Task:1. Database Design:

1. Create the database named "TechShop"

```
MySQL 8.0 Command Line Client
| mysql              |
| performance_schema |
| sakila             |
| student            |
| sys                |
| world              |
+--------------------+
8 rows in set (0.00 sec)

mysql> create database TechShop
    -> ;
Query OK, 1 row affected (0.15 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydb               |
| mysql              |
| performance_schema |
| sakila             |
| student            |
| sys                |
| techshop           |
| world              |
+--------------------+
9 rows in set (0.00 sec)

mysql> use techshop
Database changed
mysql>
```

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

```
MySQL 8.0 Command Line Client
mysql> CREATE TABLE Customers (
    ->     CustomerID INT AUTO_INCREMENT PRIMARY KEY,
    ->     FirstName VARCHAR(255),
    ->     LastName VARCHAR(255),
    ->     Email VARCHAR(255),
    ->     Phone VARCHAR(20),
    ->     Address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.93 sec)

mysql>
mysql> -- Create new Products table
mysql> CREATE TABLE Products (
    ->     ProductID INT AUTO_INCREMENT PRIMARY KEY,
    ->     ProductName VARCHAR(255),
    ->     Description TEXT,
    ->     Price DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (1.55 sec)

mysql>
mysql> -- Create new Orders table
mysql> CREATE TABLE Orders (
    ->     OrderID INT AUTO_INCREMENT PRIMARY KEY,
    ->     CustomerID INT,
    ->     OrderDate DATETIME,
    ->     TotalAmount DECIMAL(10, 2),
    ->     Status VARCHAR(50),
    ->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
    -> );
Query OK, 0 rows affected (0.71 sec)

mysql>
```

Making Email field unique in Customers table -

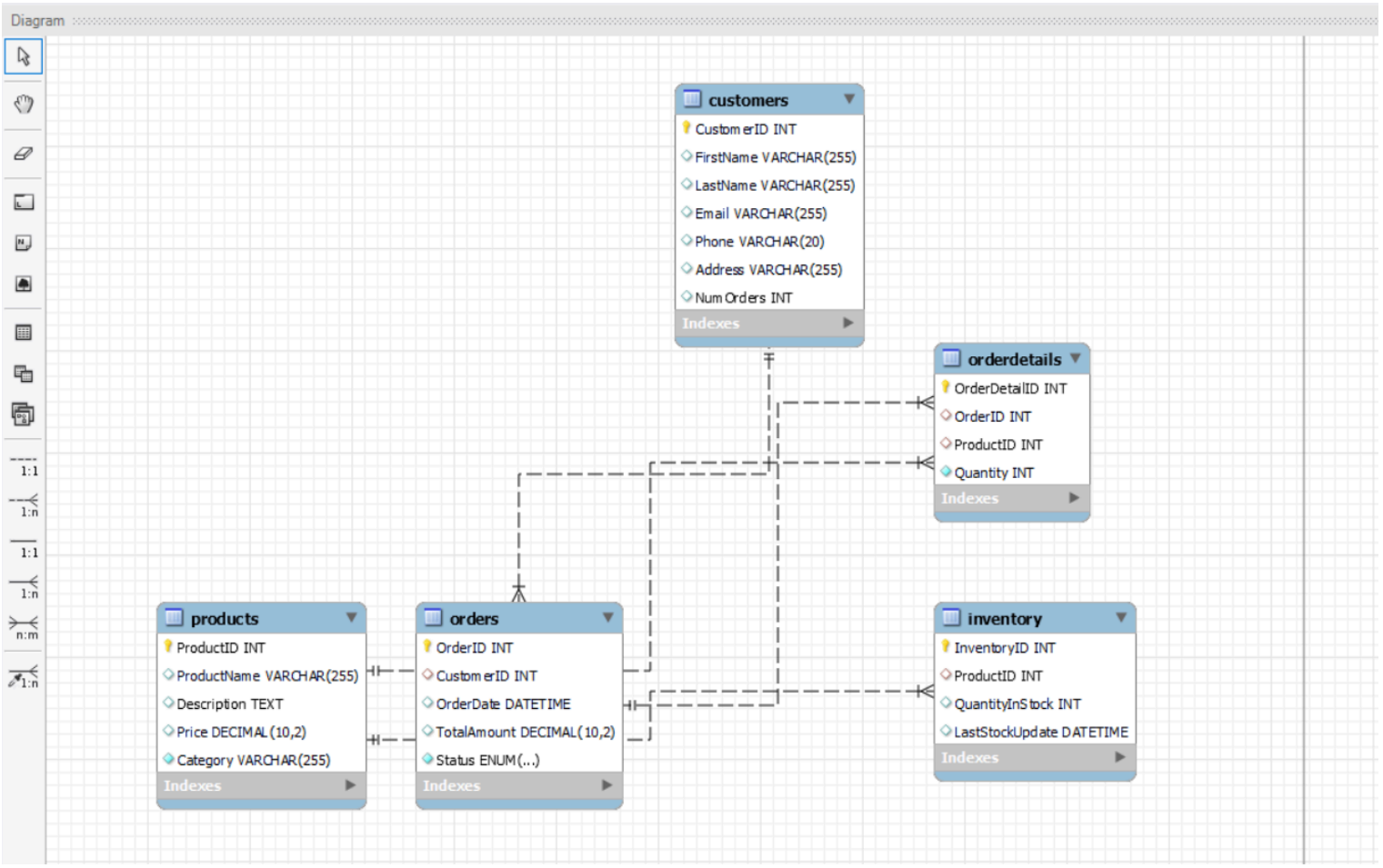
```
mysql> ALTER TABLE Customers
    -> ADD CONSTRAINT unique_email UNIQUE (Email);
Query OK, 0 rows affected (1.82 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MySQL 8.0 Command Line Client
mysql> -- Create new OrderDetails table
mysql> CREATE TABLE OrderDetails (
    ->     OrderDetailID INT AUTO_INCREMENT PRIMARY KEY,
    ->     OrderID INT,
    ->     ProductID INT,
    ->     Quantity INT NOT NULL,
    ->     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
Query OK, 0 rows affected (1.08 sec)

mysql>
mysql> -- Create new Inventory table
mysql> CREATE TABLE Inventory (
    ->     InventoryID INT AUTO_INCREMENT PRIMARY KEY,
    ->     ProductID INT,
    ->     QuantityInStock INT,
    ->     LastStockUpdate DATETIME,
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
Query OK, 0 rows affected (1.30 sec)

mysql>
```

Making Status field Enum in Orders table -

```
mysql> ALTER TABLE Orders
    -> MODIFY COLUMN Status ENUM('Pending', 'Processing', 'Shipped', 'Delivered') DEFAULT 'Pending' NOT NULL;
Query OK, 10 rows affected (2.74 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

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



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

MySQL 8.0 Command Line Client

```
mysql> desc orders;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| OrderID     | int          | NO   | PRI | NULL    | auto_increment |
| CustomerID  | int          | YES  | MUL | NULL    |                |
| OrderDate   | datetime     | YES  |     | NULL    |                |
| TotalAmount | decimal(10,2) | YES  |     | NULL    |                |
| Status      | varchar(50)  | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
5 rows in set (0.16 sec)

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

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

mysql>
```

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

MySQL 8.0 Command Line Client

