Task:1. Database Design

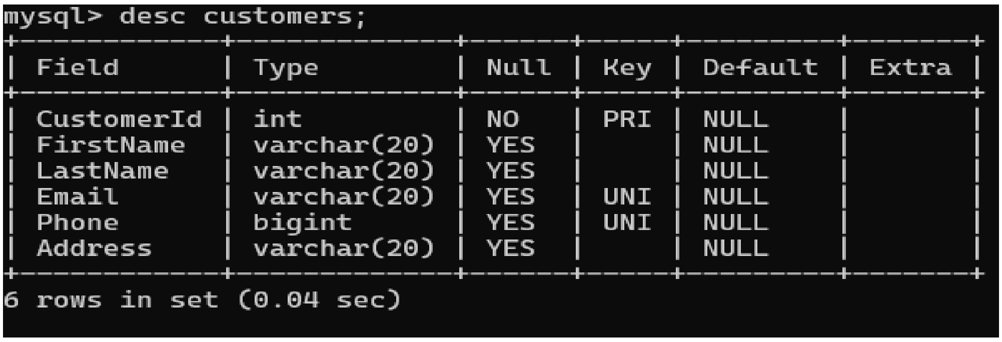
1. Create the database named "TechShop"

```
=>CREATE DATABASE TechShop;
  Use Techshop;
```

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

=>Customers Table

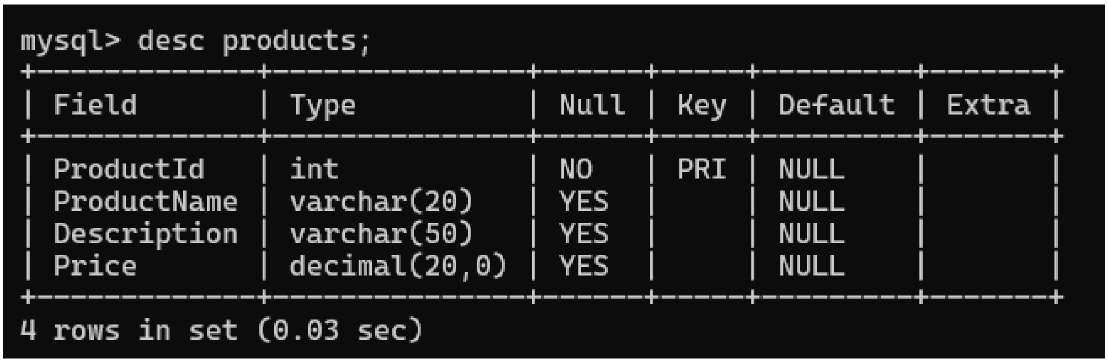
```
 CREATE TABLE Customers(
                    CustomerId integer(5),
                    FirstName varchar(20),
                    LastName  Varchar(20),
                    Email varchar(20) UNIQUE ,
                    Phone bigint(15) UNIQUE,
                    Address Varchar(20),
                      constraint customers_customerId_pk primary key(customerId)
                    );
```



```
mysql> desc customers;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| CustomerId | int         | NO   | PRI | NULL    |       |
| FirstName  | varchar(20) | YES  |     | NULL    |       |
| LastName   | varchar(20) | YES  |     | NULL    |       |
| Email      | varchar(20) | YES  | UNI | NULL    |       |
| Phone      | bigint      | YES  | UNI | NULL    |       |
| Address    | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.04 sec)
```

=>Products Table

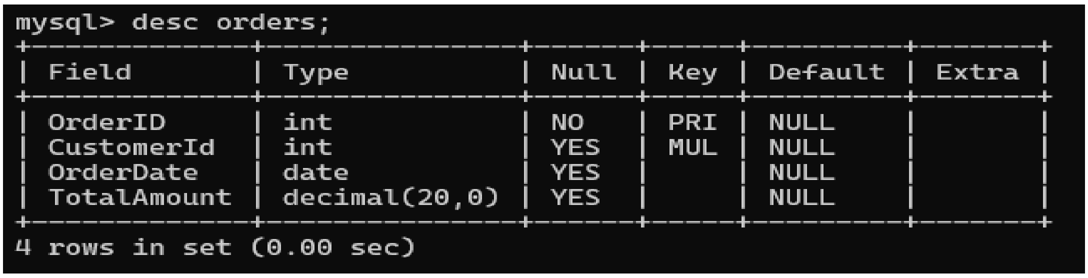
```
 CREATE TABLE Products(
                                 ProductId integer(8),
                                  ProductName varchar(20),
                                 Description Varchar(50),
                                 Price Decimal(20) ,
                                 constraint products_ProductId_pk primary key(productId)
                                 );
```



```
mysql> desc products;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| ProductId   | int           | NO   | PRI | NULL    |       |
| ProductName | varchar(20)   | YES  |     | NULL    |       |
| Description | varchar(50)   | YES  |     | NULL    |       |
| Price       | decimal(20,0) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.03 sec)
```

=>Orders Table

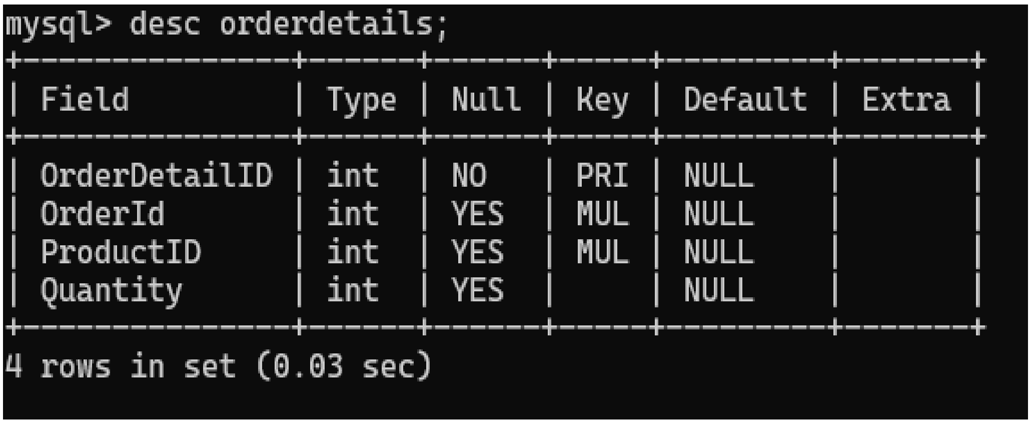
```
CREATE TABLE Orders(
                    OrderID integer(8),
                    CustomerId integer(5) ,
                    OrderDate Date,
                    TotalAmount Decimal(20),
                            constraint orders_orderId_pk primary key(orderId),
                    constraint orders_customerId_fk foreign key(customerId)
references customers(customerId)
                    );
```



```
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(20,0) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

=>OrderDetails Table

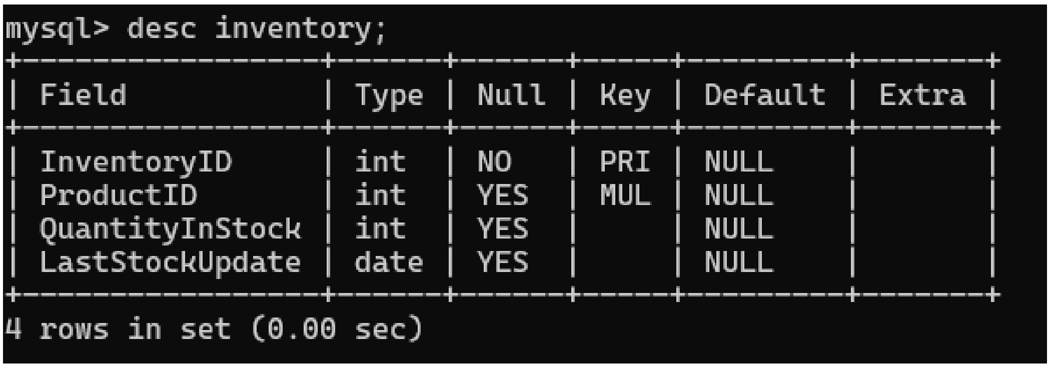
```
CREATE TABLE OrderDetails(
                    OrderDetailID integer(10),
                    constraint ordersDetails_OrderDetailID_pk primary
key(OrderDetailID),
                    OrderId integer(8),
                    constraint ordersDetails_orderId_fk foreign key(orderId) REFERENCES Orders(OrderID),

                    ProductID integer(8),

                    constraint ordersDetails_productId_fk foreign key(productId) REFERENCES Products(ProductId),

                    Quantity Integer(50)

                    );
```



```
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 (0.03 sec)
```

=>Inventory Table

```
CREATE TABLE Inventory(

                        InventoryID integer(10),

                        constraint Inventory_InventoryID_pk Primary Key(InventoryID),

                        ProductID integer(8),

                        constraint Inventory_productId_fk foreign key(productId) REFERENCES Products(ProductID),

