
-->    'Verma',
-->    'sneha@gmail.com',
-->    '9876543214',
-->    '654 JKL Street'
-->  ),
-->  (
-->    6,
-->    'Vikram',
-->    'Yadav',
-->    'vikram@gmail.com',
-->    '9876543215',
-->    '987 RST Street'
-->  ),
```

```

-->      7,
-->      'Neha',
-->      'Shah',
-->      'neha@gmail.com',
-->      '9876543216',
-->      '654 UVW Street'
-->  ),
-->  (
-->      8,
-->      'Rohit',
-->      'Mehta',
-->      'rohit@gmail.com',
-->      '9876543217',
-->      '321 GHI Street'
-->  ),
-->  (
-->      9,
-->      'Anita',
-->      'Jain',
-->      'anita@gmail.com',
-->      '9876543218',
-->      '789 DEF Street'
-->  ),
-->  (
-->      10,
-->      'Sanjay',
-->      'Kumar',
-->      'sanjay@gmail.com',
-->      '9876543219',
-->      '123 MNO Street'
-->  );
Query OK, 10 rows affected (0.01 sec)

```

CustomerID	FirstName	LastName	Email	Phone	Address
1	Rahul	Sharma	rahul@gmail.com	9876543210	123 ABC Street
2	Amit	Patel	amit@gmail.com	9876543211	456 XYZ Street
3	Priya	Singh	priya@gmail.com	9876543212	789 PQR Street
4	Rajesh	Gupta	rajesh@gmail.com	9876543213	321 LMN Street
5	Sneha	Verma	sneha@gmail.com	9876543214	654 JKL Street
6	Vikram	Yadav	vikram@gmail.com	9876543215	987 RST Street
7	Neha	Shah	neha@gmail.com	9876543216	654 UVW Street
8	Rohit	Mehta	rohit@gmail.com	9876543217	321 GHI Street
9	Anita	Jain	anita@gmail.com	9876543218	789 DEF Street
10	Sanjay	Kumar	sanjay@gmail.com	9876543219	123 MNO Street

## b. Products

```
mysql> INSERT INTO
->   Products (ProductID, ProductName, Description, Price)
-> VALUES
->   (
->     1,
->     'Mobile Phone',
->     'Smartphone with 6GB RAM',
->     15000
->   ),
->   (
->     2,
->     'Laptop',
->     '15.6 inch, 8GB RAM, 1TB HDD',
->     40000
->   ),
->   (
->     3,
->     'Headphones',
->     'Wireless Bluetooth headphones',
->     2000
->   ),
->   (
->     4,
->     'Smartwatch',
->     'Fitness tracker with heart rate monitor',
->     5000
->   ),
->   (5, 'Camera', 'DSLR camera with 24MP', 30000),
->   (6, 'Television', '55 inch LED TV', 50000),
->   (7, 'Tablet', '10 inch Android tablet', 10000),
->   (8, 'Gaming Console', 'PlayStation 5', 40000),
```

```
->   (8, 'Gaming Console', 'PlayStation 5', 40000),
->   (
->     9,
->     'Speakers',
->     '2.1 channel multimedia speakers',
->     3000
->   ),
->   (10, 'Printer', 'All-in-one printer', 5000);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> |
```

```

mysql> SELECT * FROM Products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 1 | Mobile Phone | Smartphone with 6GB RAM | 15000.00 |
| 2 | Laptop | 15.6 inch, 8GB RAM, 1TB HDD | 40000.00 |
| 3 | Headphones | Wireless Bluetooth headphones | 2000.00 |
| 4 | Smartwatch | Fitness tracker with heart rate monitor | 5000.00 |
| 5 | Camera | DSLR camera with 24MP | 30000.00 |
| 6 | Television | 55 inch LED TV | 50000.00 |
| 7 | Tablet | 10 inch Android tablet | 10000.00 |
| 8 | Gaming Console | PlayStation 5 | 40000.00 |
| 9 | Speakers | 2.1 channel multimedia speakers | 3000.00 |
| 10 | Printer | All-in-one printer | 5000.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

```

### c. Orders

