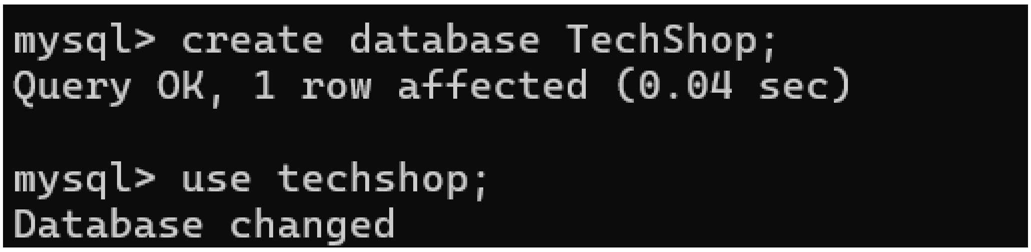
HEXAWARE ASSIGNMENT - 1

NAME : AISHWARYA B
SUPERSET ID: 5006869
ASSIGNMENT NAME: 1- ELECTRONIC GADGET

Task:1 Database Design:

1. Create the database named "TechShop"

Query:
-> create database TechShop;
-> use database TechShop;

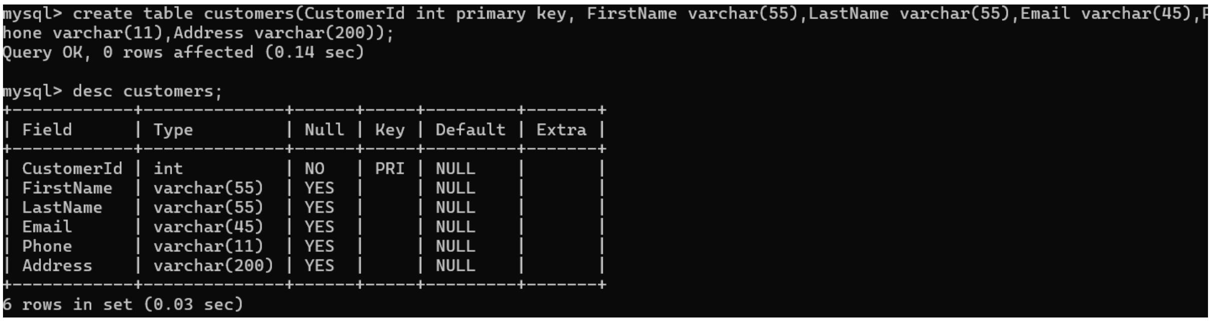


```
mysql> create database TechShop;
Query OK, 1 row affected (0.04 sec)

mysql> use techshop;
Database changed
```

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

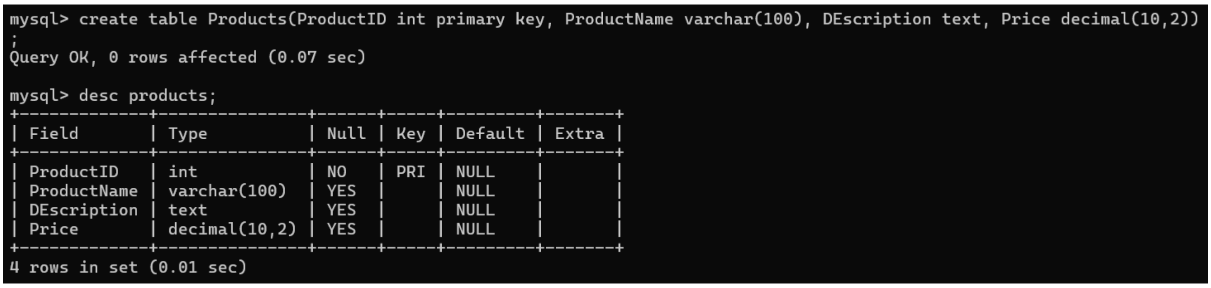
Query:

1. Customer:
   -> create table customers(CustomerId int primary key, FirstName varchar(55),LastName varchar(55),Email varchar(45),Phone varchar(11),Address varchar(200));