```
mysql> INSERT INTO Customers (FirstName, LastName, Email, Phone, Address)
    -> VALUES
    ->     ('Rahul', 'Kumar', 'rahul.kumar@example.com', '9876543210', '123, MG Road, Bangalore'),
    ->     ('Priya', 'Sharma', 'priya.sharma@example.com', '8765432109', '456, Jubilee Hills, Hyderabad'),
    ->     ('Vikram', 'Singh', 'vikram.singh@example.com', '7654321098', '789, Malviya Nagar, Delhi'),
    ->     ('Deepika', 'Patel', 'deepika.patel@example.com', '6543210987', '101, Koregaon Park, Pune'),
    ->     ('Amit', 'Verma', 'amit.verma@example.com', '5432109876', '234, Baner Road, Pune'),
    ->     ('Ananya', 'Nair', 'ananya.nair@example.com', '4321098765', '567, Marathahalli, Bangalore'),
    ->     ('Raj', 'Malhotra', 'raj.malhotra@example.com', '3210987654', '890, HSR Layout, Bangalore'),
    ->     ('Neha', 'Srivastava', 'neha.srivastava@example.com', '2109876543', '123, Andheri West, Mumbai'),
    ->     ('Sandeep', 'Gupta', 'sandeep.gupta@example.com', '1098765432', '456, Aundh, Pune'),
    ->     ('Shreya', 'Rajput', 'shreya.rajput@example.com', '9876543210', '789, Banashankari, Bangalore');
Query OK, 10 rows affected (0.62 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from customers
    -> ;
+------------+-----------+-----------+---------------------------+------------+-------------------------------+
| CustomerID | FirstName | LastName  | Email                     | Phone      | Address                       |
+------------+-----------+-----------+---------------------------+------------+-------------------------------+
|          1 | Rahul     | Kumar     | rahul.kumar@example.com   | 9876543210 | 123, MG Road, Bangalore       |
|          2 | Priya     | Sharma    | priya.sharma@example.com  | 8765432109 | 456, Jubilee Hills, Hyderabad |
|          3 | Vikram    | Singh     | vikram.singh@example.com  | 7654321098 | 789, Malviya Nagar, Delhi     |
|          4 | Deepika   | Patel     | deepika.patel@example.com | 6543210987 | 101, Koregaon Park, Pune      |
|          5 | Amit      | Verma     | amit.verma@example.com    | 5432109876 | 234, Baner Road, Pune         |
|          6 | Ananya    | Nair      | ananya.nair@example.com   | 4321098765 | 567, Marathahalli, Bangalore  |
|          7 | Raj       | Malhotra  | raj.malhotra@example.com  | 3210987654 | 890, HSR Layout, Bangalore    |
|          8 | Neha      | Srivastava | neha.srivastava@example.com | 2109876543 | 123, Andheri West, Mumbai     |
|          9 | Sandeep   | Gupta     | sandeep.gupta@example.com | 1098765432 | 456, Aundh, Pune              |
|         10 | Shreya    | Rajput    | shreya.rajput@example.com | 9876543210 | 789, Banashankari, Bangalore  |
+------------+-----------+-----------+---------------------------+------------+-------------------------------+
10 rows in set (0.00 sec)

mysql>
```

    b. Products

MySQL 8.0 Command Line Client

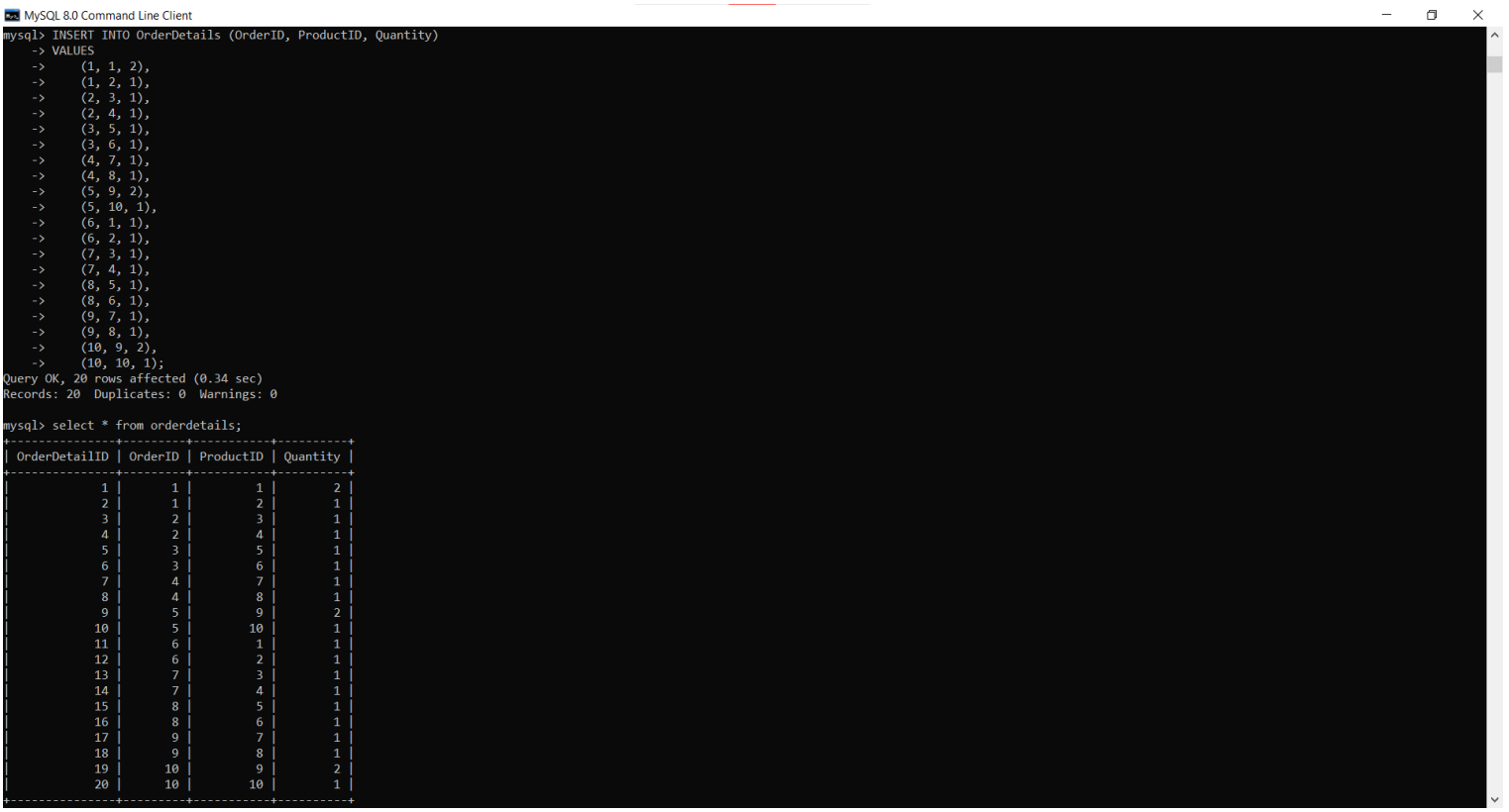
```
10 rows in set (0.00 sec)

mysql> INSERT INTO Products (ProductName, Description, Price)
    -> VALUES
    ->     ('LED TV', 'Full HD Smart LED TV', 35000.00),
    ->     ('Air Conditioner', '1.5 Ton Split AC', 30000.00),
    ->     ('Washing Machine', 'Front Load Fully Automatic', 25000.00),
    ->     ('Refrigerator', 'Double Door Refrigerator', 20000.00),
    ->     ('Mobile Phone', 'Latest Android Smartphone', 15000.00),
    ->     ('Laptop', 'Thin and Light Laptop', 45000.00),
    ->     ('Microwave Oven', 'Convection Microwave Oven', 12000.00),
    ->     ('Camera', 'DSLR Camera with Kit Lens', 35000.00),
    ->     ('Water Purifier', 'RO + UV Water Purifier', 12000.00),
    ->     ('Vacuum Cleaner', 'Robotic Vacuum Cleaner', 18000.00);
Query OK, 10 rows affected (0.35 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from products;
+-----------+-----------------+---------------------------+----------+
| ProductID | ProductName     | Description               | Price    |
+-----------+-----------------+---------------------------+----------+
|         1 | LED TV          | Full HD Smart LED TV      | 35000.00 |
|         2 | Air Conditioner | 1.5 Ton Split AC          | 30000.00 |
|         3 | Washing Machine | Front Load Fully Automatic | 25000.00 |
|         4 | Refrigerator    | Double Door Refrigerator  | 20000.00 |
|         5 | Mobile Phone    | Latest Android Smartphone | 15000.00 |
|         6 | Laptop          | Thin and Light Laptop     | 45000.00 |
|         7 | Microwave Oven  | Convection Microwave Oven | 12000.00 |
|         8 | Camera          | DSLR Camera with Kit Lens | 35000.00 |
|         9 | Water Purifier  | RO + UV Water Purifier    | 12000.00 |
|        10 | Vacuum Cleaner  | Robotic Vacuum Cleaner    | 18000.00 |
+-----------+-----------------+---------------------------+----------+
10 rows in set (0.00 sec)