```

mysql> INSERT INTO
->   Orders (OrderID, CustomerID, OrderDate, TotalAmount)
-> VALUES
->   (1, 1, '2022-01-01', 20000),
->   (2, 2, '2022-01-02', 45000),
->   (3, 3, '2022-01-03', 10000),
->   (4, 4, '2022-01-04', 35000),
->   (5, 5, '2022-01-05', 25000),
->   (6, 6, '2022-01-06', 15000),
->   (7, 7, '2022-01-07', 30000),
->   (8, 8, '2022-01-08', 50000),
->   (9, 9, '2022-01-09', 40000),
->   (10, 10, '2022-01-10', 60000);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

```

```
mysql> SELECT * FROM Orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
|      1 |         1 | 2022-01-01 |    20000.00 |
|      2 |         2 | 2022-01-02 |    45000.00 |
|      3 |         3 | 2022-01-03 |    10000.00 |
|      4 |         4 | 2022-01-04 |    35000.00 |
|      5 |         5 | 2022-01-05 |    25000.00 |
|      6 |         6 | 2022-01-06 |    15000.00 |
|      7 |         7 | 2022-01-07 |    30000.00 |
|      8 |         8 | 2022-01-08 |    50000.00 |
|      9 |         9 | 2022-01-09 |    40000.00 |
|     10 |        10 | 2022-01-10 |    60000.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

#### d. OrderDetails

```
mysql> INSERT INTO
->   OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
-> VALUES
->   (1, 1, 1, 2),
->   (2, 2, 2, 1),
->   (3, 3, 3, 3),
->   (4, 4, 4, 2),
->   (5, 5, 5, 1),
->   (6, 6, 6, 2),
->   (7, 7, 7, 1),
->   (8, 8, 8, 1),
->   (9, 9, 9, 2),
->   (10, 10, 10, 3);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```

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

```

#### e. Inventory

```

mysql> INSERT INTO
    >   Inventory (
    >     InventoryID,
    >     ProductID,
    >     QuantityInStock,
    >     LastStockUpdate
    >   )
    > VALUES
    >   (1, 1, 10, '2022-01-01'),
    >   (2, 2, 5, '2022-01-02'),
    >   (3, 3, 20, '2022-01-03'),
    >   (4, 4, 8, '2022-01-04'),
    >   (5, 5, 15, '2022-01-05'),
    >   (6, 6, 12, '2022-01-06'),
    >   (7, 7, 7, '2022-01-07'),
    >   (8, 8, 3, '2022-01-08'),
    >   (9, 9, 10, '2022-01-09'),
    >   (10, 10, 6, '2022-01-10');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