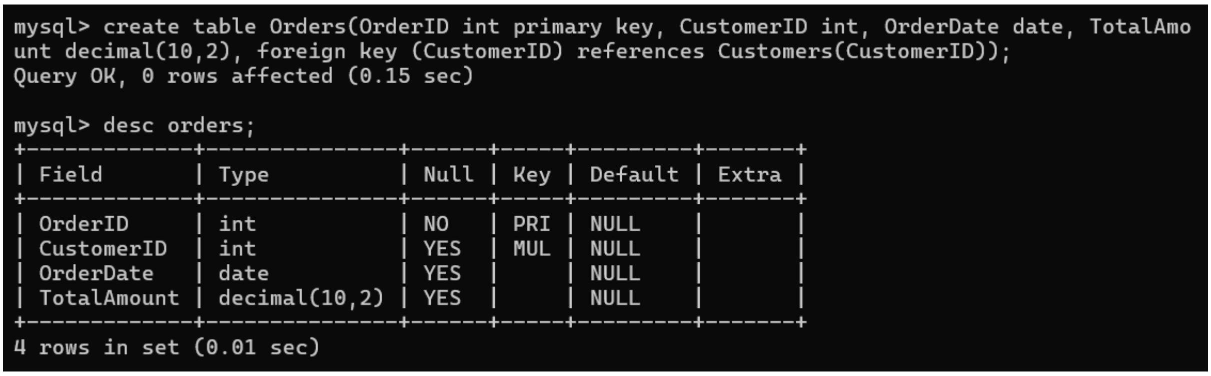
```
mysql> create table customers(CustomerId int primary key, FirstName varchar(55),LastName varchar(55),Email varchar(45),P
hone varchar(11),Address varchar(200));
Query OK, 0 rows affected (0.14 sec)

mysql> desc customers;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| CustomerId | int          | NO   | PRI | NULL    |       |
| FirstName  | varchar(55)  | YES  |     | NULL    |       |
| LastName   | varchar(55)  | YES  |     | NULL    |       |
| Email      | varchar(45)  | YES  |     | NULL    |       |
| Phone      | varchar(11)  | YES  |     | NULL    |       |
| Address    | varchar(200) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

2. Products:
   -> create table Products(ProductID int primary key, ProductName varchar(100), Description text, Price decimal(10,2));



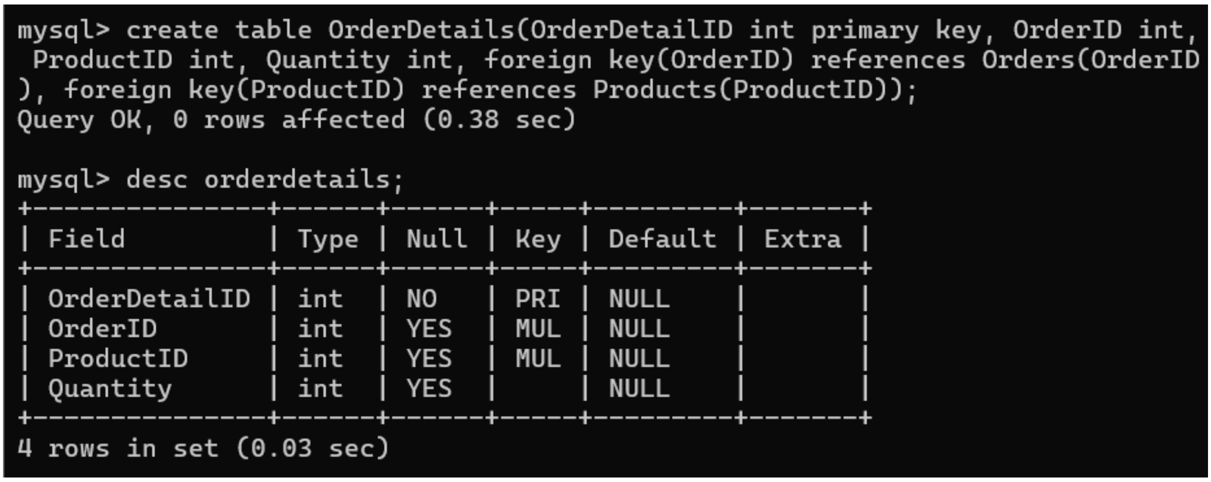
```
mysql> create table Products(ProductID int primary key, ProductName varchar(100), DEscription text, Price decimal(10,2))
;
Query OK, 0 rows affected (0.07 sec)

mysql> desc products;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| ProductID   | int           | NO   | PRI | NULL    |       |
| ProductName | varchar(100)  | YES  |     | NULL    |       |
| DEscription | text          | YES  |     | NULL    |       |
| Price       | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

3. Orders:
   -> create table Orders(OrderID int primary key, CustomerID int, OrderDate date, TotalAmount decimal(10,2), foreign key (CustomerID) references Customers(CustomerID));



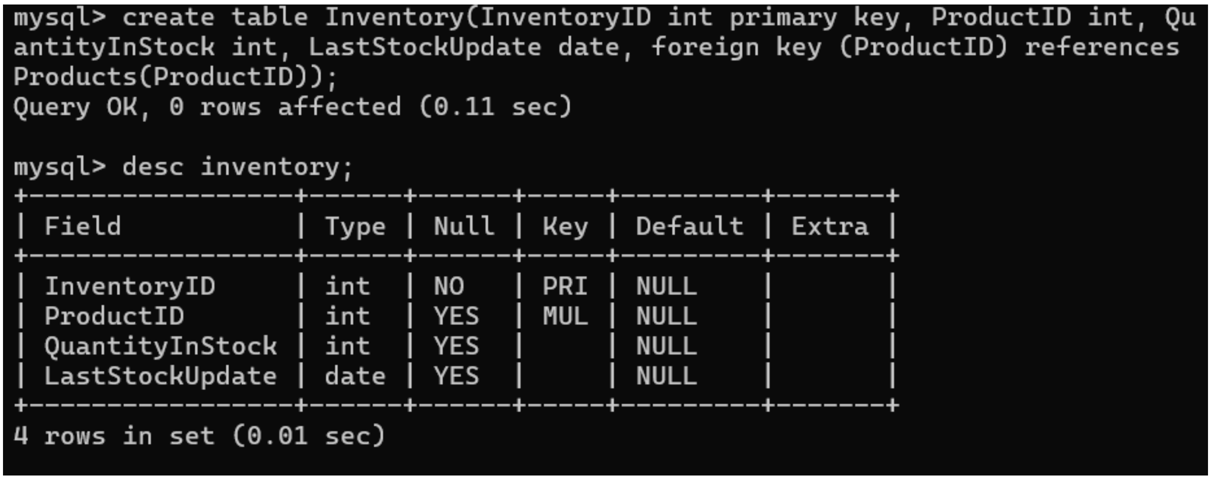
```
mysql> create table Orders(OrderID int primary key, CustomerID int, OrderDate date, TotalAmo
unt decimal(10,2), foreign key (CustomerID) references Customers(CustomerID));
Query OK, 0 rows affected (0.15 sec)

mysql> desc orders;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| OrderID     | int           | NO   | PRI | NULL    |       |
| CustomerID  | int           | YES  | MUL | NULL    |       |
| OrderDate   | date          | YES  |     | NULL    |       |
| TotalAmount | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

4. OrderDetails:
   -> create table OrderDetails(OrderDetailID int primary key, OrderID int, ProductID int, Quantity int, foreign key(OrderID) references Orders(OrderID), foreign key(ProductID) references Products(ProductID));



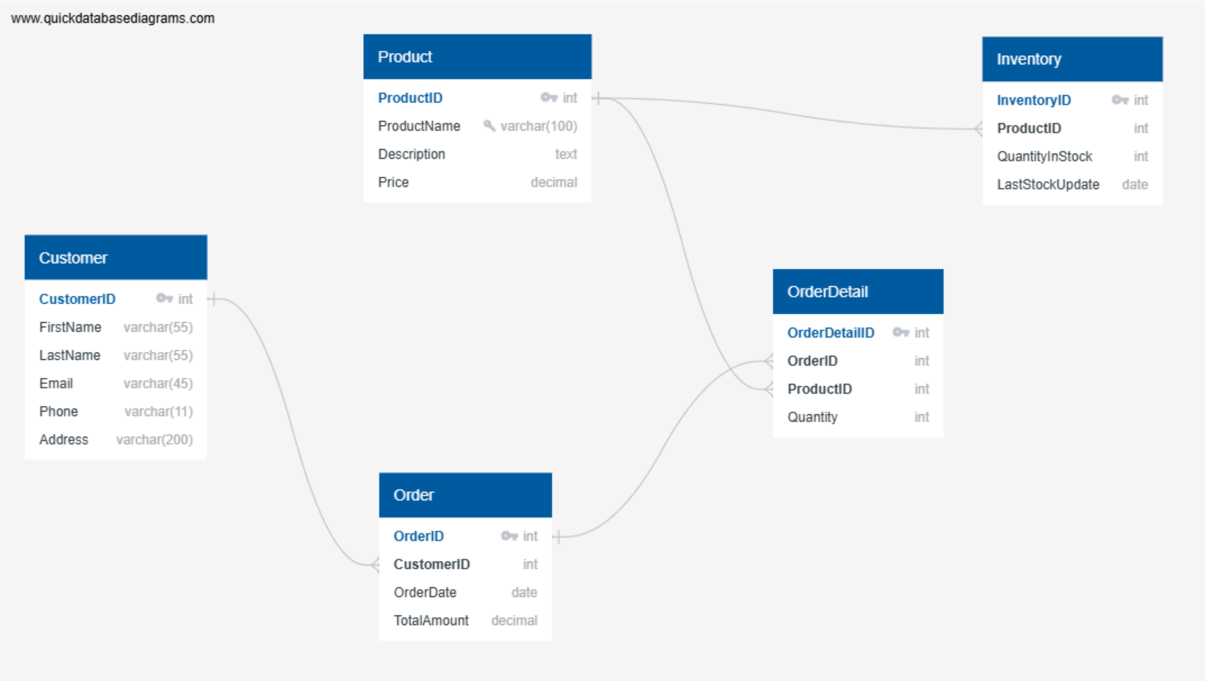
```
mysql> create table OrderDetails(OrderDetailID int primary key, OrderID int,
 ProductID int, Quantity int, foreign key(OrderID) references Orders(OrderID
), foreign key(ProductID) references Products(ProductID));
Query OK, 0 rows affected (0.38 sec)