                        QuantityInStock integer(100),

                        LastStockUpdate Date

                        );
```



```
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)
```

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

a. Customer :

insert into customers values(101,'Arav','Josh','arav@gmailcom',9876543212,'Bangalore');

insert into customers values(102,'Harsh','Singh','harsh@gmailcom',9865433276,'Hyderbad');

insert into customers values(103,'Siya','Gupta','siya@gmailcom',9766578654,'Mumbai');

insert into customers values(104,'Siri','Kumar','siri@gmailcom',9776589076,'Pune');

insert into customers values(105,'Sam','sharma','sam@gmailcom',9765432178,'Bangalore');

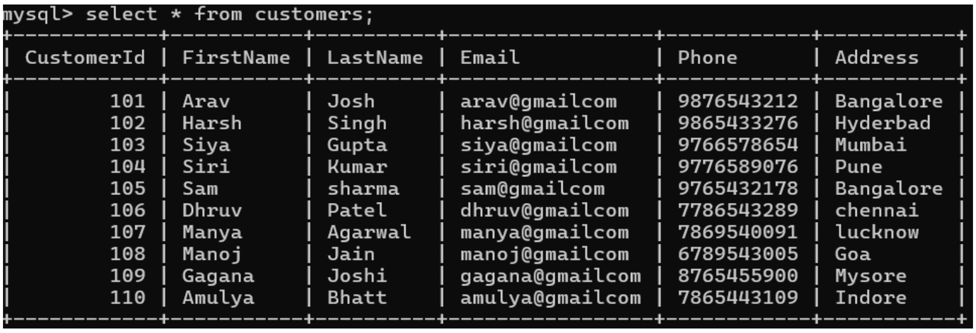
insert into customers values(106,'Dhruv','Patel','dhruv@gmailcom',7786543289,'chennai');

insert into customers values(107,'Manya','Agarwal','manya@gmailcom',7869540091,'lucknow');

insert into customers values(108,'Manoj','Jain','manoj@gmailcom',6789543005,'Goa');

insert into customers values(109,'Gagana','Joshi','gagana@gmailcom',8765455900,'Mysore');

insert into customers values(110,'Amulya','Bhatt','amulya@gmailcom',7865443109,'Indore');



```
mysql> select * from customers;
+------------+-----------+----------+------------------+------------+-----------+
| CustomerId | FirstName | LastName | Email            | Phone      | Address   |
+------------+-----------+----------+------------------+------------+-----------+
|        101 | Arav      | Josh     | arav@gmailcom    | 9876543212 | Bangalore |
|        102 | Harsh     | Singh    | harsh@gmailcom   | 9865433276 | Hyderbad  |
|        103 | Siya      | Gupta    | siya@gmailcom    | 9766578654 | Mumbai    |
|        104 | Siri      | Kumar    | siri@gmailcom    | 9776589076 | Pune      |
|        105 | Sam       | sharma   | sam@gmailcom     | 9765432178 | Bangalore |
|        106 | Dhruv     | Patel    | dhruv@gmailcom   | 7786543289 | chennai   |
|        107 | Manya     | Agarwal  | manya@gmailcom   | 7869540091 | lucknow   |
|        108 | Manoj     | Jain     | manoj@gmailcom   | 6789543005 | Goa       |
|        109 | Gagana    | Joshi    | gagana@gmailcom  | 8765455900 | Mysore    |
|        110 | Amulya    | Bhatt    | amulya@gmailcom  | 7865443109 | Indore    |
+------------+-----------+----------+------------------+------------+-----------+
```

b.Products :

insert into products values(1,'Phone','samsung',100000);

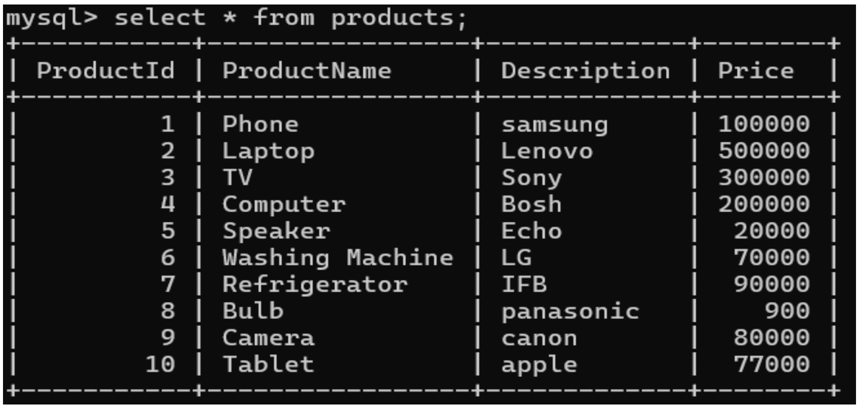
insert into products values(2,'Laptop','Lenovo',500000);

insert into products values(3,'TV','Sony',300000);

insert into products values(4,'Computer','Bosh',200000);

insert into products values(5,'Speaker','Echo',20000);

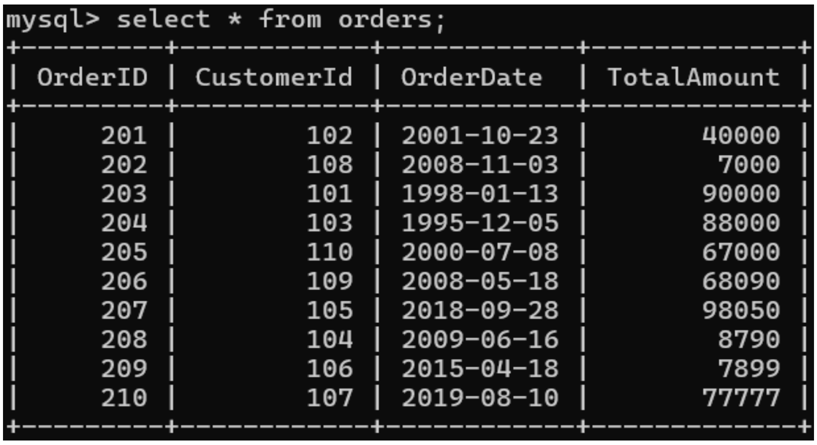
```
insert into products values(6,'Washing Machine','LG',70000);

insert into products values(7,'Refrigerator','IFB',90000);

insert into products values(8,'Bulb','panasonic',900);

insert into products values(9,'Camera','canon',80000);

insert into products values(10,'Tablet','apple',77000);
```



```
mysql> select * from products;
+-----------+-----------------+-------------+--------+
| ProductId | ProductName     | Description | Price  |
+-----------+-----------------+-------------+--------+
|         1 | Phone           | samsung     | 100000 |
|         2 | Laptop          | Lenovo      | 500000 |
|         3 | TV              | Sony        | 300000 |
|         4 | Computer        | Bosh        | 200000 |
|         5 | Speaker         | Echo        |  20000 |
|         6 | Washing Machine | LG          |  70000 |
|         7 | Refrigerator    | IFB         |  90000 |
|         8 | Bulb            | panasonic   |    900 |
|         9 | Camera          | canon       |  80000 |
|        10 | Tablet          | apple       |  77000 |
+-----------+-----------------+-------------+--------+
```

c.Orders :

```
insert into orders values(201,102,'2001-10-23',40000);

insert into orders values(202,108,'2008-11-03',7000);

insert into orders values(203,101,'1998-01-13',90000);

insert into orders values(204,103,'1995-12-05',88000);

insert into orders values(205,110,'2000-07-08',67000);

insert into orders values(206,109,'2008-05-18',68090);

insert into orders values(207,105,'2018-09-28',98050);

insert into orders values(208,104,'2009-06-16',8790);

insert into orders values(209,106,'2015-04-18',7899);

insert into orders values(210,107,'2019-08-10',77777);
```



```
mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerId | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     201 |        102 | 2001-10-23 |       40000 |
|     202 |        108 | 2008-11-03 |        7000 |
|     203 |        101 | 1998-01-13 |       90000 |
|     204 |        103 | 1995-12-05 |       88000 |
|     205 |        110 | 2000-07-08 |       67000 |
|     206 |        109 | 2008-05-18 |       68090 |
|     207 |        105 | 2018-09-28 |       98050 |
|     208 |        104 | 2009-06-16 |        8790 |
|     209 |        106 | 2015-04-18 |        7899 |
|     210 |        107 | 2019-08-10 |       77777 |
+---------+------------+------------+-------------+
```

c.Ordersdetails :

insert into orderdetails values(1001,202,4,2);

insert into orderdetails values(1002,205,3,3);

insert into orderdetails values(1003,201,1,5);

insert into orderdetails values(1004,210,10,2);

insert into orderdetails values(1005,209,9,1);

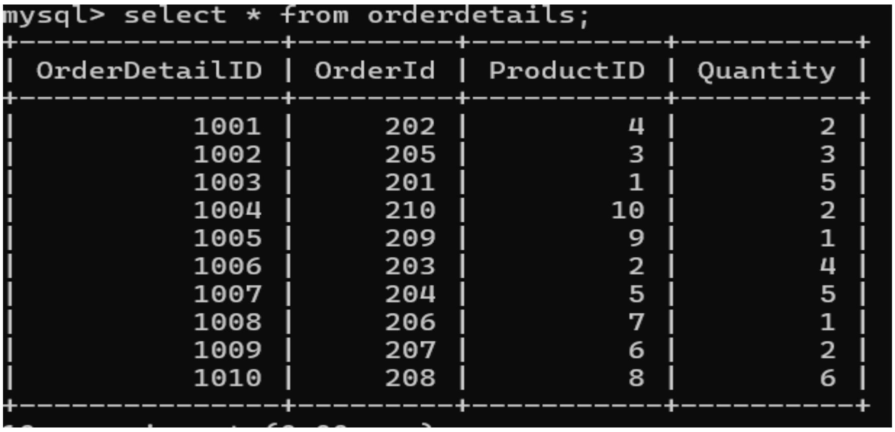
insert into orderdetails values(1006,203,2,4);

insert into orderdetails values(1007,204,5,5);

insert into orderdetails values(1008,206,7,1);

insert into orderdetails values(1009,207,6,2);

insert into orderdetails values(1010,208,8,6);