```

```
mysql> SELECT * FROM Inventory;
+-----+-----+-----+-----+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-----+-----+-----+-----+
| 1 | 1 | 10 | 2022-01-01 |
| 2 | 2 | 5 | 2022-01-02 |
| 3 | 3 | 20 | 2022-01-03 |
| 4 | 4 | 8 | 2022-01-04 |
| 5 | 5 | 15 | 2022-01-05 |
| 6 | 6 | 12 | 2022-01-06 |
| 7 | 7 | 7 | 2022-01-07 |
| 8 | 8 | 3 | 2022-01-08 |
| 9 | 9 | 10 | 2022-01-09 |
| 10 | 10 | 6 | 2022-01-10 |
+-----+-----+-----+-----+
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 |
+-----+-----+-----+
| Rahul | Sharma | rahul@gmail.com |
| Amit | Patel | amit@gmail.com |
| Priya | Singh | priya@gmail.com |
| Rajesh | Gupta | rajesh@gmail.com |
| Sneha | Verma | sneha@gmail.com |
| Vikram | Yadav | vikram@gmail.com |
| Neha | Shah | neha@gmail.com |
| Rohit | Mehta | rohit@gmail.com |
| Anita | Jain | anita@gmail.com |
| Sanjay | Kumar | sanjay@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,
->     CONCAT(Customers.FirstName, ' ', Customers.LastName) AS CustomerName
-> FROM
->     Orders
-> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
+-----+-----+-----+
| OrderID | OrderDate | CustomerName |
+-----+-----+-----+
|      1 | 2022-01-01 | Rahul Sharma |
|      2 | 2022-01-02 | Amit Patel |
|      3 | 2022-01-03 | Priya Singh |
|      4 | 2022-01-04 | Rajesh Gupta |
|      5 | 2022-01-05 | Sneha Verma |
|      6 | 2022-01-06 | Vikram Yadav |
|      7 | 2022-01-07 | Neha Shah |
|      8 | 2022-01-08 | Rohit Mehta |
|      9 | 2022-01-09 | Anita Jain |
|     10 | 2022-01-10 | Sanjay Kumar |
+-----+-----+-----+
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,
->         Phone,
->         Address
->     )
-> VALUES
->     (
->         11,
->         'John',
->         'Doe',
->         'john@gmail.com',
->         '9876543220',
->         '789 XYZ Street'
->     );
Query OK, 1 row affected (0.01 sec)
```

```

mysql> SELECT * FROM Customers;
+-----+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address |
+-----+-----+-----+-----+-----+-----+
| 1 | Rahul | Sharma | rahul@gmail.com | 9876543210 | 123 ABC Street |
| 2 | Amit | Patel | amit@gmail.com | 9876543211 | 456 XYZ Street |
| 3 | Priya | Singh | priya@gmail.com | 9876543212 | 789 PQR Street |
| 4 | Rajesh | Gupta | rajesh@gmail.com | 9876543213 | 321 LMN Street |
| 5 | Sneha | Verma | sneha@gmail.com | 9876543214 | 654 JKL Street |
| 6 | Vikram | Yadav | vikram@gmail.com | 9876543215 | 987 RST Street |
| 7 | Neha | Shah | neha@gmail.com | 9876543216 | 654 UVW Street |
| 8 | Rohit | Mehta | rohit@gmail.com | 9876543217 | 321 GHI Street |
| 9 | Anita | Jain | anita@gmail.com | 9876543218 | 789 DEF Street |
| 10 | Sanjay | Kumar | sanjay@gmail.com | 9876543219 | 123 MNO Street |
| 11 | John | Doe | john@gmail.com | 9876543220 | 789 XYZ Street |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

```

**4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.**

```

mysql> UPDATE Products
      -> SET Price = Price * 1.1;
Query OK, 10 rows affected (0.01 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> SELECT * FROM Products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 1 | Mobile Phone | Smartphone with 6GB RAM | 16500.00 |
| 2 | Laptop | 15.6 inch, 8GB RAM, 1TB HDD | 44000.00 |
| 3 | Headphones | Wireless Bluetooth headphones | 2200.00 |
| 4 | Smartwatch | Fitness tracker with heart rate monitor | 5500.00 |
| 5 | Camera | DSLR camera with 24MP | 33000.00 |
| 6 | Television | 55 inch LED TV | 55000.00 |
| 7 | Tablet | 10 inch Android tablet | 11000.00 |
| 8 | Gaming Console | PlayStation 5 | 44000.00 |
| 9 | Speakers | 2.1 channel multimedia speakers | 3300.00 |
| 10 | Printer | All-in-one printer | 5500.00 |
+-----+-----+-----+-----+
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" tables. Allow users to input the order ID as a parameter.

```
mysql> SET @InputOrderID = 3;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM
->   Orders
-> WHERE
->   OrderID = @InputOrderID
-> DELETE FROM
->   OrderDetails
-> WHERE
->   OrderID = @InputOrderID;
```

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

```
mysql> INSERT INTO
->   Orders (OrderID, CustomerID, OrderDate, TotalAmount)
-> VALUES
->   (11, 11, '2022-01-11', 35000);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT * FROM ORDERS;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
|      1 |          1 | 2022-01-01 |    20000.00 |
|      2 |          2 | 2022-01-02 |    45000.00 |
|      3 |          3 | 2022-01-03 |    10000.00 |
|      4 |          4 | 2022-01-04 |    35000.00 |
|      5 |          5 | 2022-01-05 |    25000.00 |
|      6 |          6 | 2022-01-06 |    15000.00 |
|      7 |          7 | 2022-01-07 |    30000.00 |
|      8 |          8 | 2022-01-08 |    50000.00 |
|      9 |          9 | 2022-01-09 |    40000.00 |
|     10 |         10 | 2022-01-10 |    60000.00 |
|     11 |         11 | 2022-01-11 |    35000.00 |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