mysql> desc orderdetails;
+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| OrderDetailID | int  | NO   | PRI | NULL    |       |
| OrderID       | int  | YES  | MUL | NULL    |       |
| ProductID     | int  | YES  | MUL | NULL    |       |
| Quantity      | int  | YES  |     | NULL    |       |
+---------------+------+------+-----+---------+-------+
4 rows in set (0.03 sec)
```

5. Inventory:
   -> create table Inventory(InventoryID int primary key, ProductID int, QuantityInStock int, LastStockUpdate date, foreign key (ProductID) references Products(ProductID));



```
mysql> create table Inventory(InventoryID int primary key, ProductID int, Qu
antityInStock int, LastStockUpdate date, foreign key (ProductID) references
Products(ProductID));
Query OK, 0 rows affected (0.11 sec)

mysql> desc inventory;
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| InventoryID     | int  | NO   | PRI | NULL    |       |
| ProductID       | int  | YES  | MUL | NULL    |       |
| QuantityInStock | int  | YES  |     | NULL    |       |
| LastStockUpdate | date | YES  |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

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



Product	
ProductID	int
ProductName	varchar(100)
Description	text
Price	decimal

Inventory	
InventoryID	int
ProductID	int
QuantityInStock	int
LastStockUpdate	date

Customer	
CustomerID	int
FirstName	varchar(55)
LastName	varchar(55)
Email	varchar(45)
Phone	varchar(11)
Address	varchar(200)

OrderDetail	
OrderDetailID	int
OrderID	int
ProductID	int
Quantity	int

Order	
OrderID	int
CustomerID	int
OrderDate	date
TotalAmount	decimal

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

In this project, the **Primary Key (PK)** and **Foreign Key (FK)** constraints have been implemented to ensure referential integrity between tables.

Primary Keys:

- Customers: CustomerID is the Primary Key
- Products: ProductID is the Primary Key
- Orders: OrderID is the Primary Key
- OrderDetails: OrderDetailID is the Primary Key
- Inventory: InventoryID is the Primary Key

Foreign Keys:

Relationships between tables are enforced through Foreign Keys:

- Orders.CustomerID → references Customers.CustomerID
- OrderDetails.OrderID → references Orders.OrderID
- OrderDetails.ProductID → references Products.ProductID
- Inventory.ProductID → references Products.ProductID

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

a. Customers
b. Products
c. Orders
d. OrderDetails
e. Inventory

Query:

a. Customer:
   -> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0001,"Neha","Ashok","nehaashok@gmail.com","9856423715","No.21, Richard Street, Chennai");

   -> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0002,"Arjun","Nagesh","arjun2004@gmail.com","8435127648","No.78, Rainbow Nagar, Chennai");

   -> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0003,"Sophie","Victor","sophievic@gmail.com","6473519421","No.31/A, Ganga Roads,Parrys, Chennai");

   -> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0004,"Syed","Ahmed","syedahmed@gmail.com","8429516443","FF1,Block2, Lake ViewApartments,Adayar, Chennai");

   -> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0005,"Kiran","Ghosh","kiranghosh@gmail.com","9600044476","C-34,Block22, Shanti Park,Egmore, Chennai");