```
mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderId | ProductID | Quantity |
+---------------+---------+-----------+----------+
|          1001 |     202 |         4 |        2 |
|          1002 |     205 |         3 |        3 |
|          1003 |     201 |         1 |        5 |
|          1004 |     210 |        10 |        2 |
|          1005 |     209 |         9 |        1 |
|          1006 |     203 |         2 |        4 |
|          1007 |     204 |         5 |        5 |
|          1008 |     206 |         7 |        1 |
|          1009 |     207 |         6 |        2 |
|          1010 |     208 |         8 |        6 |
+---------------+---------+-----------+----------+
```

d.Inventory :

insert into inventory values (2001,3,10, ‘2024-01-18’);

insert into inventory values (2002,2,15 , ‘2023-12-28’);

insert into inventory values (2003,1,18 , ‘2023-11-12’);

insert into inventory values (2004,10,12 , ‘2024-03-10’);

insert into inventory values (2005,9,11, ‘2023-12-12’);

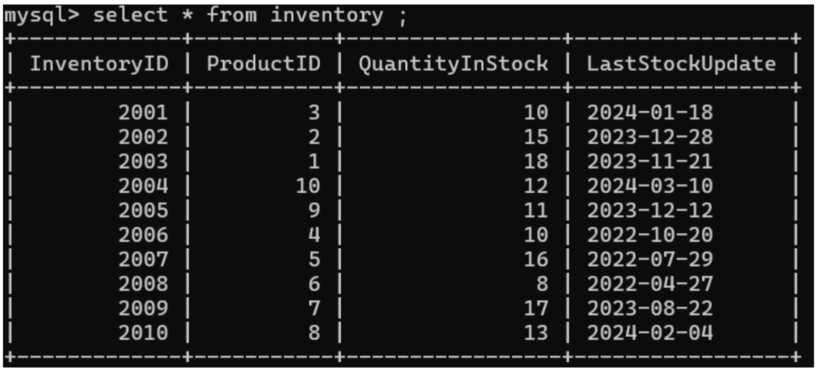
insert into inventory values (2006,4,10 , ‘2022-10-20’);

insert into inventory values (2007,5,16 , ‘2022-07-29’);

insert into inventory values (2008,6,8 , ‘2022-04-27’);

insert into inventory values (2009,7,17 , ‘2023-08-22’);

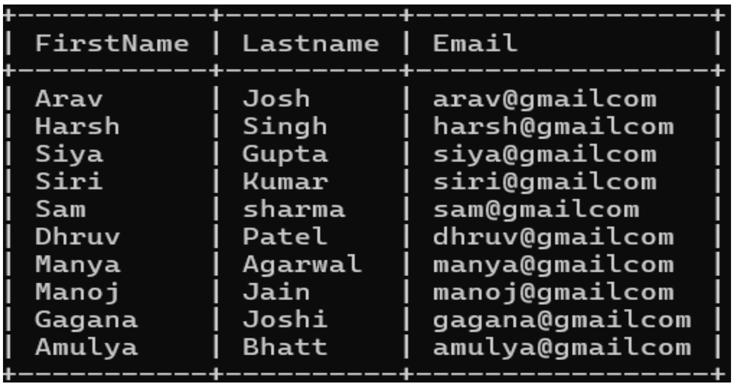
insert into inventory values (2010,8,13 , ‘2024-02-04’);



```
mysql> select * from inventory ;
+-------------+-----------+-----------------+-----------------+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-------------+-----------+-----------------+-----------------+
|        2001 |         3 |              10 | 2024-01-18      |
|        2002 |         2 |              15 | 2023-12-28      |
|        2003 |         1 |              18 | 2023-11-21      |
|        2004 |        10 |              12 | 2024-03-10      |
|        2005 |         9 |              11 | 2023-12-12      |
|        2006 |         4 |              10 | 2022-10-20      |
|        2007 |         5 |              16 | 2022-07-29      |
|        2008 |         6 |               8 | 2022-04-27      |
|        2009 |         7 |              17 | 2023-08-22      |
|        2010 |         8 |              13 | 2024-02-04      |
+-------------+-----------+-----------------+-----------------+
```

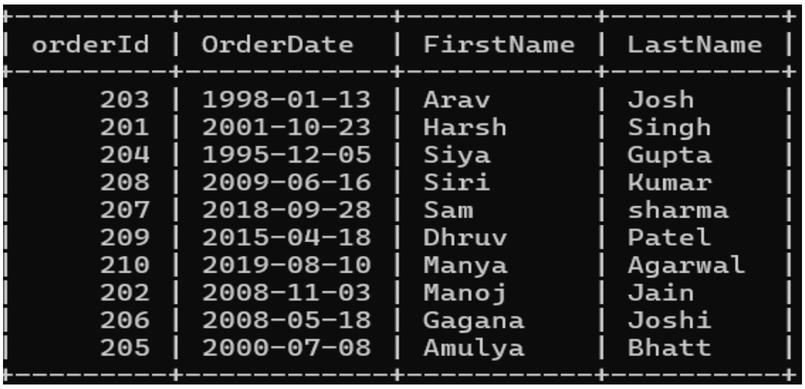
Tasks 2: Select, Where, Between, AND, LIKE:

1. Write an SQL query to retrieve the names and emails of all customers.
   ->select FirstName, Lastname, Email from customers;



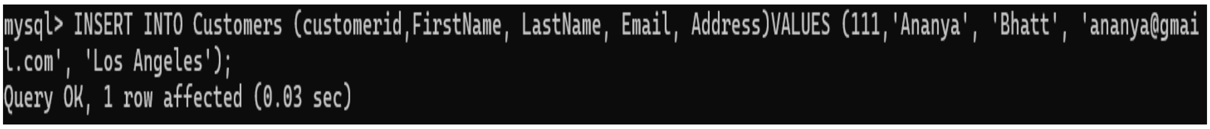
```
+-----------+----------+-----------------+
| FirstName | Lastname | Email           |
+-----------+----------+-----------------+
| Arav      | Josh     | arav@gmailcom   |
| Harsh     | Singh    | harsh@gmailcom  |
| Siya      | Gupta    | siya@gmailcom   |
| Siri      | Kumar    | siri@gmailcom   |
| Sam       | sharma   | sam@gmailcom    |
| Dhruv     | Patel    | dhruv@gmailcom  |
| Manya     | Agarwal  | manya@gmailcom  |
| Manoj     | Jain     | manoj@gmailcom  |
| Gagana    | Joshi    | gagana@gmailcom |
| Amulya    | Bhatt    | amulya@gmailcom |
+-----------+----------+-----------------+
```

2. Write an SQL query to list all orders with their order dates and corresponding customer names.
   -> select o.orderId ,o.OrderDate ,c.FirstName,c.LastName from orders o ,customers   c where o.customerid=c.customerid;



```
+---------+------------+-----------+----------+
| orderId | OrderDate  | FirstName | LastName |
+---------+------------+-----------+----------+
|     203 | 1998-01-13 | Arav      | Josh     |
|     201 | 2001-10-23 | Harsh     | Singh    |
|     204 | 1995-12-05 | Siya      | Gupta    |
|     208 | 2009-06-16 | Siri      | Kumar    |
|     207 | 2018-09-28 | Sam       | sharma   |
|     209 | 2015-04-18 | Dhruv     | Patel    |
|     210 | 2019-08-10 | Manya     | Agarwal  |
|     202 | 2008-11-03 | Manoj     | Jain     |
|     206 | 2008-05-18 | Gagana    | Joshi    |
|     205 | 2000-07-08 | Amulya    | Bhatt    |
+---------+------------+-----------+----------+
```

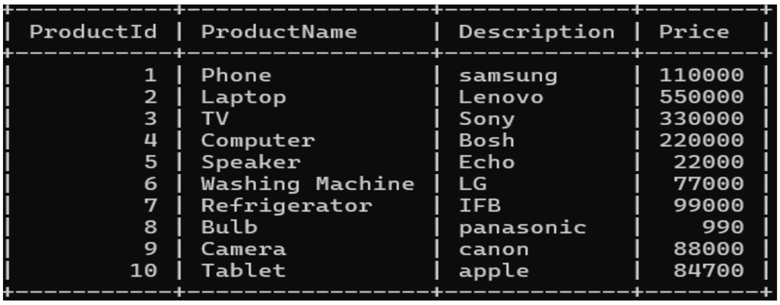
3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.
   -> INSERT INTO Customers (customerid,FirstName, LastName, Email, Address)VALUES (111,'Ananya', 'Bhatt', 'ananya@gmai
   l.com', 'Los Angeles');