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

```

mysql> SET @NewEmail = 'akash@gmail.com';
Query OK, 0 rows affected (0.00 sec)

mysql> SET @NewAddress = 'juhu tara road mumbai';
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE
    ->   Customers
    ->   SET
    ->     Email = @NewEmail,
    ->     Address = @NewAddress
    ->   WHERE
    ->     CustomerID = @InputCustomerID;
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 | Rahul    | Sharma   | new_email@gmail.com | 9876543210 | new_address   |
|       2 | Amit     | Patel    | akash@gmail.com    | 9876543211 | juhu tara road mumbai |
|       3 | Priya    | Singh    | priya@gmail.com   | 9876543212 | 789 PQR Street  |
|       4 | Rajesh   | Gupta    | rajesh@gmail.com  | 9876543213 | 321 LMN Street  |
|       5 | Sneha   | Verma    | sneha@gmail.com   | 9876543214 | 654 JKL Street  |
|       6 | Vikram   | Yadav    | vikram@gmail.com  | 9876543215 | 987 RST Street  |
|       7 | Neha     | Shah     | neha@gmail.com    | 9876543216 | 654 UVW Street  |
|       8 | Rohit   | Mehta    | rohit@gmail.com   | 9876543217 | 321 GHI Street  |
|       9 | Anita   | Jain     | anita@gmail.com   | 9876543218 | 789 DEF Street  |
|      10 | Sanjay   | Kumar    | sanjay@gmail.com  | 9876543219 | 123 MNO Street  |
|      11 | John     | Doe      | john@gmail.com   | 9876543220 | 789 XYZ Street  |
+-----+-----+-----+-----+-----+-----+

```

**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(Quantity * Price)
    ->       FROM
    ->         OrderDetails
    ->       JOIN Products ON OrderDetails.ProductID = Products.ProductID
    ->       WHERE
    ->         Orders.OrderID = OrderDetails.OrderID
    ->     )
    ->   WHERE
    ->     OrderID IN (
    ->       SELECT
    ->         OrderID
    ->       FROM
    ->         OrderDetails
    ->     )
    -> ;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 10  Changed: 0  Warnings: 0

```

```

mysql> select * from orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
|      1 |         1 | 2022-01-01 |    30000.00 |
|      2 |         2 | 2022-01-02 |    40000.00 |
|      3 |         3 | 2022-01-03 |     6000.00 |
|      4 |         4 | 2022-01-04 |    10000.00 |
|      5 |         5 | 2022-01-05 |    30000.00 |
|      6 |         6 | 2022-01-06 | 100000.00 |
|      7 |         7 | 2022-01-07 |    10000.00 |
|      8 |         8 | 2022-01-08 |    40000.00 |
|      9 |         9 | 2022-01-09 |     6000.00 |
|     10 |        10 | 2022-01-10 |    15000.00 |
|     11 |        11 | 2022-01-11 |    35000.00 |
+-----+-----+-----+-----+
11 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> DELETE FROM
->   OrderDetails
-> WHERE
->   OrderID IN (
->     SELECT
->       OrderID
->     FROM
->       Orders
->     WHERE
->       CustomerID = 1
->   )

```

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,
    >     'charger',
    >     'Apple charger type - c',
    >     10000
    >   );
Query OK, 1 row affected (0.00 sec)