mysql>
```

c. Orders

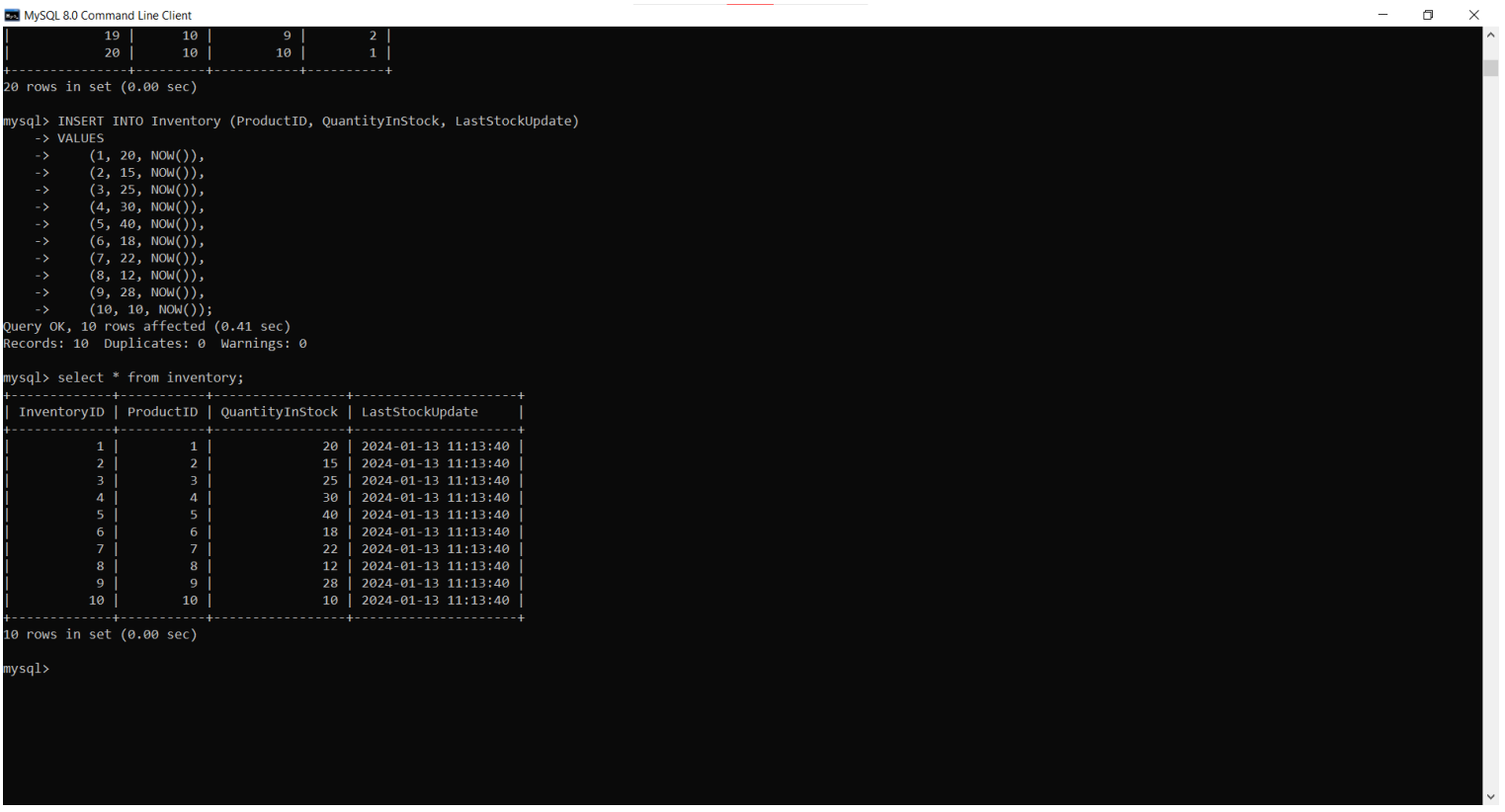
MySQL 8.0 Command Line Client

```
mysql> INSERT INTO Orders (CustomerID, OrderDate, TotalAmount, Status)
    -> VALUES
    ->     (1, NOW(), 35000.00, 'Pending'),
    ->     (2, NOW(), 30000.00, 'Shipped'),
    ->     (3, NOW(), 25000.00, 'Delivered'),
    ->     (4, NOW(), 20000.00, 'Pending'),
    ->     (5, NOW(), 15000.00, 'Shipped'),
    ->     (6, NOW(), 45000.00, 'Delivered'),
    ->     (7, NOW(), 12000.00, 'Pending'),
    ->     (8, NOW(), 35000.00, 'Shipped'),
    ->     (9, NOW(), 12000.00, 'Delivered'),
    ->     (10, NOW(), 18000.00, 'Pending');
Query OK, 10 rows affected (0.38 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from orders;
+---------+------------+---------------------+-------------+-----------+
| OrderID | CustomerID | OrderDate           | TotalAmount | Status    |
+---------+------------+---------------------+-------------+-----------+
|       1 |          1 | 2024-01-13 11:12:04 |    35000.00 | Pending   |
|       2 |          2 | 2024-01-13 11:12:04 |    30000.00 | Shipped   |
|       3 |          3 | 2024-01-13 11:12:04 |    25000.00 | Delivered |
|       4 |          4 | 2024-01-13 11:12:04 |    20000.00 | Pending   |
|       5 |          5 | 2024-01-13 11:12:04 |    15000.00 | Shipped   |
|       6 |          6 | 2024-01-13 11:12:04 |    45000.00 | Delivered |
|       7 |          7 | 2024-01-13 11:12:04 |    12000.00 | Pending   |
|       8 |          8 | 2024-01-13 11:12:04 |    35000.00 | Shipped   |
|       9 |          9 | 2024-01-13 11:12:04 |    12000.00 | Delivered |
|      10 |         10 | 2024-01-13 11:12:04 |    18000.00 | Pending   |
+---------+------------+---------------------+-------------+-----------+
10 rows in set (0.00 sec)

mysql>
```

d. OrderDetails



e. Inventory



Tasks 2: Select, Where, Between, AND, LIKE:

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

MySQL 8.0 Command Line Client