```
mysql> INSERT INTO Customers (customerid,FirstName, LastName, Email, Address)VALUES (111,'Ananya', 'Bhatt', 'ananya@gmai
l.com', 'Los Angeles');
Query OK, 1 row affected (0.03 sec)
```

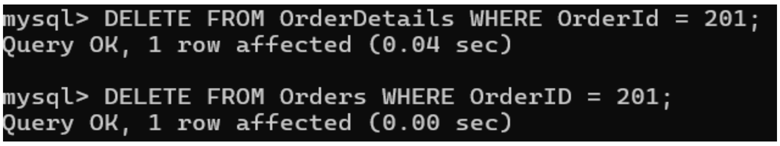
4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.

   ->UPDATE Products
   -> SET Price = Price * 1.10;
   select * from products;



```
+-----------+-----------------+-------------+--------+
| ProductId | ProductName     | Description | Price  |
+-----------+-----------------+-------------+--------+
|         1 | Phone           | samsung     | 110000 |
|         2 | Laptop          | Lenovo      | 550000 |
|         3 | TV              | Sony        | 330000 |
|         4 | Computer        | Bosh        | 220000 |
|         5 | Speaker         | Echo        |  22000 |
|         6 | Washing Machine | LG          |  77000 |
|         7 | Refrigerator    | IFB         |  99000 |
|         8 | Bulb            | panasonic   |    990 |
|         9 | Camera          | canon       |  88000 |
|        10 | Tablet          | apple       |  84700 |
+-----------+-----------------+-------------+--------+
```

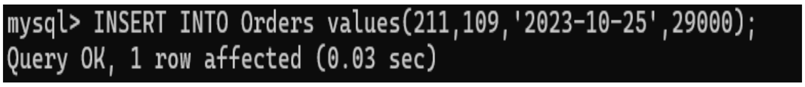
5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.
   -> DELETE FROM OrderDetails WHERE OrderId = 201;
   -> DELETE FROM Orders WHERE OrderID = 201;



```
mysql> DELETE FROM OrderDetails WHERE OrderId = 201;
Query OK, 1 row affected (0.04 sec)

mysql> DELETE FROM Orders WHERE OrderID = 201;
Query OK, 1 row affected (0.00 sec)
```

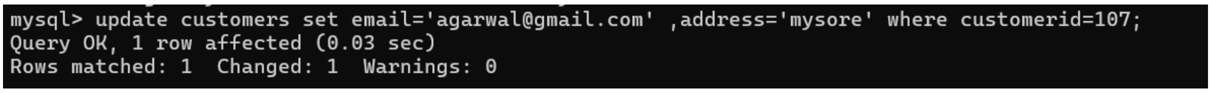
6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.

   ->INSERT INTO Orders values(211,109,'2023-10-25',29000);

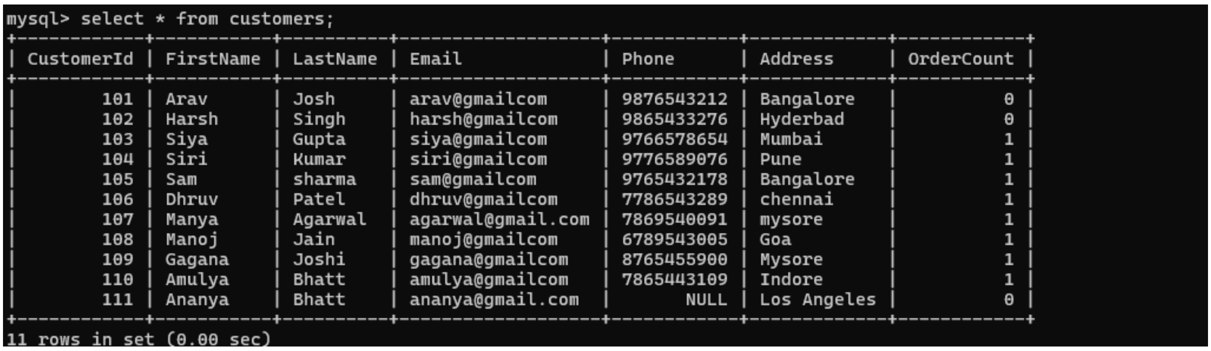


```
mysql> INSERT INTO Orders values(211,109,'2023-10-25',29000);
Query OK, 1 row affected (0.03 sec)
```

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.
   -> update customers set email='agarwal@gmail.com' ,address='mysore' where customerid=107;



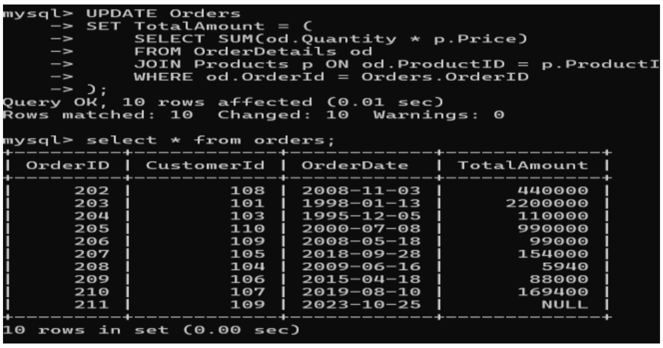
```
mysql> update customers set email='agarwal@gmail.com' ,address='mysore' where customerid=107;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```



```
mysql> select * from customers;
+------------+-----------+----------+-------------------+------------+-------------+------------+
| CustomerId | FirstName | LastName | Email             | Phone      | Address     | OrderCount |
+------------+-----------+----------+-------------------+------------+-------------+------------+
|        101 | Arav      | Josh     | arav@gmailcom     | 9876543212 | Bangalore   |          0 |
|        102 | Harsh     | Singh    | harsh@gmailcom    | 9865433276 | Hyderbad    |          0 |
|        103 | Siya      | Gupta    | siya@gmailcom     | 9766578654 | Mumbai      |          1 |
|        104 | Siri      | Kumar    | siri@gmailcom     | 9776589076 | Pune        |          1 |
|        105 | Sam       | sharma   | sam@gmailcom      | 9765432178 | Bangalore   |          1 |
|        106 | Dhruv     | Patel    | dhruv@gmailcom    | 7786543289 | chennai     |          1 |
|        107 | Manya     | Agarwal  | agarwal@gmail.com | 7869540091 | mysore      |          1 |
|        108 | Manoj     | Jain     | manoj@gmailcom    | 6789543005 | Goa         |          1 |
|        109 | Gagana    | Joshi    | gagana@gmailcom   | 8765455900 | Mysore      |          1 |
|        110 | Amulya    | Bhatt    | amulya@gmailcom   | 7865443109 | Indore      |          1 |
|        111 | Ananya    | Bhatt    | ananya@gmail.com  |       NULL | Los Angeles |          0 |
+------------+-----------+----------+-------------------+------------+-------------+------------+
11 rows in set (0.00 sec)
```

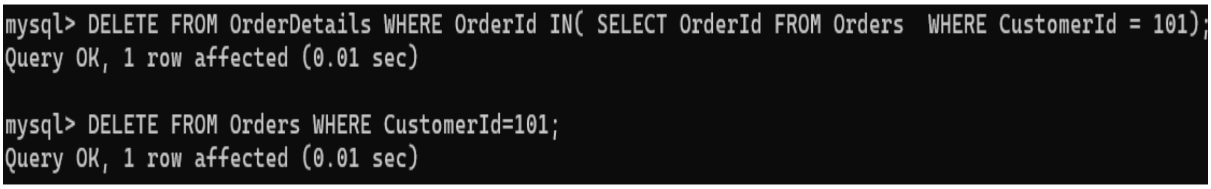
8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.
   UPDATE Orders

```
   -> SET TotalAmount = (
   ->     SELECT SUM(od.Quantity * p.Price)
   ->     FROM OrderDetails od
   ->     JOIN Products p ON od.ProductID = p.ProductId
   ->     WHERE od.OrderId = Orders.OrderID
   -> );
```