```

```

mysql> select * from Products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 1 | Mobile Phone | Smartphone with 6GB RAM | 15000.00 |
| 2 | Laptop | 15.6 inch, 8GB RAM, 1TB HDD | 40000.00 |
| 3 | Headphones | Wireless Bluetooth headphones | 2000.00 |
| 4 | Smartwatch | Fitness tracker with heart rate monitor | 5000.00 |
| 5 | Camera | DSLR camera with 24MP | 30000.00 |
| 6 | Television | 55 inch LED TV | 50000.00 |
| 7 | Tablet | 10 inch Android tablet | 10000.00 |
| 8 | Gaming Console | PlayStation 5 | 40000.00 |
| 9 | Speakers | 2.1 channel multimedia speakers | 3000.00 |
| 10 | Printer | All-in-one printer | 5000.00 |
| 11 | charger | Apple charger type - c | 10000.00 |
+-----+-----+-----+-----+
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> select orderid,orderdate, if(orderdate>'2022-01-06','Pending','Shipped') from orders;
+-----+-----+-----+
| orderid | orderdate | if(orderdate>'2022-01-06','Pending','Shipped') |
+-----+-----+-----+
| 1 | 2022-01-01 | Shipped |
| 2 | 2022-01-02 | Shipped |
| 3 | 2022-01-03 | Shipped |
| 4 | 2022-01-04 | Shipped |
| 5 | 2022-01-05 | Shipped |
| 6 | 2022-01-06 | Shipped |
| 7 | 2022-01-07 | Pending |
| 8 | 2022-01-08 | Pending |
| 9 | 2022-01-09 | Pending |
| 10 | 2022-01-10 | Pending |
| 11 | 2022-01-11 | Pending |
+-----+-----+-----+
11 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 FROM OrdersDetails SET Quantity = (SELECT COUNT(*) FROM Orders  
->      WHERE  
->      Customers.CustomerID = Orders.CustomerID  
->      )  
-> WHERE CustomerID IN (SELECT CustomerID FROM Orders);
```

```
mysql> select * from OrderDetails;
```