```
mysql> SELECT FirstName, LastName, Email FROM Customers;
+-----------+----------+-------------------------+
| FirstName | LastName | Email                   |
+-----------+----------+-------------------------+
| Rahul     | Verma    | rahul.verma@example.com |
| Priya     | Sharma   | priya.sharma@example.com |
| Amit      | Patil    | amit.patil@example.com  |
| Sneha     | Rao      | sneha.rao@example.com   |
| Anand     | Joshi    | anand.joshi@example.com |
| Pooja     | Mehta    | pooja.mehta@example.com |
| Vikas     | Kumar    | vikas.kumar@example.com |
| Neha      | Singh    | neha.singh@example.com  |
| Raj       | Gupta    | raj.gupta@example.com   |
| Deepa     | Nair     | deepa.nair@example.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 8.0 Command Line Client

```
+-----------+----------+-------------------------+
10 rows in set (0.00 sec)

mysql> SELECT o.OrderID, o.OrderDate, c.FirstName, c.LastName
    -> FROM Orders o
    -> JOIN Customers c ON o.CustomerID = c.CustomerID;
+---------+---------------------+-----------+----------+
| OrderID | OrderDate           | FirstName | LastName |
+---------+---------------------+-----------+----------+
|       1 | 2024-01-12 16:22:22 | Rahul     | Verma    |
|       3 | 2024-01-10 16:22:22 | Amit      | Patil    |
|       4 | 2024-01-09 16:22:22 | Sneha     | Rao      |
|       5 | 2024-01-08 16:22:22 | Anand     | Joshi    |
|       6 | 2024-01-07 16:22:22 | Pooja     | Mehta    |
|       7 | 2024-01-06 16:22:22 | Vikas     | Kumar    |
|       8 | 2024-01-05 16:22:22 | Neha      | Singh    |
|       9 | 2024-01-04 16:22:22 | Raj       | Gupta    |
|      10 | 2024-01-03 16:22:22 | Deepa     | Nair     |
+---------+---------------------+-----------+----------+
9 rows in set (0.00 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> INSERT INTO Customers (FirstName, LastName, Email, Phone, Address)
    -> VALUES ('Vatsal', 'Patel', 'vatsal.patel@email.com', '9876543210', '123, Main Street, Surat');
Query OK, 1 row affected (0.14 sec)

mysql> SELECT * FROM Customers
    -> ;
+------------+-----------+------------+---------------------------+------------+-----------------------------+
| CustomerID | FirstName | LastName   | Email                     | Phone      | Address                     |
+------------+-----------+------------+---------------------------+------------+-----------------------------+
|          1 | Rahul     | Kumar      | rahul.kumar@example.com   | 9876543210 | 123, MG Road, Bangalore     |
|          2 | Priya     | Sharma     | priya.sharma@example.com  | 8765432109 | 456, Jubilee Hills, Hyderabad |
|          3 | Vikram    | Singh      | vikram.singh@example.com  | 7654321098 | 789, Malviya Nagar, Delhi   |
|          4 | Deepika   | Patel      | deepika.patel@example.com | 6543210987 | 101, Koregaon Park, Pune    |
|          5 | Amit      | Verma      | amit.verma@example.com    | 5432109876 | 234, Baner Road, Pune       |
|          6 | Ananya    | Nair       | ananya.nair@example.com   | 4321098765 | 567, Marathahalli, Bangalore |
|          7 | Raj       | Malhotra   | raj.malhotra@example.com  | 3210987654 | 890, HSR Layout, Bangalore  |
|          8 | Neha      | Srivastava | neha.srivastava@example.com | 2109876543 | 123, Andheri West, Mumbai |
|          9 | Sandeep   | Gupta      | sandeep.gupta@example.com | 1098765432 | 456, Aundh, Pune            |
|         10 | Shreya    | Rajput     | shreya.rajput@example.com | 9876543210 | 789, Banashankari, Bangalore |
|         12 | Vatsal    | Patel      | vatsal.patel@email.com    | 9876543210 | 123, Main Street, Surat     |
+------------+-----------+------------+---------------------------+------------+-----------------------------+
11 rows in set (0.00 sec)

mysql>
```

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

MySQL 8.0 Command Line Client

```
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`techshop`.`inventory`, CONSTRAINT `inventory_ibfk_1` FOREIGN KEY (`ProductID`) REFERENCES `product
mysql> SELECT * FROM Products;
+-----------+-----------------+---------------------------+----------+
| ProductID | ProductName     | Description               | Price    |
+-----------+-----------------+---------------------------+----------+
|         1 | LED TV          | Full HD Smart LED TV      | 35000.00 |
|         2 | Air Conditioner | 1.5 Ton Split AC          | 30000.00 |
|         3 | Washing Machine | Front Load Fully Automatic | 25000.00 |
|         4 | Refrigerator    | Double Door Refrigerator  | 20000.00 |
|         5 | Mobile Phone    | Latest Android Smartphone | 15000.00 |
|         6 | Laptop          | Thin and Light Laptop     | 45000.00 |
|         7 | Microwave Oven  | Convection Microwave Oven | 12000.00 |
|         8 | Camera          | DSLR Camera with Kit Lens | 35000.00 |
|         9 | Water Purifier  | RO + UV Water Purifier    | 12000.00 |
|        10 | Vacuum Cleaner  | Robotic Vacuum Cleaner    | 18000.00 |
+-----------+-----------------+---------------------------+----------+
10 rows in set (0.00 sec)

mysql> UPDATE Products
    -> Price = Price* 1.1;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '= Price* 1.1' at line 2
mysql> UPDATE Products
    -> SET Price = Price* 1.1;
Query OK, 10 rows affected (0.16 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> SELECT * FROM Products;
+-----------+-----------------+---------------------------+----------+
| ProductID | ProductName     | Description               | Price    |
+-----------+-----------------+---------------------------+----------+
|         1 | LED TV          | Full HD Smart LED TV      | 38500.00 |
|         2 | Air Conditioner | 1.5 Ton Split AC          | 33000.00 |
|         3 | Washing Machine | Front Load Fully Automatic | 27500.00 |
|         4 | Refrigerator    | Double Door Refrigerator  | 22000.00 |
|         5 | Mobile Phone    | Latest Android Smartphone | 16500.00 |
|         6 | Laptop          | Thin and Light Laptop     | 49500.00 |
|         7 | Microwave Oven  | Convection Microwave Oven | 13200.00 |
|         8 | Camera          | DSLR Camera with Kit Lens | 38500.00 |
|         9 | Water Purifier  | RO + UV Water Purifier    | 13200.00 |
|        10 | Vacuum Cleaner  | Robotic Vacuum Cleaner    | 19800.00 |
+-----------+-----------------+---------------------------+----------+
10 rows in set (0.00 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> ^C
mysql> DELIMITER //
mysql> CREATE PROCEDURE DeleteOrderAndDetails(IN p_OrderID INT)
    -> BEGIN
    ->          DELETE FROM OrderDetails
    ->          WHERE OrderID = p_OrderID;
    ->          DELETE FROM Orders
    ->          WHERE OrderID = p_OrderID;
    -> END //
Query OK, 0 rows affected (0.26 sec)

mysql> DELIMITER ;
mysql> CALL DeleteOrderAndDetails(3);
Query OK, 1 row affected (0.34 sec)
```

MySQL 8.0 Command Line Client

```
mysql> select * from orders;
+---------+------------+---------------------+-------------+-----------+
| OrderID | CustomerID | OrderDate           | TotalAmount | Status    |
+---------+------------+---------------------+-------------+-----------+
|       1 |          1 | 2024-01-13 11:12:04 |    35000.00 | Pending   |
|       2 |          2 | 2024-01-13 11:12:04 |    30000.00 | Shipped   |
|       4 |          4 | 2024-01-13 11:12:04 |    20000.00 | Pending   |
|       5 |          5 | 2024-01-13 11:12:04 |    15000.00 | Shipped   |
|       6 |          6 | 2024-01-13 11:12:04 |    45000.00 | Delivered |
|       7 |          7 | 2024-01-13 11:12:04 |    12000.00 | Pending   |
|       8 |          8 | 2024-01-13 11:12:04 |    35000.00 | Shipped   |
|       9 |          9 | 2024-01-13 11:12:04 |    12000.00 | Delivered |
|      10 |         10 | 2024-01-13 11:12:04 |    18000.00 | Pending   |
+---------+------------+---------------------+-------------+-----------+
9 rows in set (0.00 sec)

mysql> select * from orderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|             1 |       1 |         1 |        2 |
|             2 |       1 |         2 |        1 |
|             3 |       2 |         3 |        1 |
|             4 |       2 |         4 |        1 |
|             7 |       4 |         7 |        1 |
|             8 |       4 |         8 |        1 |
|             9 |       5 |         9 |        2 |
|            10 |       5 |        10 |        1 |
|            11 |       6 |         1 |        1 |
|            12 |       6 |         2 |        1 |
|            13 |       7 |         3 |        1 |
|            14 |       7 |         4 |        1 |
|            15 |       8 |         5 |        1 |
|            16 |       8 |         6 |        1 |
|            17 |       9 |         7 |        1 |
|            18 |       9 |         8 |        1 |
|            19 |      10 |         9 |        2 |
|            20 |      10 |        10 |        1 |
+---------------+---------+-----------+----------+
18 rows in set (0.00 sec)

mysql>
```