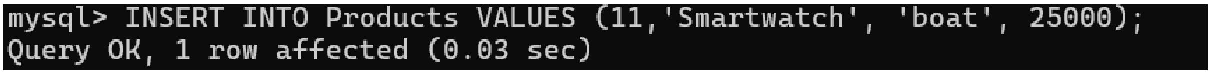
```
mysql> UPDATE Orders
    -> SET TotalAmount = (
    ->     SELECT SUM(od.Quantity * p.Price)
    ->     FROM OrderDetails od
    ->     JOIN Products p ON od.ProductID = p.ProductI
    ->     WHERE od.OrderId = Orders.OrderID
    -> );
Query OK, 10 rows affected (0.01 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerId | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     202 |        108 | 2008-11-03 |      440000 |
|     203 |        101 | 1998-01-13 |     2200000 |
|     204 |        103 | 1995-12-05 |      110000 |
|     205 |        110 | 2000-07-08 |      990000 |
|     206 |        109 | 2008-05-18 |       99000 |
|     207 |        105 | 2018-09-28 |      154000 |
|     208 |        104 | 2009-06-16 |        5940 |
|     209 |        106 | 2015-04-18 |       88000 |
|     210 |        107 | 2019-08-10 |      169400 |
|     211 |        109 | 2023-10-25 |        NULL |
+---------+------------+------------+-------------+
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.
   ->DELETE FROM OrderDetails WHERE OrderId IN(SELECT OrderId FROM Orders WHERE CustomerId = 101);
   ->DELETE FROM Orders WHERE CustomerId=101;



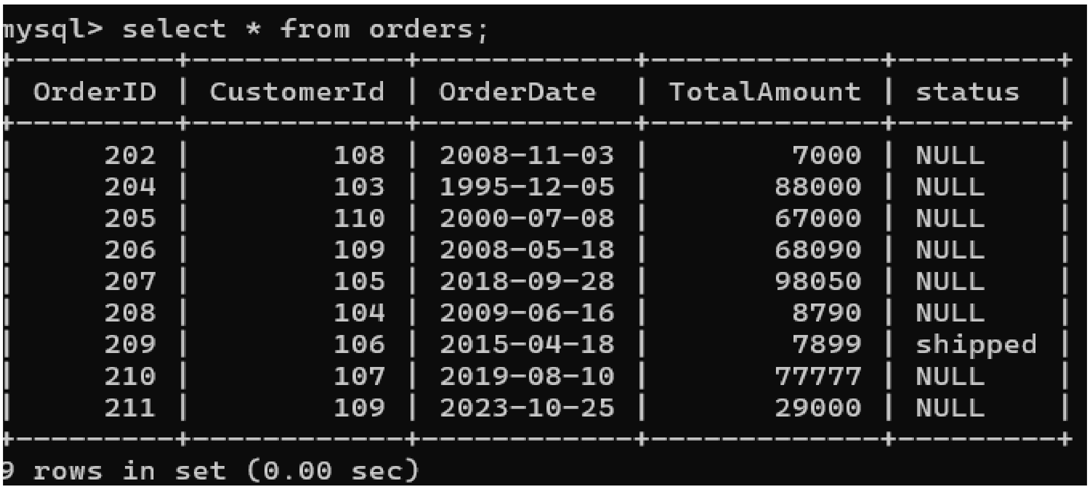
```
mysql> DELETE FROM OrderDetails WHERE OrderId IN( SELECT OrderId FROM Orders  WHERE CustomerId = 101);
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM Orders WHERE CustomerId=101;
Query OK, 1 row affected (0.01 sec)
```

10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.
    -> INSERT INTO Products VALUES (11,'Smartwatch', 'boat', 25000);



```
mysql> INSERT INTO Products VALUES (11,'Smartwatch', 'boat', 25000);
Query OK, 1 row affected (0.03 sec)
```

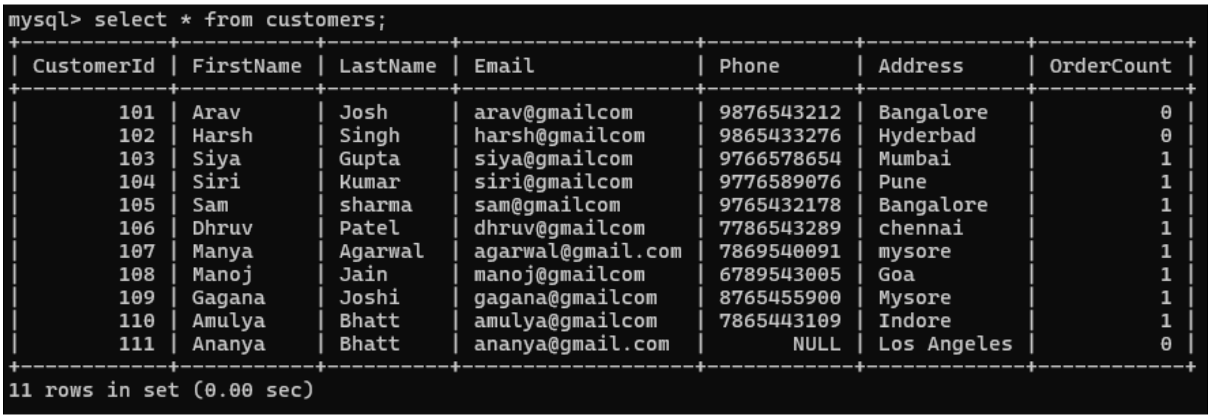
11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.
    -> ALTER TABLE Orders ADD status VARCHAR(20);
    -> UPDATE Orders SET status = 'pending' WHERE OrderID =209;
    -> UPDATE Orders SET status = 'shipped' WHERE OrderID =209;



```
mysql> select * from orders;
+---------+------------+------------+-------------+---------+
| OrderID | CustomerId | OrderDate  | TotalAmount | status  |
+---------+------------+------------+-------------+---------+
|     202 |        108 | 2008-11-03 |        7000 | NULL    |
|     204 |        103 | 1995-12-05 |       88000 | NULL    |
|     205 |        110 | 2000-07-08 |       67000 | NULL    |
|     206 |        109 | 2008-05-18 |       68090 | NULL    |
|     207 |        105 | 2018-09-28 |       98050 | NULL    |
|     208 |        104 | 2009-06-16 |        8790 | NULL    |
|     209 |        106 | 2015-04-18 |        7899 | shipped |
|     210 |        107 | 2019-08-10 |       77777 | NULL    |
|     211 |        109 | 2023-10-25 |       29000 | NULL    |
+---------+------------+------------+-------------+---------+
9 rows in set (0.00 sec)
```

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

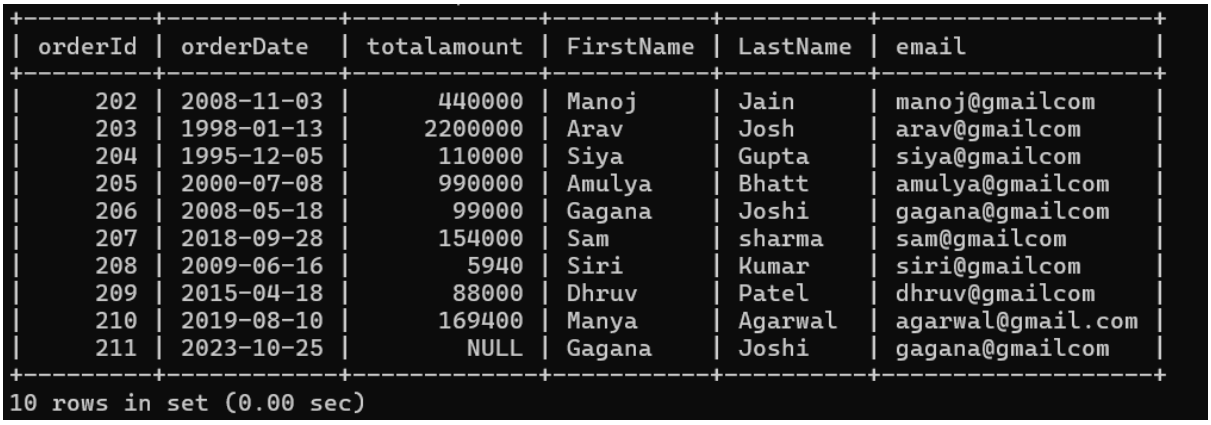
```
-> ALTER TABLE Customers
   -> ADD OrderCount INT DEFAULT 0;
->UPDATE Customers AS c
   -> SET OrderCount = (
   ->     SELECT COUNT(DISTINCT od.OrderId)
   ->     FROM OrderDetails AS od
   ->     JOIN Orders AS o ON od.OrderId = o.OrderId
   ->     WHERE o.CustomerId = c.CustomerId
   -> );
```