OrderDetailID	OrderID	ProductID	Quantity
1	1	1	2
2	2	2	1
4	4	4	2
5	5	5	1
6	6	6	2
7	7	7	1
8	8	8	1
9	9	9	2
10	10	10	3

```
9 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
    ->     o.OrderID,
    ->     c.FirstName,
    ->     c.LastName,
    ->     c.Email,
    ->     c.Phone,
    ->     c.Address
    -> FROM
    ->     Orders o
    -> JOIN Customers c ON o.CustomerID = c.CustomerID;
+-----+-----+-----+-----+-----+-----+
| OrderID | FirstName | LastName | Email      | Phone     | Address   |
+-----+-----+-----+-----+-----+-----+
|      1 | Rahul     | Sharma   | new_email@gmail.com | 9876543210 | new_address |
|      2 | Amit      | Patel    | amit@gmail.com    | 9876543211 | 456 XYZ Street |
|      3 | Priya     | Singh    | priya@gmail.com   | 9876543212 | 789 PQR Street |
|      4 | Rajesh    | Gupta    | rajesh@gmail.com  | 9876543213 | 321 LMN Street |
|      5 | Sneha     | Verma   | sneha@gmail.com   | 9876543214 | 654 JKL Street |
|      6 | Vikram    | Yadav    | vikram@gmail.com  | 9876543215 | 987 RST Street |
|      7 | Neha      | Shah     | neha@gmail.com    | 9876543216 | 654 UVW Street |
|      8 | Rohit     | Mehta   | rohit@gmail.com   | 9876543217 | 321 GHI Street |
|      9 | Anita     | Jain    | anita@gmail.com   | 9876543218 | 789 DEF Street |
|     10 | Sanjay    | Kumar   | sanjay@gmail.com  | 9876543219 | 123 MNO Street |
|     11 | John      | Doe     | john@gmail.com    | 9876543220 | 789 XYZ Street |
+-----+-----+-----+-----+-----+-----+
11 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
    ->     p.ProductName,
    ->     SUM(od.Quantity * p.Price) AS TotalRevenue
    -> FROM
    ->     Products p
    ->     INNER JOIN OrderDetails od ON p.ProductID = od.ProductID
    -> GROUP BY
    ->     p.ProductName;
+-----+-----+
| ProductName | TotalRevenue |
+-----+-----+
| Mobile Phone | 30000.00 |
| Laptop       | 40000.00 |
| Headphones   | 6000.00  |
| Smartwatch   | 10000.00 |
| Camera       | 30000.00 |
| Television   | 100000.00 |
| Tablet       | 10000.00  |
| Gaming Console | 40000.00 |
| Speakers     | 6000.00  |
| Printer      | 15000.00 |
+-----+-----+
10 rows in set (0.00 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
    ->   FirstName,
    ->   LastName,
    ->   Email,
    ->   Phone,
    ->   Address
    -> FROM
    ->   Customers
    -> WHERE
    ->   CustomerID IN (
    ->     SELECT
    ->       CustomerID
    ->     FROM
    ->       Orders
    ->   );
+-----+-----+-----+-----+-----+
| FirstName | LastName | Email           | Phone      | Address        |
+-----+-----+-----+-----+-----+
| Rahul    | Sharma   | new_email@gmail.com | 9876543210 | new_address   |
| Amit     | Patel    | amit@gmail.com    | 9876543211 | 456 XYZ Street |
| Priya    | Singh    | priya@gmail.com   | 9876543212 | 789 PQR Street |
| Rajesh   | Gupta    | rajesh@gmail.com  | 9876543213 | 321 LMN Street |
| Sneha   | Verma    | sneha@gmail.com   | 9876543214 | 654 JKL Street |
| Vikram  | Yadav    | vikram@gmail.com  | 9876543215 | 987 RST Street |
| Neha    | Shah     | neha@gmail.com    | 9876543216 | 654 UVW Street |
| Rohit   | Mehta   | rohit@gmail.com   | 9876543217 | 321 GHI Street |
| Anita   | Jain     | anita@gmail.com   | 9876543218 | 789 DEF Street |
| Sanjay  | Kumar   | sanjay@gmail.com  | 9876543219 | 123 MNO Street |
| John    | Doe     | john@gmail.com    | 9876543220 | 789 XYZ Street |
+-----+-----+-----+-----+-----+
11 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
    ->   ProductName,
    ->   SUM(Quantity) AS TotalQuantityOrdered
    -> FROM
    ->   Products
    ->   JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID
    -> GROUP BY
    ->   ProductName
    -> ORDER BY
    ->   TotalQuantityOrdered DESC
    -> LIMIT
    ->   1;
+-----+-----+
| ProductName | TotalQuantityOrdered |
+-----+-----+
| Printer     |                      3 |
+-----+-----+
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
->     p.ProductName,
->     c.CategoryName
-> FROM
->     Products p
-> JOIN Categories c ON p.CategoryID = c.CategoryID
-> WHERE
->     c.CategoryName = 'Electronic Gadgets';
+-----+-----+
| ProductName | Description          |
+-----+-----+
| Mobile Phone | Smartphone with 6GB RAM
| Laptop        | 15.6 inch, 8GB RAM, 1TB HDD
| Headphones    | Wireless Bluetooth headphones
| Smartwatch    | Fitness tracker with heart rate monitor
| Camera         | DSLR camera with 24MP
| Television    | 55 inch LED TV
| Tablet         | 10 inch Android tablet
| Gaming Console | PlayStation 5
| Speakers       | 2.1 channel multimedia speakers
| Printer        | All-in-one printer
+-----+-----+
10 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
    ->   c.FirstName,
    ->   c.LastName,
    ->   SUM(o.TotalAmount) / SUM(od.Quantity) AS AverageOrderValue
    -> FROM
    ->   Customers c
    ->   JOIN Orders o ON c.CustomerID = o.CustomerID
    ->   JOIN OrderDetails od ON o.OrderID = od.OrderID
    -> GROUP BY
    ->   c.FirstName,
    ->   c.LastName;
+-----+-----+-----+
| FirstName | LastName | AverageOrderValue |
+-----+-----+-----+
| Rahul     | Sharma   | 15000.000000 |
| Amit      | Patel    | 40000.000000 |
| Priya     | Singh    | 2000.000000  |
| Rajesh    | Gupta    | 5000.000000  |
| Sneha     | Verma   | 30000.000000 |
| Vikram    | Yadav    | 50000.000000 |
| Neha      | Shah     | 10000.000000 |
| Rohit     | Mehta   | 40000.000000 |
| Anita     | Jain     | 3000.000000  |
| Sanjay    | Kumar    | 5000.000000  |
+-----+-----+-----+
10 rows in set (0.00 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
    ->   o.OrderID,
    ->   c.FirstName,
    ->   c.LastName,
    ->   c.Email,
    ->   c.Phone,
    ->   c.Address,
    ->   o.TotalAmount
    -> FROM
    ->   Orders o
    ->   JOIN Customers c ON o.CustomerID = c.CustomerID
    -> WHERE
    ->   o.TotalAmount = (
    ->       SELECT
    ->           MAX(TotalAmount)
    ->       FROM
    ->           Orders
    ->   );
+-----+-----+-----+-----+-----+-----+-----+
| OrderID | FirstName | LastName | Email        | Phone      | Address    | TotalAmount |
+-----+-----+-----+-----+-----+-----+-----+
|       6 | Vikram   | Yadav   | vikram@gmail.com | 9876543215 | 987 RST Street | 100000.00 |
+-----+-----+-----+-----+-----+-----+-----+
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
    ->     p.ProductName,
    ->     SUM(od.Quantity) AS TotalOrders
    -> FROM
    ->     Products p
    -> JOIN OrderDetails od ON p.ProductID = od.ProductID
    -> GROUP BY
    ->     p.ProductName;
+-----+-----+
| ProductName | TotalOrders |
+-----+-----+
| Mobile Phone | 2 |
| Laptop | 1 |
| Headphones | 3 |
| Smartwatch | 2 |
| Camera | 1 |
| Television | 2 |
| Tablet | 1 |
| Gaming Console | 1 |
| Speakers | 2 |
| Printer | 3 |
+-----+-----+
10 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> SET @ProductName = 'Laptop';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT
    ->     c.CustomerID,
    ->     c.FirstName,
    ->     c.LastName,
    ->     c.Email,
    ->     c.Phone,
    ->     c.Address
    -> FROM
    ->     Customers c
    -> JOIN Orders o ON c.CustomerID = o.CustomerID
    -> JOIN OrderDetails od ON o.OrderID = od.OrderID
    -> JOIN Products p ON od.ProductID = p.ProductID
    -> WHERE
    ->     p.ProductName = @ProductName;
+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address |
+-----+-----+-----+-----+-----+
| 2 | Amit | Patel | amit@gmail.com | 9876543211 | 456 XYZ Street |
+-----+-----+-----+-----+-----+
1 row 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> SET @StartDate = '2022-01-01';
Query OK, 0 rows affected (0.00 sec)

mysql> SET @EndDate = '2022-01-10';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT
    ->     SUM(TotalAmount) AS TotalRevenue
    ->     FROM
    ->         Orders
    ->     WHERE
    ->         OrderDate >= @StartDate
    ->         AND OrderDate <= @EndDate;
+-----+
| TotalRevenue |
+-----+
|      287000.00 |
+-----+
1 row in set (0.00 sec)
```