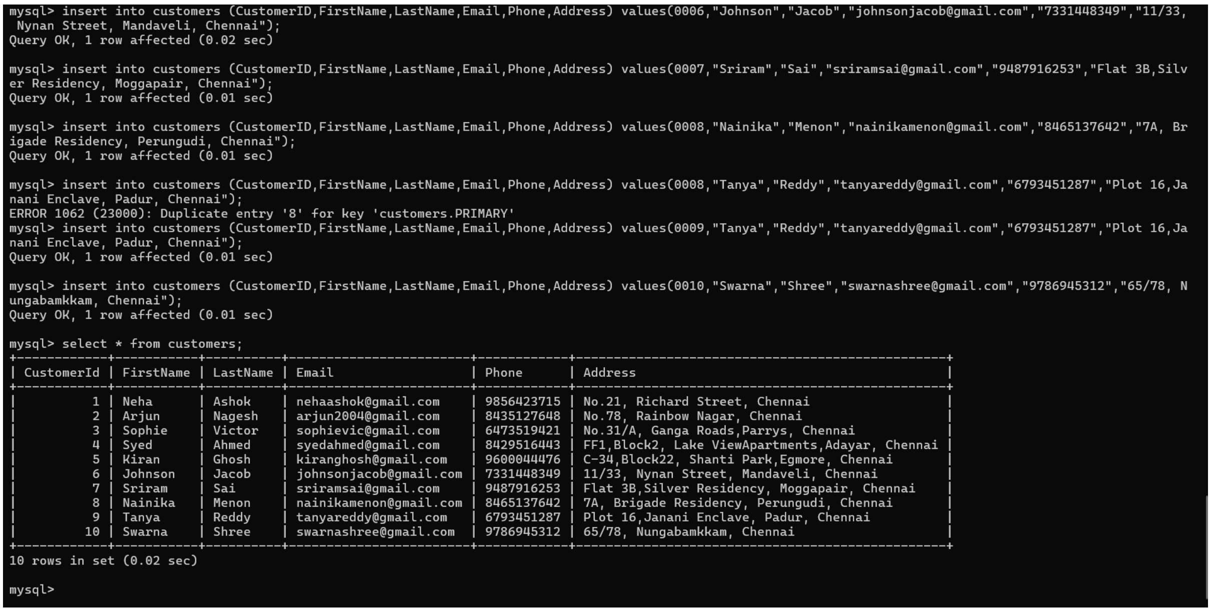
   -> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0006,"Johnson","Jacob","johnsonjacob@gmail.com","7331448349","11/33, Nynan Street, Mandaveli, Chennai");

   -> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0007,"Sriram","Sai","sriramsai@gmail.com","9487916253","Flat 3B,Silver Residency, Moggapair, Chennai");

   -> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0008,"Nainika","Menon","nainikamenon@gmail.com","8465137642","7A, Brigade Residency, Perungudi, Chennai");

-> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0009,"Tanya","Reddy","tanyareddy@gmail.com","6793451287","Plot 16,Janani Enclave, Padur, Chennai");

-> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0010,"Swarna","Shree","swarnashree@gmail.com","9786945312","65/78, Nungabamkkam, Chennai");



```
mysql> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0006,"Johnson","Jacob","johnsonjacob@gmail.com","7331448349","11/33,
 Nynan Street, Mandaveli, Chennai");
Query OK, 1 row affected (0.02 sec)

mysql> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0007,"Sriram","Sai","sriramsai@gmail.com","9487916253","Flat 3B,Silv
er Residency, Moggapair, Chennai");
Query OK, 1 row affected (0.01 sec)

mysql> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0008,"Nainika","Menon","nainikamenon@gmail.com","8465137642","7A, Br
igade Residency, Perungudi, Chennai");
Query OK, 1 row affected (0.01 sec)

mysql> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0008,"Tanya","Reddy","tanyareddy@gmail.com","6793451287","Plot 16,Ja
nani Enclave, Padur, Chennai");
ERROR 1062 (23000): Duplicate entry '8' for key 'customers.PRIMARY'
mysql> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0009,"Tanya","Reddy","tanyareddy@gmail.com","6793451287","Plot 16,Ja
nani Enclave, Padur, Chennai");
Query OK, 1 row affected (0.01 sec)

mysql> insert into customers (CustomerID,FirstName,LastName,Email,Phone,Address) values(0010,"Swarna","Shree","swarnashree@gmail.com","9786945312","65/78, N
ungabamkkam, Chennai");
Query OK, 1 row affected (0.01 sec)

mysql> select * from customers;
+------------+-----------+----------+------------------------+------------+-------------------------------------------------+
| CustomerId | FirstName | LastName | Email                  | Phone      | Address                                         |
+------------+-----------+----------+------------------------+------------+-------------------------------------------------+
|          1 | Neha      | Ashok    | nehaashok@gmail.com    | 9856423715 | No.21, Richard Street, Chennai                  |
|          2 | Arjun     | Nagesh   | arjun2004@gmail.com    | 8435127648 | No.78, Rainbow Nagar, Chennai                   |
|          3 | Sophie    | Victor   | sophievic@gmail.com    | 6473519421 | No.31/A, Ganga Roads,Parrys, Chennai            |
|          4 | Syed      | Ahmed    | syedahmed@gmail.com    | 8429516443 | FF1,Block2, Lake ViewApartments,Adayar, Chennai |
|          5 | Kiran     | Ghosh    | kiranghosh@gmail.com   | 9600044476 | C-34,Block22, Shanti Park,Egmore, Chennai       |
|          6 | Johnson   | Jacob    | johnsonjacob@gmail.com | 7331448349 | 11/33, Nynan Street, Mandaveli, Chennai         |
|          7 | Sriram    | Sai      | sriramsai@gmail.com    | 9487916253 | Flat 3B,Silver Residency, Moggapair, Chennai    |
|          8 | Nainika   | Menon    | nainikamenon@gmail.com | 8465137642 | 7A, Brigade Residency, Perungudi, Chennai       |
|          9 | Tanya     | Reddy    | tanyareddy@gmail.com   | 6793451287 | Plot 16,Janani Enclave, Padur, Chennai          |
|         10 | Swarna    | Shree    | swarnashree@gmail.com  | 9786945312 | 65/78, Nungabamkkam, Chennai                    |
+------------+-----------+----------+------------------------+------------+-------------------------------------------------+
10 rows in set (0.02 sec)

mysql>
```

b. Products:

-> insert into products (productid, productname, description, price) values (1001, "redmi note 13", "6.6-inch AMOLED display, 128GB storage, 5G support", 15999.00);

-> insert into products (productid, productname, description, price) values (1002, "boat airdopes 161", "bluetooth 5.2, up to 40 hrs playback, fast charging", 1299.00);

-> insert into products (productid, productname, description, price) values (1003, "dell inspiron 15", "15.6-inch FHD, i5 12th Gen, 512GB SSD, 8GB RAM", 51999.00);

-> insert into products (productid, productname, description, price) values (1004, "logitech m331 silent mouse", "2.4GHz wireless, ergonomic, 18-month battery", 899.00);

-> insert into products (productid, productname, description, price) values (1005, "lg ultragear 27-inch", "QHD gaming monitor, 144Hz, IPS panel", 23999.00);

-> insert into products (productid, productname, description, price) values (1006, "jbl flip 6", "portable bluetooth speaker, IP67 waterproof, 12 hrs battery", 8999.00);

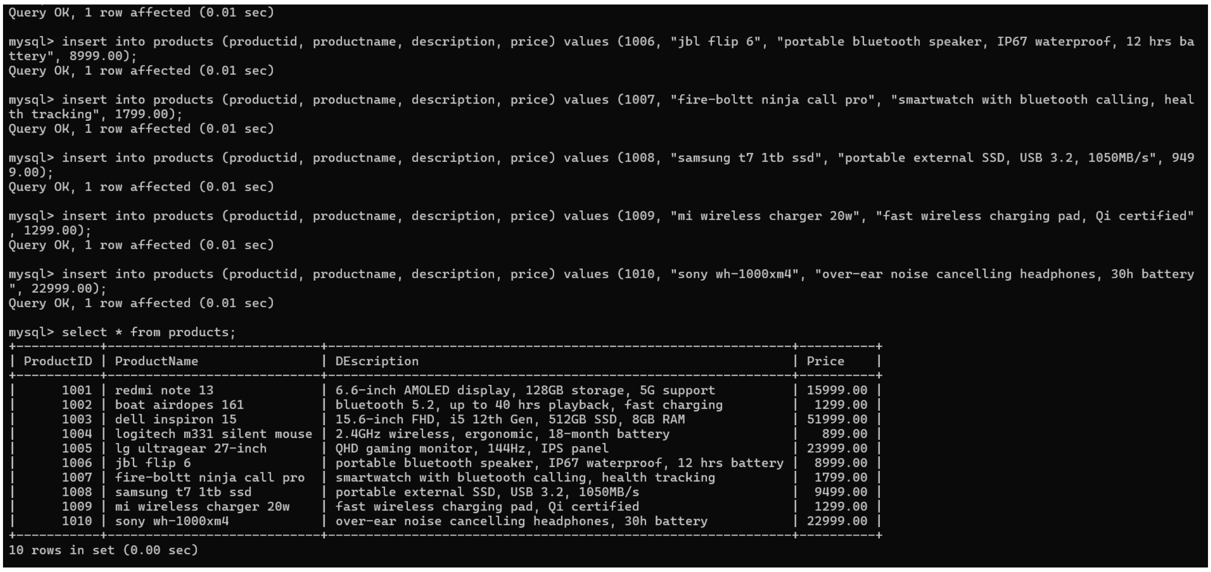
-> insert into products (productid, productname, description, price) values (1007, "fire-boltt ninja call pro", "smartwatch with bluetooth calling, health tracking", 1799.00);

-> insert into products (productid, productname, description, price) values (1008, "samsung t7 1tb ssd", "portable external SSD, USB 3.2, 1050MB/s", 9499.00);

-> insert into products (productid, productname, description, price) values (1009, "mi wireless charger 20w", "fast wireless charging pad, Qi certified", 1299.00);

-> insert into products (productid, productname, description, price) values (1010, "sony wh-1000xm4", "over-ear noise cancelling headphones, 30h battery", 22999.00);