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 (CustomerID, OrderDate, TotalAmount, Status)
    -> VALUES (1, NOW(), 150.00, 'Pending');
Query OK, 1 row affected (0.17 sec)

mysql> select * from orders;
+---------+------------+---------------------+-------------+-----------+
| OrderID | CustomerID | OrderDate           | TotalAmount | Status    |
+---------+------------+---------------------+-------------+-----------+
|       1 |          1 | 2024-01-13 11:12:04 |    35000.00 | Pending   |
|       2 |          2 | 2024-01-13 11:12:04 |    30000.00 | Shipped   |
|       4 |          4 | 2024-01-13 11:12:04 |    20000.00 | Pending   |
|       5 |          5 | 2024-01-13 11:12:04 |    15000.00 | Shipped   |
|       6 |          6 | 2024-01-13 11:12:04 |    45000.00 | Delivered |
|       7 |          7 | 2024-01-13 11:12:04 |    12000.00 | Pending   |
|       8 |          8 | 2024-01-13 11:12:04 |    35000.00 | Shipped   |
|       9 |          9 | 2024-01-13 11:12:04 |    12000.00 | Delivered |
|      10 |         10 | 2024-01-13 11:12:04 |    18000.00 | Pending   |
|      11 |          1 | 2024-01-13 12:33:22 |      150.00 | Pending   |
+---------+------------+---------------------+-------------+-----------+
10 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.

```
1   DELIMITER //
2
3 • CREATE PROCEDURE UpdateContactInfo(IN p_CustomerID INT, IN p_NewEmail VARCHAR(255), IN p_NewPhone VARCHAR(20))
4   BEGIN
5       UPDATE Customers
6       SET
7           Email = IFNULL(p_NewEmail, Email),
8           Phone = IFNULL(p_NewPhone, Phone)
9       WHERE CustomerID = p_CustomerID;
10  END //
11
12  DELIMITER ;
13
```

MySQL 8.0 Command Line Client

```
mysql> CALL UpdateContactInfo(4, 'newemail@gmail.com', '+91 1111111111');
Query OK, 1 row affected (0.10 sec)

mysql> select * from customers;
+------------+-----------+------------+---------------------------+----------------+-------------------------------+
| CustomerID | FirstName | LastName   | Email                     | Phone          | Address                       |
+------------+-----------+------------+---------------------------+----------------+-------------------------------+
|          1 | Rahul     | Kumar      | rahul.kumar@example.com   | 9876543210     | 123, MG Road, Bangalore       |
|          2 | Priya     | Sharma     | priya.sharma@example.com  | 8765432109     | 456, Jubilee Hills, Hyderabad |
|          3 | Vikram    | Singh      | vikram.singh@example.com  | 7654321098     | 789, Malviya Nagar, Delhi     |
|          4 | Deepika   | Patel      | newemail@gmail.com        | +91 1111111111 | 101, Koregaon Park, Pune      |
|          5 | Amit      | Verma      | amit.verma@example.com    | 5432109876     | 234, Baner Road, Pune         |
|          6 | Ananya    | Nair       | ananya.nair@example.com   | 4321098765     | 567, Marathahalli, Bangalore  |
|          7 | Raj       | Malhotra   | raj.malhotra@example.com  | 3210987654     | 890, HSR Layout, Bangalore    |
|          8 | Neha      | Srivastava | neha.srivastava@example.com | 2109876543   | 123, Andheri West, Mumbai     |
|          9 | Sandeep   | Gupta      | sandeep.gupta@example.com | 1098765432     | 456, Aundh, Pune              |
|         10 | Shreya    | Rajput     | shreya.rajput@example.com | 9876543210     | 789, Banashankari, Bangalore  |
|         12 | Vatsal    | Patel      | vatsal.patel@email.com    | 9876543210     | 123, Main Street, Surat       |
+------------+-----------+------------+---------------------------+----------------+-------------------------------+
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 8.0 Command Line Client

```
10 rows in set (0.00 sec)

mysql> -- Update the total cost of each order in the Orders table
mysql> UPDATE Orders o
    -> SET TotalAmount = (
    ->     SELECT SUM(p.Price * od.Quantity)
    ->     FROM OrderDetails od
    ->     JOIN Products p ON od.ProductID = p.ProductID
    ->     WHERE od.OrderID = o.OrderID
    -> );
Query OK, 10 rows affected (0.17 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql>
mysql> -- If there are orders without details, you can set TotalAmount to 0 for those orders
mysql> UPDATE Orders
    -> SET TotalAmount = 0
    -> WHERE OrderID NOT IN (SELECT DISTINCT OrderID FROM OrderDetails);
Query OK, 1 row affected (0.27 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from orders;
+---------+------------+---------------------+-------------+-----------+
| OrderID | CustomerID | OrderDate           | TotalAmount | Status    |
+---------+------------+---------------------+-------------+-----------+
|       1 |          1 | 2024-01-13 11:12:04 |   110000.00 | Pending   |
|       2 |          2 | 2024-01-13 11:12:04 |    49500.00 | Shipped   |
|       4 |          4 | 2024-01-13 11:12:04 |    51700.00 | Pending   |
|       5 |          5 | 2024-01-13 11:12:04 |    46200.00 | Shipped   |
|       6 |          6 | 2024-01-13 11:12:04 |    71500.00 | Delivered |
|       7 |          7 | 2024-01-13 11:12:04 |    49500.00 | Pending   |
|       8 |          8 | 2024-01-13 11:12:04 |    66000.00 | Shipped   |
|       9 |          9 | 2024-01-13 11:12:04 |    51700.00 | Delivered |
|      10 |         10 | 2024-01-13 11:12:04 |    46200.00 | Pending   |
|      11 |          1 | 2024-01-13 12:33:22 |        0.00 | Pending   |
+---------+------------+---------------------+-------------+-----------+
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.

```
DELIMITER //

CREATE PROCEDURE DeleteOrdersForCustomer(IN p_CustomerID INT)
BEGIN
    -- Delete from OrderDetails table
    DELETE FROM OrderDetails
    WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = p_CustomerID);

    -- Delete from Orders table
    DELETE FROM Orders
    WHERE CustomerID = p_CustomerID;
END //

DELIMITER ;
```

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.