#### Task 4. Subquery and its type

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

```
mysql> SELECT
    ->     c.CustomerID,
    ->     c.FirstName,
    ->     c.LastName
    ->     FROM
    ->     Customers c
    ->     LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
    ->     WHERE
    ->         o.OrderID IS NULL;
Empty set (0.00 sec)
```

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

```
mysql> SELECT
->   COUNT(*) AS TotalProducts
->   FROM
->     Products;
+-----+
| TotalProducts |
+-----+
|          11 |
+-----+
1 row in set (0.01 sec)
```

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

```
mysql> SELECT
->   SUM(od.Quantity * p.Price) AS TotalRevenue
->   FROM
->     Orders o
->   JOIN OrderDetails od ON o.OrderID = od.OrderID
->   JOIN Products p ON od.ProductID = p.ProductID;
+-----+
| TotalRevenue |
+-----+
|    287000.00 |
+-----+
1 row in set (0.00 sec)
```

**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.**

```
mysql> SET @InputProductName = 'Headphones';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT
    ->     AVG(Quantity) AS AverageQuantity
    ->     FROM
    ->     OrderDetails
    ->     JOIN Products ON OrderDetails.ProductID = Products.ProductID
    ->     WHERE
    ->     Products.ProductName = @InputProductName;
+-----+
| AverageQuantity |
+-----+
|      3.0000 |
+-----+
1 row in set (0.00 sec)
```

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.

```
mysql> SET @InputCustomerID = 8;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT
    ->     SUM(od.Quantity * p.Price) AS TotalRevenue
    ->     FROM
    ->     Orders o
    ->     JOIN OrderDetails od ON o.OrderID = od.OrderID
    ->     JOIN Products p ON od.ProductID = p.ProductID
    ->     WHERE
    ->     o.CustomerID = @InputCustomerID;
+-----+
| TotalRevenue |
+-----+
|   40000.00 |
+-----+
1 row in set (0.00 sec)
```