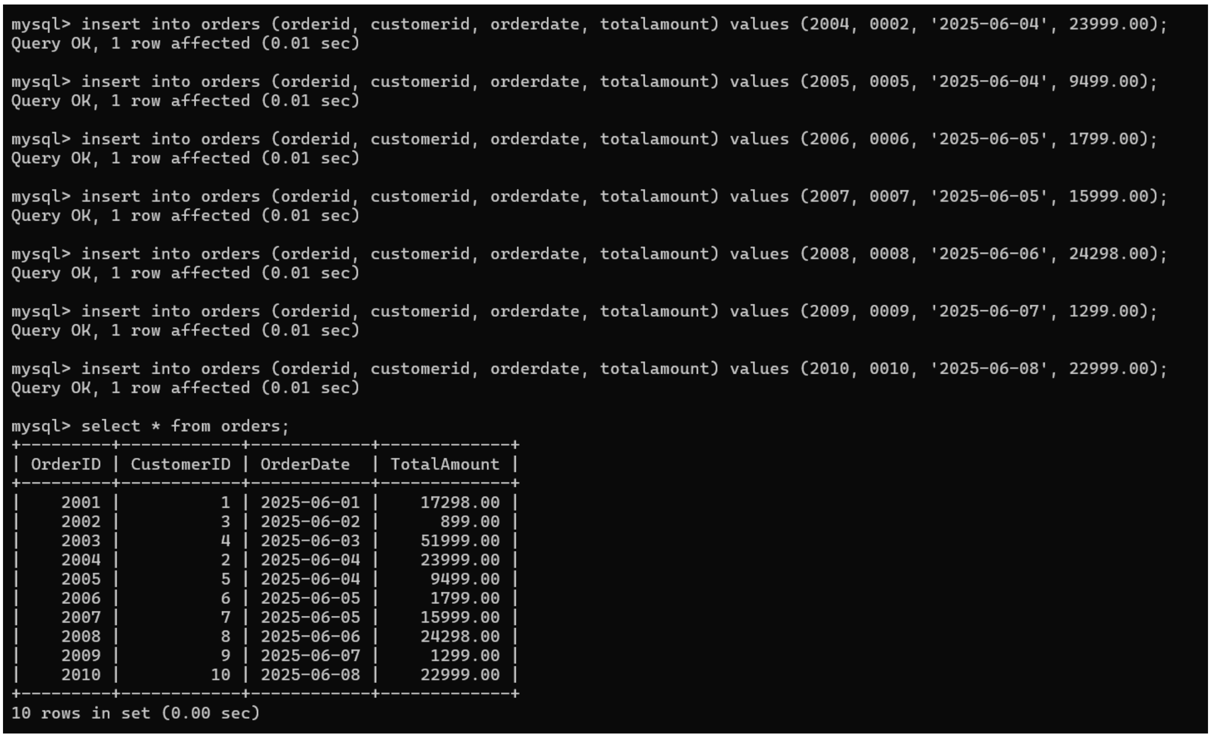
-> select * from products;



```
Query OK, 1 row affected (0.01 sec)

mysql> insert into products (productid, productname, description, price) values (1006, "jbl flip 6", "portable bluetooth speaker, IP67 waterproof, 12 hrs ba
ttery", 8999.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into products (productid, productname, description, price) values (1007, "fire-boltt ninja call pro", "smartwatch with bluetooth calling, heal
th tracking", 1799.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into products (productid, productname, description, price) values (1008, "samsung t7 1tb ssd", "portable external SSD, USB 3.2, 1050MB/s", 949
9.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into products (productid, productname, description, price) values (1009, "mi wireless charger 20w", "fast wireless charging pad, Qi certified"
, 1299.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into products (productid, productname, description, price) values (1010, "sony wh-1000xm4", "over-ear noise cancelling headphones, 30h battery
", 22999.00);
Query OK, 1 row affected (0.01 sec)

mysql> select * from products;
+-----------+----------------------------+-------------------------------------------------------------+----------+
| ProductID | ProductName                | DEscription                                                 | Price    |
+-----------+----------------------------+-------------------------------------------------------------+----------+
|      1001 | redmi note 13              | 6.6-inch AMOLED display, 128GB storage, 5G support          | 15999.00 |
|      1002 | boat airdopes 161          | bluetooth 5.2, up to 40 hrs playback, fast charging         |  1299.00 |
|      1003 | dell inspiron 15           | 15.6-inch FHD, i5 12th Gen, 512GB SSD, 8GB RAM              | 51999.00 |
|      1004 | logitech m331 silent mouse | 2.4GHz wireless, ergonomic, 18-month battery                |   899.00 |
|      1005 | lg ultragear 27-inch       | QHD gaming monitor, 144Hz, IPS panel                        | 23999.00 |
|      1006 | jbl flip 6                 | portable bluetooth speaker, IP67 waterproof, 12 hrs battery |  8999.00 |
|      1007 | fire-boltt ninja call pro  | smartwatch with bluetooth calling, health tracking          |  1799.00 |
|      1008 | samsung t7 1tb ssd         | portable external SSD, USB 3.2, 1050MB/s                    |  9499.00 |
|      1009 | mi wireless charger 20w    | fast wireless charging pad, Qi certified                    |  1299.00 |
|      1010 | sony wh-1000xm4            | over-ear noise cancelling headphones, 30h battery           | 22999.00 |
+-----------+----------------------------+-------------------------------------------------------------+----------+
10 rows in set (0.00 sec)
```