```
mysql> select * from customers;
+------------+-----------+----------+-------------------+------------+-------------+------------+
| CustomerId | FirstName | LastName | Email             | Phone      | Address     | OrderCount |
+------------+-----------+----------+-------------------+------------+-------------+------------+
|        101 | Arav      | Josh     | arav@gmailcom     | 9876543212 | Bangalore   |          0 |
|        102 | Harsh     | Singh    | harsh@gmailcom    | 9865433276 | Hyderbad    |          0 |
|        103 | Siya      | Gupta    | siya@gmailcom     | 9766578654 | Mumbai      |          1 |
|        104 | Siri      | Kumar    | siri@gmailcom     | 9776589076 | Pune        |          1 |
|        105 | Sam       | sharma   | sam@gmailcom      | 9765432178 | Bangalore   |          1 |
|        106 | Dhruv     | Patel    | dhruv@gmailcom    | 7786543289 | chennai     |          1 |
|        107 | Manya     | Agarwal  | agarwal@gmail.com | 7869540091 | mysore      |          1 |
|        108 | Manoj     | Jain     | manoj@gmailcom    | 6789543005 | Goa         |          1 |
|        109 | Gagana    | Joshi    | gagana@gmailcom   | 8765455900 | Mysore      |          1 |
|        110 | Amulya    | Bhatt    | amulya@gmailcom   | 7865443109 | Indore      |          1 |
|        111 | Ananya    | Bhatt    | ananya@gmail.com  |       NULL | Los Angeles |          0 |
+------------+-----------+----------+-------------------+------------+-------------+------------+
11 rows in set (0.00 sec)
```

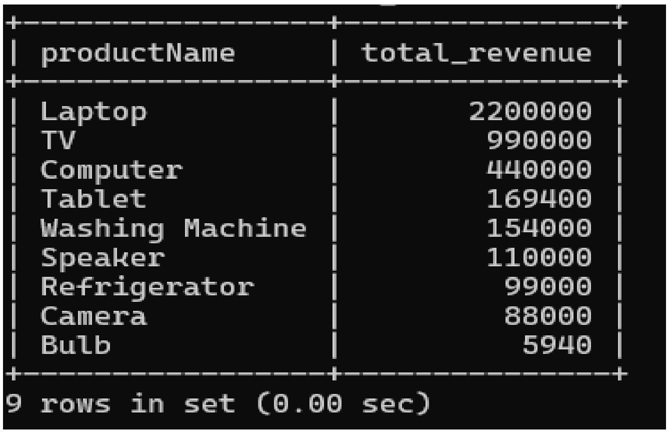
Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.
   -> select o.orderId ,o.orderDate , o.totalamount ,c.FirstName ,c.LastName,c.email from orders o inner join customers c on o.customerId=c.customerId;



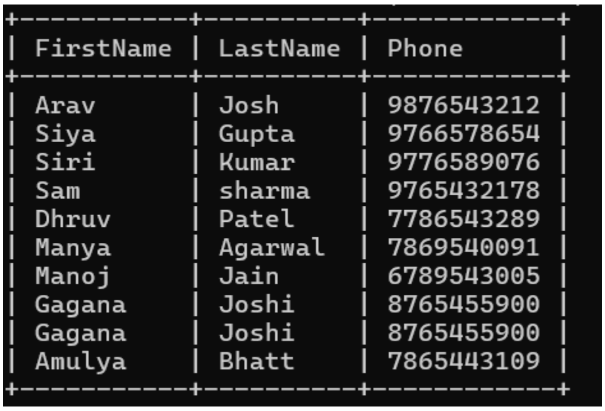
```
+---------+------------+-------------+-----------+----------+-------------------+
| orderId | orderDate  | totalamount | FirstName | LastName | email             |
+---------+------------+-------------+-----------+----------+-------------------+
|     202 | 2008-11-03 |      440000 | Manoj     | Jain     | manoj@gmailcom    |
|     203 | 1998-01-13 |     2200000 | Arav      | Josh     | arav@gmailcom     |
|     204 | 1995-12-05 |      110000 | Siya      | Gupta    | siya@gmailcom     |
|     205 | 2000-07-08 |      990000 | Amulya    | Bhatt    | amulya@gmailcom   |
|     206 | 2008-05-18 |       99000 | Gagana    | Joshi    | gagana@gmailcom   |
|     207 | 2018-09-28 |      154000 | Sam       | sharma   | sam@gmailcom      |
|     208 | 2009-06-16 |        5940 | Siri      | Kumar    | siri@gmailcom     |
|     209 | 2015-04-18 |       88000 | Dhruv     | Patel    | dhruv@gmailcom    |
|     210 | 2019-08-10 |      169400 | Manya     | Agarwal  | agarwal@gmail.com |
|     211 | 2023-10-25 |        NULL | Gagana    | Joshi    | gagana@gmailcom   |
+---------+------------+-------------+-----------+----------+-------------------+
10 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.
   -> SELECT p.productName, SUM(od.quantity * p.Price) AS total_revenue
   -> FROM products AS p
   -> INNER JOIN orderdetails AS od ON p.productId = od.productId
   -> GROUP BY p.productId, p.productName
   -> ORDER BY total_revenue DESC;



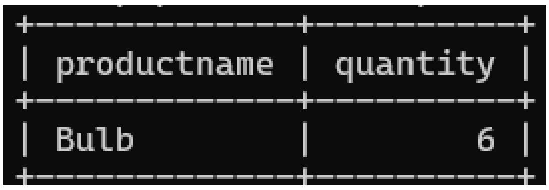
```
+------------------+---------------+
| productName      | total_revenue |
+------------------+---------------+
| Laptop           |       2200000 |
| TV               |        990000 |
| Computer         |        440000 |
| Tablet           |        169400 |
| Washing Machine  |        154000 |
| Speaker          |        110000 |
| Refrigerator     |         99000 |
| Camera           |         88000 |
| Bulb             |          5940 |
+------------------+---------------+
9 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.
   -> SELECT c.FirstName, c.LastName, c.Phone FROM Customers c INNER JOIN Orders o ON c.CustomerId = o.CustomerId;



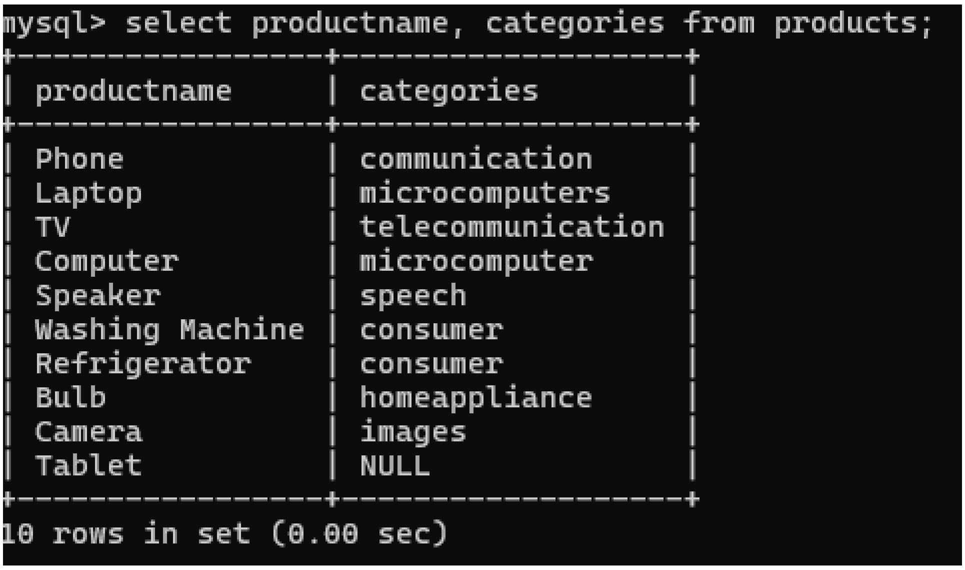
```
+-----------+----------+------------+
| FirstName | LastName | Phone      |
+-----------+----------+------------+
| Arav      | Josh     | 9876543212 |
| Siya      | Gupta    | 9766578654 |
| Siri      | Kumar    | 9776589076 |
| Sam       | sharma   | 9765432178 |
| Dhruv     | Patel    | 7786543289 |
| Manya     | Agarwal  | 7869540091 |
| Manoj     | Jain     | 6789543005 |
| Gagana    | Joshi    | 8765455900 |
| Gagana    | Joshi    | 8765455900 |
| Amulya    | Bhatt    | 7865443109 |
+-----------+----------+------------+
```

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.
   -> select p.productname, od.quantity from products as p inner join orderdetails as od where p.productid=od.productid order by od.quantity desc limit 1;



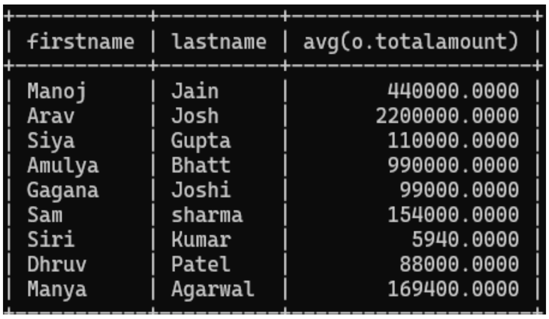
```
+-------------+----------+
| productname | quantity |
+-------------+----------+
| Bulb        |        6 |
+-------------+----------+
```

5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.
   -> alter table products add(categories varchar(20));
   update products set categories='communication' where productId=1;
   update products set categories='microcomputers' where productId=2;
   update products set categories='telecommunication' where productId=3;
   update products set categories='microcomputer' where productId=4;
   update products set categories='speech' where productId=5;
   update products set categories='consumer' where productId=6;
   update products set categories='consumer' where productId=7;
   update products set categories='homeappliance' where productId=8;
   update products set categories='images' where productId=9;
   -> select productname ,categories from products;



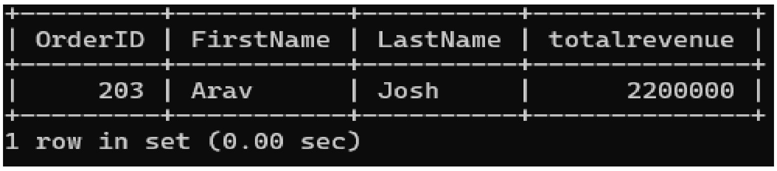
```
nysql> select productname, categories from products;
+-----------------+-------------------+
| productname     | categories        |
+-----------------+-------------------+
| Phone           | communication     |
| Laptop          | microcomputers    |
| TV              | telecommunication |
| Computer        | microcomputer     |
| Speaker         | speech            |
| Washing Machine | consumer          |
| Refrigerator    | consumer          |
| Bulb            | homeappliance     |
| Camera          | images            |
| Tablet          | NULL              |
+-----------------+-------------------+
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.
-> select c.firstname, c.lastname, avg(o.totalamount) from customers c inner join orders o on c.customerId=o.customerId group by c.firstname, c.lastname;