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.

```

mysql> SELECT
->   c.FirstName,
->   c.LastName,
->   COUNT(od.Quantity) AS Quantity
->  FROM
->    Customers c
->  JOIN Orders o ON c.CustomerID = o.CustomerID
->  JOIN OrderDetails od ON o.OrderID = od.OrderID
-> GROUP BY
->   c.FirstName,
->   c.LastName
-> ORDER BY
->   Quantity DESC
-> LIMIT
->   1;
+-----+-----+-----+
| FirstName | LastName | Quantity |
+-----+-----+-----+
| Rahul     | Sharma   |      1 |
+-----+-----+-----+
1 row in set (0.00 sec)

```

**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.**

```

mysql> SELECT
->   c.FirstName,
->   c.LastName,
->   COUNT(od.Quantity) AS Quantity
->  FROM
->    Customers c
->  JOIN Orders o ON c.CustomerID = o.CustomerID
->  JOIN OrderDetails od ON o.OrderID = od.OrderID
-> GROUP BY
->   c.FirstName,
->   c.LastName
-> ORDER BY
->   Quantity DESC
-> LIMIT
->   1;
+-----+-----+-----+
| FirstName | LastName | Quantity |
+-----+-----+-----+
| Rahul     | Sharma   |      1 |
+-----+-----+-----+
1 row in set (0.00 sec)

```

**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.**

```
mysql> SELECT
->   c.FirstName,
->   c.LastName,
->   SUM(od.Quantity * p.Price) AS TotalSpending
-> FROM
->   Customers c
->   INNER JOIN Orders o ON c.CustomerID = o.CustomerID
->   INNER JOIN OrderDetails od ON o.OrderID = od.OrderID
->   INNER JOIN Products p ON od.ProductID = p.ProductID
-> GROUP BY
->   c.CustomerID
-> ORDER BY
->   TotalSpending DESC
-> LIMIT
->   1;
+-----+-----+-----+
| FirstName | LastName | TotalSpending |
+-----+-----+-----+
| Vikram    | Yadav    |      100000.00 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```

mysql> SELECT
    ->     c.FirstName,
    ->     c.LastName,
    ->     SUM(o.TotalAmount) / SUM(od.Quantity) AS AverageOrderValue
    -> FROM
    ->     Customers c
    ->     JOIN Orders o ON c.CustomerID = o.CustomerID
    ->     JOIN OrderDetails od ON o.OrderID = od.OrderID
    -> GROUP BY
    ->     c.FirstName,
    ->     c.LastName;
+-----+-----+-----+
| FirstName | LastName | AverageOrderValue |
+-----+-----+-----+
| Rahul     | Sharma   | 15000.000000 |
| Amit      | Patel    | 40000.000000 |
| Priya     | Singh    | 2000.000000 |
| Rajesh    | Gupta    | 5000.000000 |
| Sneha     | Verma   | 30000.000000 |
| Vikram    | Yadav    | 50000.000000 |
| Neha      | Shah     | 10000.000000 |
| Rohit    | Mehta   | 40000.000000 |
| Anita    | Jain     | 3000.000000 |
| Sanjay    | Kumar    | 5000.000000 |
+-----+-----+-----+
10 rows in set (0.00 sec)

```

- 10.** Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.

```
mysql> SELECT
->   c.FirstName,
->   c.LastName,
->   COUNT(od.Quantity) AS OrderCount
-> FROM
->   Customers c
->   JOIN Orders o ON c.CustomerID = o.CustomerID
->   JOIN OrderDetails od ON o.OrderID = od.OrderID
-> GROUP BY
->   c.FirstName,
->   c.LastName;
+-----+-----+-----+
| FirstName | LastName | OrderCount |
+-----+-----+-----+
| Rahul     | Sharma   |          1 |
| Amit      | Patel    |          1 |
| Priya     | Singh    |          1 |
| Rajesh    | Gupta    |          1 |
| Sneha     | Verma   |          1 |
| Vikram    | Yadav    |          1 |
| Neha      | Shah     |          1 |
| Rohit     | Mehta   |          1 |
| Anita     | Jain     |          1 |
| Sanjay    | Kumar   |          1 |
+-----+-----+-----+
10 rows in set (0.00 sec)
```