c. Orders:

```
->  insert into orders (orderid, customerid, orderdate, totalamount) values (2001, 0001, '2025-06-01', 17298.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2002, 0003, '2025-06-02', 899.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2003, 0004, '2025-06-03', 51999.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2004, 0002, '2025-06-04', 23999.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2005, 0005, '2025-06-04', 9499.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2006, 0006, '2025-06-05', 1799.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2007, 0007, '2025-06-05', 15999.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2008, 0008, '2025-06-06', 24298.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2009, 0009, '2025-06-07', 1299.00);

-> insert into orders (orderid, customerid, orderdate, totalamount) values (2010, 0010, '2025-06-08', 22999.00);
```



```
mysql> insert into orders (orderid, customerid, orderdate, totalamount) values (2004, 0002, '2025-06-04', 23999.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders (orderid, customerid, orderdate, totalamount) values (2005, 0005, '2025-06-04', 9499.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders (orderid, customerid, orderdate, totalamount) values (2006, 0006, '2025-06-05', 1799.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders (orderid, customerid, orderdate, totalamount) values (2007, 0007, '2025-06-05', 15999.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders (orderid, customerid, orderdate, totalamount) values (2008, 0008, '2025-06-06', 24298.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders (orderid, customerid, orderdate, totalamount) values (2009, 0009, '2025-06-07', 1299.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders (orderid, customerid, orderdate, totalamount) values (2010, 0010, '2025-06-08', 22999.00);
Query OK, 1 row affected (0.01 sec)

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|    2001 |          1 | 2025-06-01 |    17298.00 |
|    2002 |          3 | 2025-06-02 |      899.00 |
|    2003 |          4 | 2025-06-03 |    51999.00 |
|    2004 |          2 | 2025-06-04 |    23999.00 |
|    2005 |          5 | 2025-06-04 |     9499.00 |
|    2006 |          6 | 2025-06-05 |     1799.00 |
|    2007 |          7 | 2025-06-05 |    15999.00 |
|    2008 |          8 | 2025-06-06 |    24298.00 |
|    2009 |          9 | 2025-06-07 |     1299.00 |
|    2010 |         10 | 2025-06-08 |    22999.00 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

d. OrderDetails:

-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3001, 2001, 1001, 1);

-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3002, 2001, 1002, 1);

-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3003, 2002, 1004, 1);

-l> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3004, 2003, 1003, 1);

-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3005, 2004, 1005, 3);

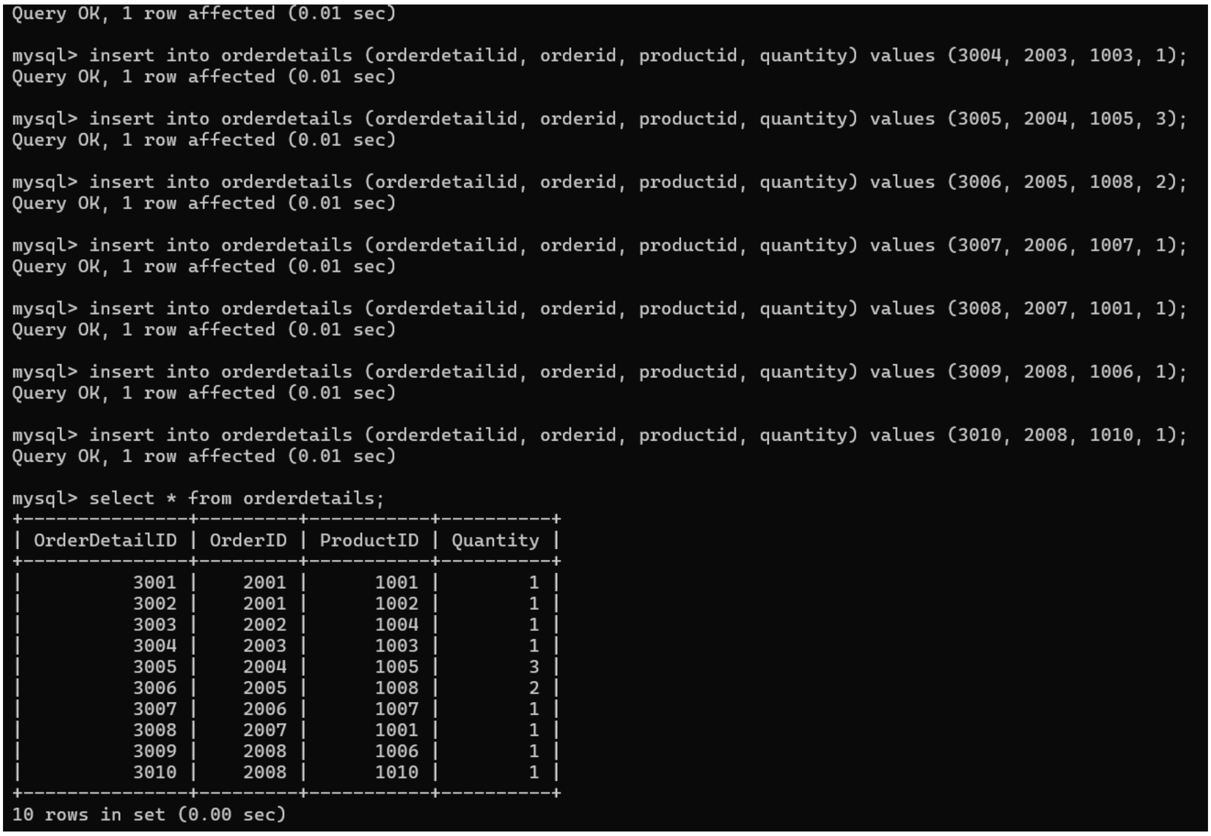
-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3006, 2005, 1008, 2);

-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3007, 2006, 1007, 1);

-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3008, 2007, 1001, 1);

-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3009, 2008, 1006, 1);

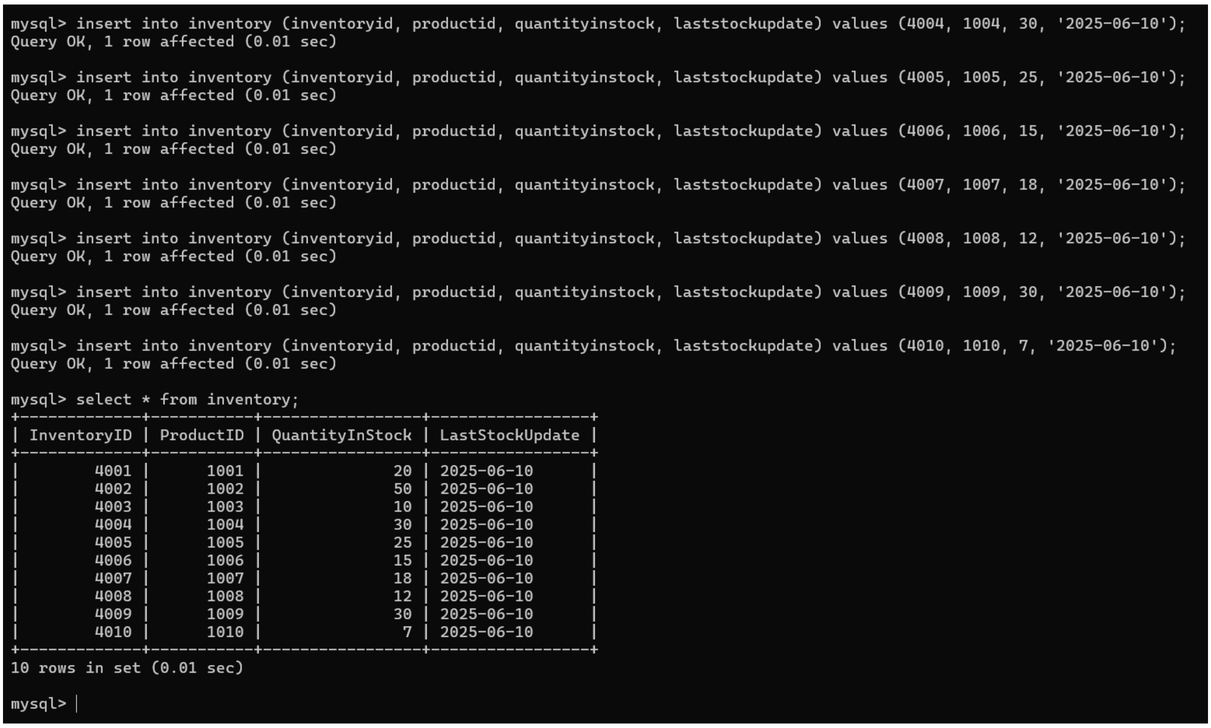
-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3010, 2008, 1010, 1);