Added category field in Products -

```
mysql> ALTER TABLE Products
    -> ADD COLUMN Category VARCHAR(255) NOT NULL;
Query OK, 0 rows affected (38.43 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> UPDATE Products
    -> SET Category = 'Electronics'
    -> WHERE ProductID IN (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11);
Query OK, 11 rows affected (1.44 sec)
Rows matched: 11  Changed: 11  Warnings: 0
```

MySQL 8.0 Command Line Client

```
Rows matched: 11  Changed: 11  Warnings: 0

mysql> INSERT INTO Products (ProductName, Description, Price, Category)
    -> VALUES ('Smartwatch', 'Fitness Tracker with Heart Rate Monitor', 7500.00, 'Wearables');
Query OK, 1 row affected (2.00 sec)

mysql> select * from products;
+-----------+----------------+---------------------------------------+---------+-------------+
| ProductID | ProductName    | Description                           | Price   | Category    |
+-----------+----------------+---------------------------------------+---------+-------------+
|         1 | LED TV         | Full HD Smart LED TV                  | 38500.00 | Electronics |
|         2 | Air Conditioner | 1.5 Ton Split AC                     | 33000.00 | Electronics |
|         3 | Washing Machine | Front Load Fully Automatic           | 27500.00 | Electronics |
|         4 | Refrigerator   | Double Door Refrigerator              | 22000.00 | Electronics |
|         5 | Mobile Phone   | Latest Android Smartphone             | 16500.00 | Electronics |
|         6 | Laptop         | Thin and Light Laptop                 | 49500.00 | Electronics |
|         7 | Microwave Oven | Convection Microwave Oven             | 13200.00 | Electronics |
|         8 | Camera         | DSLR Camera with Kit Lens             | 38500.00 | Electronics |
|         9 | Water Purifier | RO + UV Water Purifier                | 13200.00 | Electronics |
|        10 | Vacuum Cleaner | Robotic Vacuum Cleaner                | 19800.00 | Electronics |
|        11 | Keyboard       | You say it is hopelesss.              |  2900.00 | Electronics |
|        12 | Smartwatch     | Fitness Tracker with Heart Rate Monitor |  7500.00 | Wearables   |
+-----------+----------------+---------------------------------------+---------+-------------+
12 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.

```
DELIMITER //

CREATE PROCEDURE UpdateOrderStatus(IN p_OrderID INT, IN p_NewStatus VARCHAR(20))
BEGIN
    -- Update the status in Orders table
    UPDATE Orders
    SET Status = p_NewStatus
    WHERE OrderID = p_OrderID;
END //

DELIMITER ;
```

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> Alter table Customers
    -> Add column NumOrders INT default 0;
Query OK, 0 rows affected (0.46 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from Customers;
+------------+-----------+------------+---------------------------+----------------+-------------------------------+-----------+
| CustomerID | FirstName | LastName   | Email                     | Phone          | Address                       | NumOrders |
+------------+-----------+------------+---------------------------+----------------+-------------------------------+-----------+
|          1 | Rahul     | Kumar      | rahul.kumar@example.com   | 9876543210     | 123, MG Road, Bangalore       |         0 |
|          2 | Priya     | Sharma     | priya.sharma@example.com  | 8765432109     | 456, Jubilee Hills, Hyderabad |         0 |
|          3 | Vikram    | Singh      | vikram.singh@example.com  | 7654321098     | 789, Malviya Nagar, Delhi     |         0 |
|          4 | Deepika   | Patel      | newemail@gmail.com        | +91 1111111111 | 101, Koregaon Park, Pune      |         0 |
|          5 | Amit      | Verma      | amit.verma@example.com    | 5432109876     | 234, Baner Road, Pune         |         0 |
|          6 | Ananya    | Nair       | ananya.nair@example.com   | 4321098765     | 567, Marathahalli, Bangalore  |         0 |
|          7 | Raj       | Malhotra   | raj.malhotra@example.com  | 3210987654     | 890, HSR Layout, Bangalore    |         0 |
|          8 | Neha      | Srivastava | neha.srivastava@example.com | 2109876543   | 123, Andheri West, Mumbai     |         0 |
|          9 | Sandeep   | Gupta      | sandeep.gupta@example.com | 1098765432     | 456, Aundh, Pune              |         0 |
|         10 | Shreya    | Rajput     | shreya.rajput@example.com | 9876543210     | 789, Banashankari, Bangalore  |         0 |
|         12 | Vatsal    | Patel      | vatsal.patel@email.com    | 9876543210     | 123, Main Street, Surat       |         0 |
+------------+-----------+------------+---------------------------+----------------+-------------------------------+-----------+
11 rows in set (0.00 sec)
```

```
mysql> -- Update the number of orders placed by each customer in Customers table
mysql> UPDATE Customers c
    -> SET NumOrders = (
    ->     SELECT COUNT(*)
    ->     FROM Orders o
    ->     WHERE o.CustomerID = c.CustomerID
    -> );
Query OK, 9 rows affected (0.09 sec)
Rows matched: 11  Changed: 9  Warnings: 0

mysql> select * from Customers;
+------------+-----------+------------+---------------------------+----------------+-------------------------------+-----------+
| CustomerID | FirstName | LastName   | Email                     | Phone          | Address                       | NumOrders |
+------------+-----------+------------+---------------------------+----------------+-------------------------------+-----------+
|          1 | Rahul     | Kumar      | rahul.kumar@example.com   | 9876543210     | 123, MG Road, Bangalore       |         2 |
|          2 | Priya     | Sharma     | priya.sharma@example.com  | 8765432109     | 456, Jubilee Hills, Hyderabad |         1 |
|          3 | Vikram    | Singh      | vikram.singh@example.com  | 7654321098     | 789, Malviya Nagar, Delhi     |         0 |
|          4 | Deepika   | Patel      | newemail@gmail.com        | +91 1111111111 | 101, Koregaon Park, Pune      |         1 |
|          5 | Amit      | Verma      | amit.verma@example.com    | 5432109876     | 234, Baner Road, Pune         |         1 |
|          6 | Ananya    | Nair       | ananya.nair@example.com   | 4321098765     | 567, Marathahalli, Bangalore  |         1 |
|          7 | Raj       | Malhotra   | raj.malhotra@example.com  | 3210987654     | 890, HSR Layout, Bangalore    |         1 |
|          8 | Neha      | Srivastava | neha.srivastava@example.com | 2109876543   | 123, Andheri West, Mumbai     |         1 |
|          9 | Sandeep   | Gupta      | sandeep.gupta@example.com | 1098765432     | 456, Aundh, Pune              |         1 |
|         10 | Shreya    | Rajput     | shreya.rajput@example.com | 9876543210     | 789, Banashankari, Bangalore  |         1 |
|         12 | Vatsal    | Patel      | vatsal.patel@email.com    | 9876543210     | 123, Main Street, Surat       |         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 8.0 Command Line Client
5 rows in set (2.27 sec)

mysql> SELECT o.OrderID, o.OrderDate, o.TotalAmount, c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone
    -> FROM Orders o
    -> JOIN Customers c ON o.CustomerID = c.CustomerID;
+---------+---------------------+-------------+------------+-----------+-----------+-----------------------------+----------------+
| OrderID | OrderDate           | TotalAmount | CustomerID | FirstName | LastName  | Email                       | Phone          |
+---------+---------------------+-------------+------------+-----------+-----------+-----------------------------+----------------+
|       1 | 2024-01-13 11:12:04 |   110000.00 |          1 | Rahul     | Kumar     | rahul.kumar@example.com     | 9876543210     |
|       2 | 2024-01-13 11:12:04 |    49500.00 |          2 | Priya     | Sharma    | priya.sharma@example.com    | 8765432109     |
|       4 | 2024-01-13 11:12:04 |    51700.00 |          4 | Deepika   | Patel     | newemail@gmail.com          | +91 1111111111 |
|       5 | 2024-01-13 11:12:04 |    46200.00 |          5 | Amit      | Verma     | amit.verma@example.com      | 5432109876     |
|       6 | 2024-01-13 11:12:04 |    71500.00 |          6 | Ananya    | Nair      | ananya.nair@example.com     | 4321098765     |
|       7 | 2024-01-13 11:12:04 |    49500.00 |          7 | Raj       | Malhotra  | raj.malhotra@example.com    | 3210987654     |
|       8 | 2024-01-13 11:12:04 |    66000.00 |          8 | Neha      | Srivastava | neha.srivastava@example.com | 2109876543     |
|       9 | 2024-01-13 11:12:04 |    51700.00 |          9 | Sandeep   | Gupta     | sandeep.gupta@example.com   | 1098765432     |
|      10 | 2024-01-13 11:12:04 |    46200.00 |         10 | Shreya    | Rajput    | shreya.rajput@example.com   | 9876543210     |
|      11 | 2024-01-13 12:33:22 |        0.00 |          1 | Rahul     | Kumar     | rahul.kumar@example.com     | 9876543210     |
+---------+---------------------+-------------+------------+-----------+-----------+-----------------------------+----------------+
10 rows in set (5.21 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.ProductID, p.ProductName, SUM(od.Quantity * p.Price) AS TotalRevenue
    -> FROM Products p
    -> JOIN OrderDetails od ON p.ProductID = od.ProductID
    -> GROUP BY p.ProductID, p.ProductName;
+-----------+-----------------+--------------+
| ProductID | ProductName     | TotalRevenue |
+-----------+-----------------+--------------+
|         1 | LED TV          |    115500.00 |
|         2 | Air Conditioner |     66000.00 |
|         3 | Washing Machine |     55000.00 |
|         4 | Refrigerator    |     44000.00 |
|         5 | Mobile Phone    |     16500.00 |
|         6 | Laptop          |     49500.00 |
|         7 | Microwave Oven  |     26400.00 |
|         8 | Camera          |     77000.00 |
|         9 | Water Purifier  |     52800.00 |
|        10 | Vacuum Cleaner  |     39600.00 |
+-----------+-----------------+--------------+
10 rows in set (0.31 sec)
```