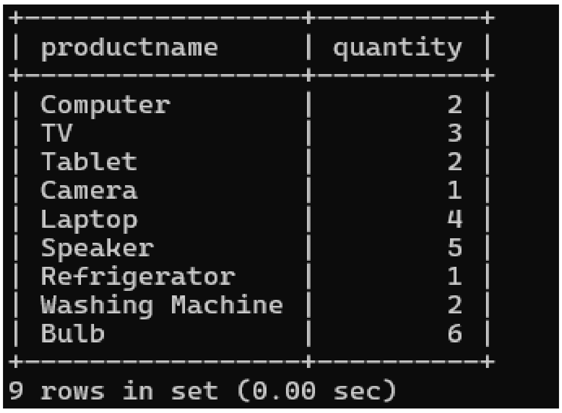
```
+-----------+----------+--------------------+
| firstname | lastname | avg(o.totalamount) |
+-----------+----------+--------------------+
| Manoj     | Jain     |        440000.0000 |
| Arav      | Josh     |       2200000.0000 |
| Siya      | Gupta    |        110000.0000 |
| Amulya    | Bhatt    |        990000.0000 |
| Gagana    | Joshi    |         99000.0000 |
| Sam       | sharma   |        154000.0000 |
| Siri      | Kumar    |          5940.0000 |
| Dhruv     | Patel    |         88000.0000 |
| Manya     | Agarwal  |        169400.0000 |
+-----------+----------+--------------------+
```

7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.
-> SELECT o.OrderID, c.FirstName, c.LastName, SUM(od.Quantity * p.Price) as totalrevenue FROM Orders AS o INNER JOIN Customers AS c ON o.CustomerId = c.CustomerId INNER JOIN OrderDetails AS od ON o.OrderID = od.OrderId INNER JOIN Products AS p ON od.ProductID = p.Pro
ductId GROUP BY o.OrderID, c.FirstName, c.LastName ORDER BY totalrevenue desc limit 1;



```
+---------+-----------+----------+--------------+
| OrderID | FirstName | LastName | totalrevenue |
+---------+-----------+----------+--------------+
|     203 | Arav      | Josh     |      2200000 |
+---------+-----------+----------+--------------+
1 row in set (0.00 sec)
```

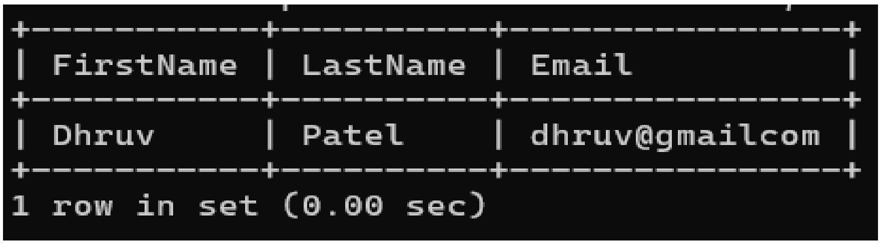
8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.
-> select p.productname ,od.quantity from products as p inner join orderdetails as od on p.productid=od.productid ;



```
+-----------------+----------+
| productname     | quantity |
+-----------------+----------+
| Computer        |        2 |
| TV              |        3 |
| Tablet          |        2 |
| Camera          |        1 |
| Laptop          |        4 |
| Speaker         |        5 |
| Refrigerator    |        1 |
| Washing Machine |        2 |
| Bulb            |        6 |
+-----------------+----------+
9 rows in set (0.00 sec)
```

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

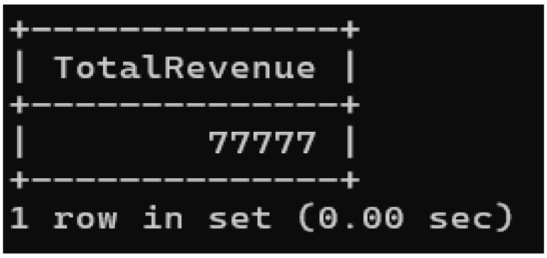
```
-> SELECT  c.FirstName, c.LastName, c.Email
   -> FROM Customers AS c JOIN Orders AS o ON c.CustomerId =
o.CustomerId
   -> JOIN OrderDetails AS od ON o.OrderID = od.OrderID
   -> JOIN Products AS p ON od.ProductID = p.ProductID
   -> WHERE p.ProductName = 'camera';
```



```
+-----------+----------+-----------------+
| FirstName | LastName | Email           |
+-----------+----------+-----------------+
| Dhruv     | Patel    | dhruv@gmailcom |
+-----------+----------+-----------------+
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 parameter.

```
-> SELECT SUM(o.TotalAmount) AS TotalRevenue
   ->  FROM Orders AS o
   ->  WHERE o.OrderDate BETWEEN '2019-08-10' AND '2022-08-29';
```



```
+--------------+
| TotalRevenue |
+--------------+
|        77777 |
+--------------+
1 row in set (0.00 sec)
```

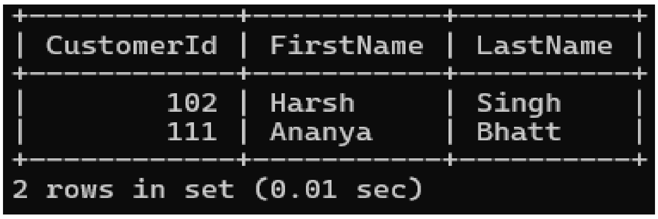
Task 4. Subquery and its type:

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

```
->SELECT c.CustomerId, c.FirstName, c.LastName
```

-> FROM Customers AS c

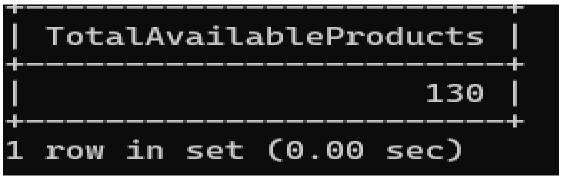
-> WHERE c.CustomerId NOT IN (SELECT o.CustomerId FROM Orders AS o);



```
+------------+-----------+----------+
| CustomerId | FirstName | LastName |
+------------+-----------+----------+
|        102 | Harsh     | Singh    |
|        111 | Ananya    | Bhatt    |
+------------+-----------+----------+
2 rows in set (0.01 sec)
```

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

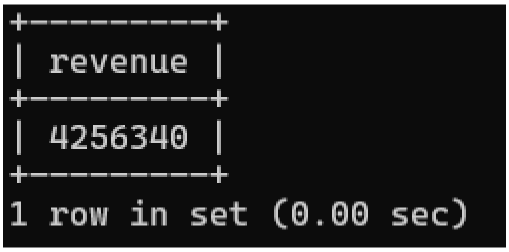
->select sum(QuantityInStock) from inventory;



```
+------------------------+
| TotalAvailableProducts |
+------------------------+
|                    130 |
+------------------------+
1 row in set (0.00 sec)
```

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

-> select sum(totalamount) as revenue from orders;



```
+---------+
| revenue |
+---------+
| 4256340 |
+---------+
1 row in set (0.00 sec)
```

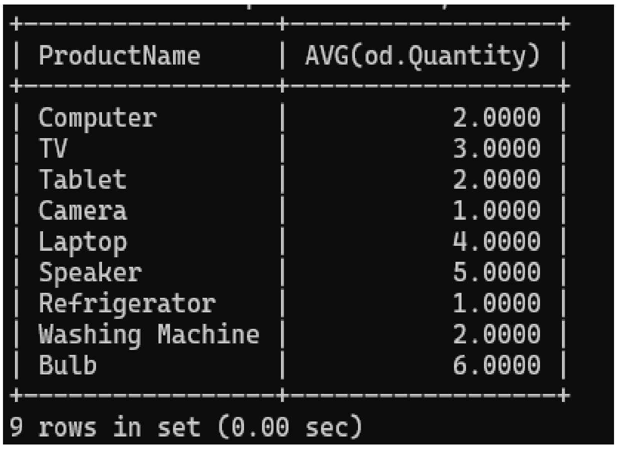
4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.

-> SELECT p.ProductName, AVG(od.Quantity)

-> FROM OrderDetails AS od

-> INNER JOIN Products AS p ON od.ProductID = p.ProductId

-> GROUP BY p.ProductName;