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

MySQL 8.0 Command Line Client

```
mysql> SELECT c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone
    -> FROM Customers c
    -> JOIN Orders o ON c.CustomerID = o.CustomerID
    -> GROUP BY c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone;
+------------+-----------+------------+-----------------------------+----------------+
| CustomerID | FirstName | LastName   | Email                       | Phone          |
+------------+-----------+------------+-----------------------------+----------------+
|          1 | Rahul     | Kumar      | rahul.kumar@example.com     | 9876543210     |
|          2 | Priya     | Sharma     | priya.sharma@example.com    | 8765432109     |
|          4 | Deepika   | Patel      | newemail@gmail.com          | +91 1111111111 |
|          5 | Amit      | Verma      | amit.verma@example.com      | 5432109876     |
|          6 | Ananya    | Nair       | ananya.nair@example.com     | 4321098765     |
|          7 | Raj       | Malhotra   | raj.malhotra@example.com    | 3210987654     |
|          8 | Neha      | Srivastava | neha.srivastava@example.com | 2109876543     |
|          9 | Sandeep   | Gupta      | sandeep.gupta@example.com   | 1098765432     |
|         10 | Shreya    | Rajput     | shreya.rajput@example.com   | 9876543210     |
+------------+-----------+------------+-----------------------------+----------------+
9 rows in set (0.10 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 8.0 Command Line Client

```
9 rows in set (0.10 sec)

mysql> SELECT p.ProductID, p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered
    -> FROM Products p
    -> JOIN OrderDetails od ON p.ProductID = od.ProductID
    -> GROUP BY p.ProductID, p.ProductName
    -> ORDER BY TotalQuantityOrdered DESC
    -> LIMIT 1;
+-----------+---------------+----------------------+
| ProductID | ProductName   | TotalQuantityOrdered |
+-----------+---------------+----------------------+
|         9 | Water Purifier |                    4 |
+-----------+---------------+----------------------+
1 row in set (0.09 sec)
```

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

MySQL 8.0 Command Line Client

```
Empty set (0.10 sec)

mysql> SELECT p.ProductID, p.ProductName, p.Category
    -> FROM Products p
    -> WHERE p.Category = 'Electronics';
+-----------+-----------------+-------------+
| ProductID | ProductName     | Category    |
+-----------+-----------------+-------------+
|         1 | LED TV          | Electronics |
|         2 | Air Conditioner | Electronics |
|         3 | Washing Machine | Electronics |
|         4 | Refrigerator    | Electronics |
|         5 | Mobile Phone    | Electronics |
|         6 | Laptop          | Electronics |
|         7 | Microwave Oven  | Electronics |
|         8 | Camera          | Electronics |
|         9 | Water Purifier  | Electronics |
|        10 | Vacuum Cleaner  | Electronics |
|        11 | Keyboard        | Electronics |
+-----------+-----------------+-------------+
11 rows in set (0.00 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> SELECT c.CustomerID, c.FirstName, c.LastName, AVG(o.TotalAmount) AS AverageOrderValue
    -> FROM Customers c
    -> JOIN Orders o ON c.CustomerID = o.CustomerID
    -> GROUP BY c.CustomerID, c.FirstName, c.LastName;
+------------+-----------+------------+-------------------+
| CustomerID | FirstName | LastName   | AverageOrderValue |
+------------+-----------+------------+-------------------+
|          1 | Rahul     | Kumar      |      55000.000000 |
|          2 | Priya     | Sharma     |      49500.000000 |
|          4 | Deepika   | Patel      |      51700.000000 |
|          5 | Amit      | Verma      |      46200.000000 |
|          6 | Ananya    | Nair       |      71500.000000 |
|          7 | Raj       | Malhotra   |      49500.000000 |
|          8 | Neha      | Srivastava |      66000.000000 |
|          9 | Sandeep   | Gupta      |      51700.000000 |
|         10 | Shreya    | Rajput     |      46200.000000 |
+------------+-----------+------------+-------------------+
9 rows in set (0.37 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, o.OrderDate, o.TotalAmount, c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone
    -> FROM Orders o
    -> JOIN Customers c ON o.CustomerID = c.CustomerID
    -> ORDER BY o.TotalAmount DESC
    -> LIMIT 1;
+---------+---------------------+-------------+------------+-----------+----------+-------------------------+------------+
| OrderID | OrderDate           | TotalAmount | CustomerID | FirstName | LastName | Email                   | Phone      |
+---------+---------------------+-------------+------------+-----------+----------+-------------------------+------------+
|       1 | 2024-01-13 11:12:04 |   110000.00 |          1 | Rahul     | Kumar    | rahul.kumar@example.com | 9876543210 |
+---------+---------------------+-------------+------------+-----------+----------+-------------------------+------------+
1 row in set (0.34 sec)
```

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

```
mysql> SELECT p.ProductID, p.ProductName, COUNT(od.OrderDetailID) AS OrderCount
    -> FROM Products p
    -> LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
    -> GROUP BY p.ProductID, p.ProductName;
+-----------+-----------------+------------+
| ProductID | ProductName     | OrderCount |
+-----------+-----------------+------------+
|         1 | LED TV          |          2 |
|         2 | Air Conditioner |          2 |
|         3 | Washing Machine |          2 |
|         4 | Refrigerator    |          2 |
|         5 | Mobile Phone    |          1 |
|         6 | Laptop          |          1 |
|         7 | Microwave Oven  |          2 |
|         8 | Camera          |          2 |
|         9 | Water Purifier  |          2 |
|        10 | Vacuum Cleaner  |          2 |
|        11 | Keyboard        |          0 |
|        12 | Smartwatch      |          0 |
+-----------+-----------------+------------+
12 rows in set (1.24 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.

```
1   DELIMITER //
2
3   CREATE PROCEDURE findCustomerByProductName(IN p_name VARCHAR(20))
4   BEGIN
5       SELECT c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone
6       FROM Customers c
7       JOIN Orders o ON c.CustomerID = o.CustomerID
8       JOIN OrderDetails od ON o.OrderID = od.OrderID
9       JOIN Products p ON od.ProductID = p.ProductID
10      WHERE p.ProductName = p_name
11      GROUP BY c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone;
12  END //
13
14  DELIMITER ;
```

MySQL 8.0 Command Line Client

```
Query OK, 0 rows affected (0.00 sec)

mysql> CALL findCustomerByProductName('LED TV');
+------------+-----------+----------+-------------------------+------------+
| CustomerID | FirstName | LastName | Email                   | Phone      |
+------------+-----------+----------+-------------------------+------------+
|          1 | Rahul     | Kumar    | rahul.kumar@example.com | 9876543210 |
|          6 | Ananya    | Nair     | ananya.nair@example.com | 4321098765 |
+------------+-----------+----------+-------------------------+------------+
2 rows in set (0.04 sec)

Query OK, 0 rows affected (0.05 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.

```
1   DELIMITER //
2
3   CREATE PROCEDURE getTotalRevenueByDateRange(IN p_StartDate DATE, IN p_EndDate DATE)
4   BEGIN
5       IF p_EndDate IS NULL THEN
6           SET p_EndDate = NOW();
7       END IF;
8       SELECT SUM(o.TotalAmount) AS TotalRevenue
9       FROM Orders o
10      WHERE o.OrderDate BETWEEN p_StartDate AND p_EndDate;
11  END //
12
13  DELIMITER ;
```

MySQL 8.0 Command Line Client

```
mysql> select * from orders
    -> ;
+---------+------------+---------------------+-------------+-----------+
| OrderID | CustomerID | OrderDate           | TotalAmount | Status    |
+---------+------------+---------------------+-------------+-----------+
|       1 |          1 | 2024-01-13 11:12:04 |   110000.00 | Pending   |
|       2 |          2 | 2024-01-13 11:12:04 |    49500.00 | Shipped   |
|       4 |          4 | 2024-01-13 11:12:04 |    51700.00 | Pending   |
|       5 |          5 | 2024-01-13 11:12:04 |    46200.00 | Shipped   |
|       6 |          6 | 2024-01-13 11:12:04 |    71500.00 | Delivered |
|       7 |          7 | 2024-01-13 11:12:04 |    49500.00 | Pending   |
|       8 |          8 | 2024-01-13 11:12:04 |    66000.00 | Shipped   |
|       9 |          9 | 2024-01-13 11:12:04 |    51700.00 | Delivered |
|      10 |         10 | 2024-01-13 11:12:04 |    46200.00 | Pending   |
|      11 |          1 | 2024-01-13 12:33:22 |        0.00 | Pending   |
+---------+------------+---------------------+-------------+-----------+
10 rows in set (0.00 sec)

mysql> CALL getTotalRevenueByDateRange('2024-01-01', '2024-12-31');
+--------------+
| TotalRevenue |
+--------------+
|    542300.00 |
+--------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Task 4. Subquery and its type:

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

```
mysql> SELECT *
    -> FROM Customers c
    -> WHERE NOT EXISTS (
    ->     SELECT 1
    ->     FROM Orders o
    ->     WHERE o.CustomerID = c.CustomerID
    -> );
+------------+-----------+----------+--------------------------+------------+--------------------------+-----------+
| CustomerID | FirstName | LastName | Email                    | Phone      | Address                  | NumOrders |
+------------+-----------+----------+--------------------------+------------+--------------------------+-----------+
|          3 | Vikram    | Singh    | vikram.singh@example.com | 7654321098 | 789, Malviya Nagar, Delhi |         0 |
|         12 | Vatsal    | Patel    | vatsal.patel@email.com   | 9876543210 | 123, Main Street, Surat   |         0 |
+------------+-----------+----------+--------------------------+------------+--------------------------+-----------+
2 rows in set (20.30 sec)
```

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

```
mysql> SELECT COUNT(*) AS TotalProducts
    -> FROM Products;
+---------------+
| TotalProducts |
+---------------+
|            12 |
+---------------+
1 row in set (11.39 sec)
```

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

```
mysql> SELECT SUM(o.TotalAmount) AS TotalRevenue
    -> FROM Orders o;
+--------------+
| TotalRevenue |
+--------------+
|    542300.00 |
+--------------+
1 row in set (0.88 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.

```
DELIMITER //

CREATE PROCEDURE getAverageQuantityOrderedByCategory(IN category VARCHAR(255))
BEGIN
    SELECT AVG(od.Quantity) AS AverageQuantityOrdered
    FROM OrderDetails od
    JOIN Products p ON od.ProductID = p.ProductID
    WHERE p.Category = category;
END //

DELIMITER ;
```

```
mysql> CALL getAverageQuantityOrderedByCategory("Electronics");
+-------------------------+
| AverageQuantityOrdered |
+-------------------------+
|                  1.1667 |
+-------------------------+
1 row in set (1.07 sec)

Query OK, 0 rows affected (1.08 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.

```
DELIMITER //

CREATE PROCEDURE getTotalRevenueByCustoemrID(IN customerID INT)
BEGIN
    SELECT SUM(o.TotalAmount) AS TotalRevenue
    FROM Orders o
    WHERE o.CustomerID = customerID;
END //

DELIMITER ;
```

```
mysql> CALL getTotalRevenueByCustoemrID(2);
+--------------+
| TotalRevenue |
+--------------+
|     49500.00 |
+--------------+
1 row in set (0.53 sec)

Query OK, 0 rows affected (0.54 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(o.OrderID) AS OrderCount
    -> FROM Customers c
    -> JOIN Orders o ON c.CustomerID = o.CustomerID
    -> GROUP BY c.CustomerID
    -> ORDER BY OrderCount DESC
    -> LIMIT 1;
+-----------+----------+------------+
| FirstName | LastName | OrderCount |
+-----------+----------+------------+
| Rahul     | Kumar    |          2 |
+-----------+----------+------------+
1 row in set (1.09 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 p.Category, SUM(od.Quantity) AS TotalQuantityOrdered
    -> FROM OrderDetails od
    -> JOIN Products p ON od.ProductID = p.ProductID
    -> GROUP BY p.Category
    -> ORDER BY TotalQuantityOrdered DESC
    -> LIMIT 1;
+-------------+----------------------+
| Category    | TotalQuantityOrdered |
+-------------+----------------------+
| Electronics |                   21 |
+-------------+----------------------+
1 row in set (0.01 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(o.TotalAmount) AS TotalSpending
    -> 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.Category = 'Electronics'
    -> GROUP BY c.CustomerID
    -> ORDER BY TotalSpending DESC
    -> LIMIT 1;
+-----------+----------+---------------+
| FirstName | LastName | TotalSpending |
+-----------+----------+---------------+
| Rahul     | Kumar    |     220000.00 |
+-----------+----------+---------------+
1 row in set (0.94 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 AVG(o.TotalAmount) AS AverageOrderValue
    -> FROM Orders o;
+-------------------+
| AverageOrderValue |
+-------------------+
|      54230.000000 |
+-------------------+
1 row 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(o.OrderID) AS OrderCount
    -> FROM Customers c
    -> LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
    -> GROUP BY c.CustomerID;
+-----------+------------+------------+
| FirstName | LastName   | OrderCount |
+-----------+------------+------------+
| Rahul     | Kumar      |          2 |
| Priya     | Sharma     |          1 |
| Vikram    | Singh      |          0 |
| Deepika   | Patel      |          1 |
| Amit      | Verma      |          1 |
| Ananya    | Nair       |          1 |
| Raj       | Malhotra   |          1 |
| Neha      | Srivastava |          1 |
| Sandeep   | Gupta      |          1 |
| Shreya    | Rajput     |          1 |
| Vatsal    | Patel      |          0 |
+-----------+------------+------------+
11 rows in set (0.06 sec)
```