```
+-----------------+------------------+
| ProductName     | AVG(od.Quantity) |
+-----------------+------------------+
| Computer        |           2.0000 |
| TV              |           3.0000 |
| Tablet          |           2.0000 |
| Camera          |           1.0000 |
| Laptop          |           4.0000 |
| Speaker         |           5.0000 |
| Refrigerator    |           1.0000 |
| Washing Machine |           2.0000 |
| Bulb            |           6.0000 |
+-----------------+------------------+
9 rows in set (0.00 sec)
```

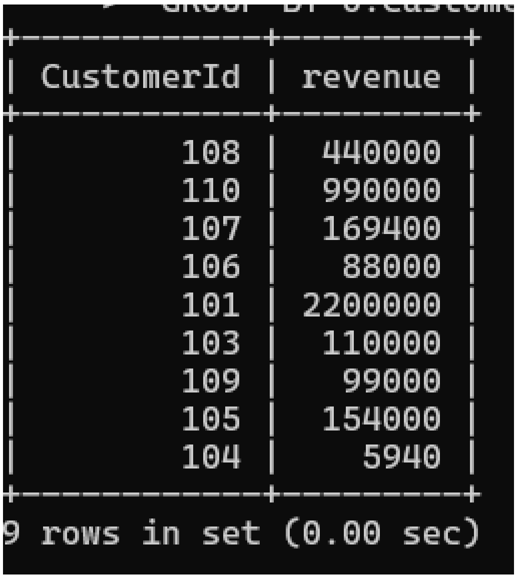
5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.

->SELECT o.CustomerId, SUM(od.Quantity * p.Price) as revenue

-> FROM Orders AS o INNER JOIN OrderDetails AS od ON o.OrderID = od.OrderId

-> INNER JOIN Products AS p ON od.ProductID = p.ProductId

-> GROUP BY o.CustomerId;



```
    ->  GROUP BY o.Custome
+------------+---------+
| CustomerId | revenue |
+------------+---------+
|        108 |  440000 |
|        110 |  990000 |
|        107 |  169400 |
|        106 |   88000 |
|        101 | 2200000 |
|        103 |  110000 |
|        109 |   99000 |
|        105 |  154000 |
|        104 |    5940 |
+------------+---------+
9 rows in set (0.00 sec)
```

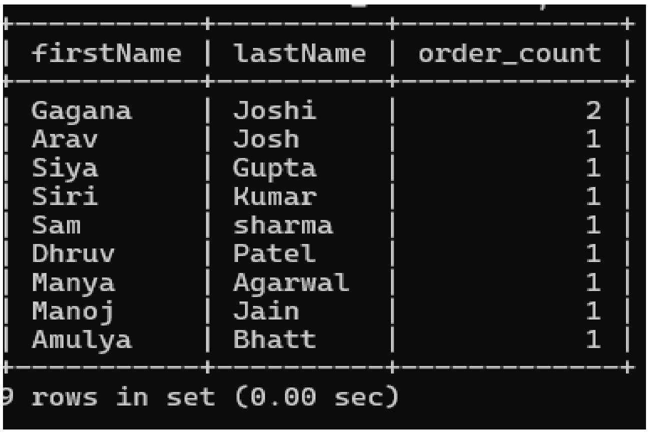
6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.

-> SELECT c.firstName,c.lastName,COUNT(o.orderId) AS order_count

-> FROM customers AS c INNER JOIN orders AS o ON c.customerId = o.customerId

-> GROUP BY c.customerId, c.firstName, c.lastName

-> ORDER BY order_count DESC;



```
+-----------+----------+-------------+
| firstName | lastName | order_count |
+-----------+----------+-------------+
| Gagana    | Joshi    |           2 |
| Arav      | Josh     |           1 |
| Siya      | Gupta    |           1 |
| Siri      | Kumar    |           1 |
| Sam       | sharma   |           1 |
| Dhruv     | Patel    |           1 |
| Manya     | Agarwal  |           1 |
| Manoj     | Jain     |           1 |
| Amulya    | Bhatt    |           1 |
+-----------+----------+-------------+
9 rows 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.

->SELECT c.firstName,c.lastName,p.productName,SUM(od.quantity) AS total_quantity_ordered

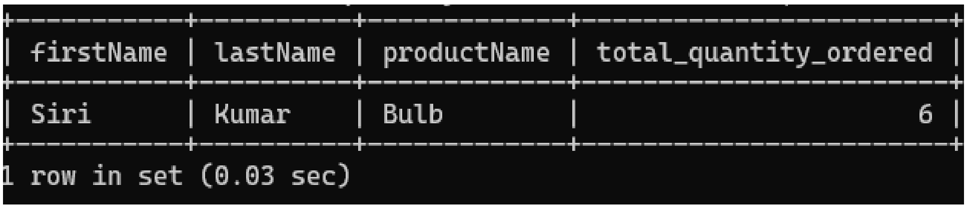
-> FROM customers AS c INNER JOIN orders AS o ON c.customerId = o.customerId

-> INNER JOIN orderdetails AS od ON o.orderId = od.orderId

-> INNER JOIN products AS p ON od.productId = p.productId

-> GROUP BY c.customerId, c.firstName, c.lastName, p.productName

-> ORDER BY total_quantity_ordered DESC limit 1;



```
+-----------+----------+-------------+------------------------+
| firstName | lastName | productName | total_quantity_ordered |
+-----------+----------+-------------+------------------------+
| Siri      | Kumar    | Bulb        |                      6 |
+-----------+----------+-------------+------------------------+
1 row in set (0.03 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.

->SELECT c.firstName, c.lastName, SUM(od.quantity * p.price) AS total_spending

-> FROM customers AS c

-> INNER JOIN orders AS o ON c.customerId = o.customerId

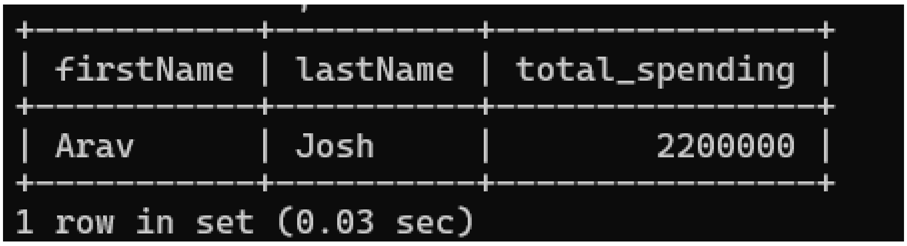
-> INNER JOIN orderdetails AS od ON o.orderId = od.orderId

-> INNER JOIN products AS p ON od.productId = p.productId

-> GROUP BY c.customerId

-> ORDER BY total_spending DESC

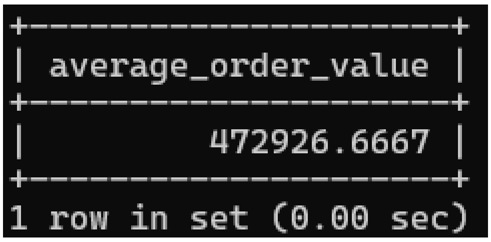
-> LIMIT 1;



```
+-----------+----------+----------------+
| firstName | lastName | total_spending |
+-----------+----------+----------------+
| Arav      | Josh     |        2200000 |
+-----------+----------+----------------+
1 row in set (0.03 sec)
```

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

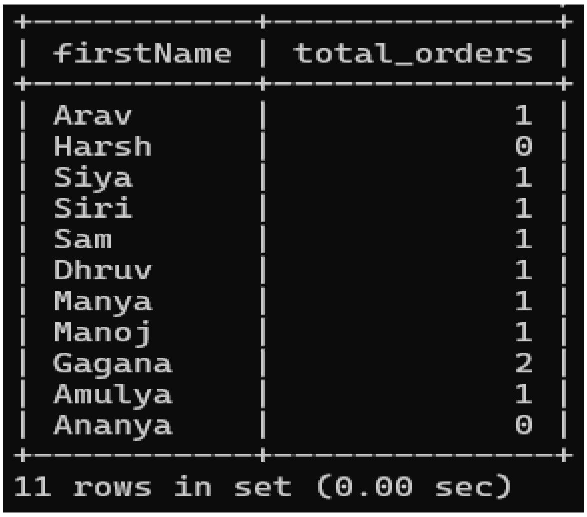
```
->SELECT AVG(order_value) AS average_order_value FROM (
   ->  SELECT  o.customerId,SUM(od.quantity * p.price) AS order_value
   ->  FROM  orders AS o INNER JOIN orderdetails AS od ON o.orderId = od.orderId
   -> INNER JOIN products AS p ON od.productId = p.productId
   -> GROUP BY o.orderId)as order_summary;
```



```
+---------------------+
| average_order_value |
+---------------------+
|         472926.6667 |
+---------------------+
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.

```
->SELECT c.firstName, COUNT(o.orderId) AS total_orders
   -> FROM customers AS c
   -> LEFT JOIN orders AS o ON c.customerId = o.customerId
   -> GROUP BY c.firstName;
```



```
+-----------+--------------+
| firstName | total_orders |
+-----------+--------------+
| Arav      |            1 |
| Harsh     |            0 |
| Siya      |            1 |
| Siri      |            1 |
| Sam       |            1 |
| Dhruv     |            1 |
| Manya     |            1 |
| Manoj     |            1 |
| Gagana    |            2 |
| Amulya    |            1 |
| Ananya    |            0 |
+-----------+--------------+
11 rows in set (0.00 sec)
```