```
Query OK, 1 row affected (0.01 sec)

mysql> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3004, 2003, 1003, 1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3005, 2004, 1005, 3);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3006, 2005, 1008, 2);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3007, 2006, 1007, 1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3008, 2007, 1001, 1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3009, 2008, 1006, 1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3010, 2008, 1010, 1);
Query OK, 1 row affected (0.01 sec)

mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|          3001 |    2001 |      1001 |        1 |
|          3002 |    2001 |      1002 |        1 |
|          3003 |    2002 |      1004 |        1 |
|          3004 |    2003 |      1003 |        1 |
|          3005 |    2004 |      1005 |        3 |
|          3006 |    2005 |      1008 |        2 |
|          3007 |    2006 |      1007 |        1 |
|          3008 |    2007 |      1001 |        1 |
|          3009 |    2008 |      1006 |        1 |
|          3010 |    2008 |      1010 |        1 |
+---------------+---------+-----------+----------+
10 rows in set (0.00 sec)
```

e. Inventory:

```
->  insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4001, 1001, 20, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4002, 1002, 50, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4003, 1003, 10, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4004, 1004, 30, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4005, 1005, 25, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4006, 1006, 15, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4007, 1007, 18, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4008, 1008, 12, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4009, 1009, 30, '2025-06-10');

-> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate)
values (4010, 1010, 7, '2025-06-10');
```



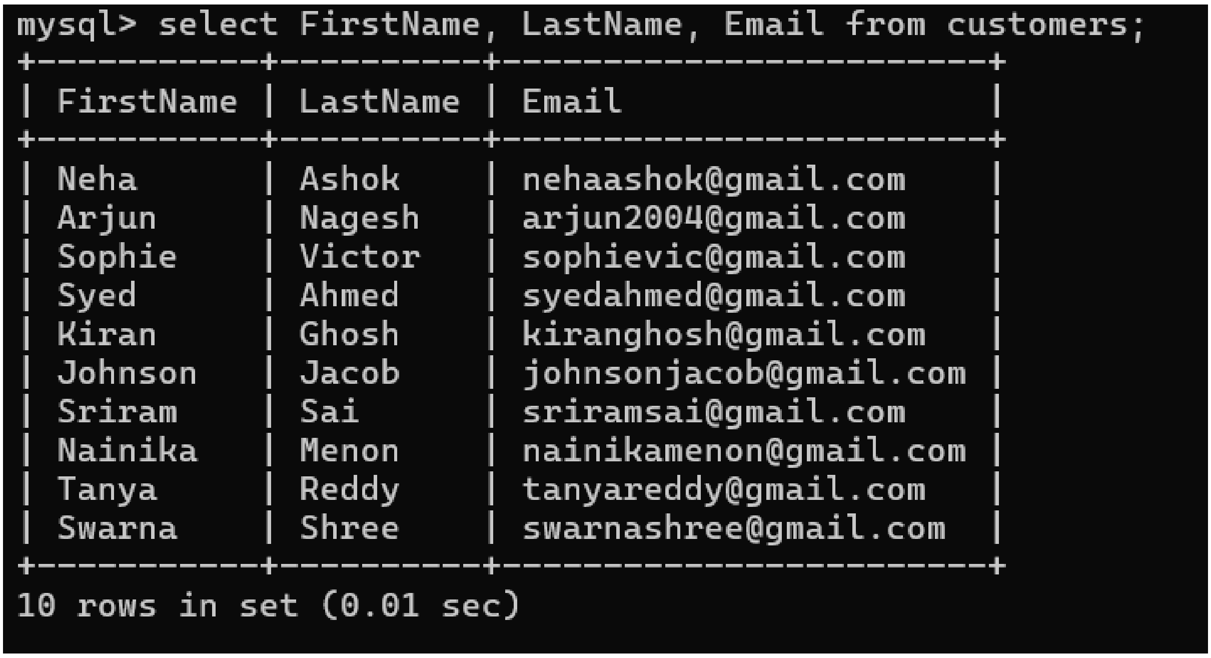
```
mysql> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate) values (4004, 1004, 30, '2025-06-10');
Query OK, 1 row affected (0.01 sec)

mysql> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate) values (4005, 1005, 25, '2025-06-10');
Query OK, 1 row affected (0.01 sec)

mysql> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate) values (4006, 1006, 15, '2025-06-10');
Query OK, 1 row affected (0.01 sec)

mysql> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate) values (4007, 1007, 18, '2025-06-10');
Query OK, 1 row affected (0.01 sec)

mysql> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate) values (4008, 1008, 12, '2025-06-10');
Query OK, 1 row affected (0.01 sec)

mysql> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate) values (4009, 1009, 30, '2025-06-10');
Query OK, 1 row affected (0.01 sec)

mysql> insert into inventory (inventoryid, productid, quantityinstock, laststockupdate) values (4010, 1010, 7, '2025-06-10');
Query OK, 1 row affected (0.01 sec)

mysql> select * from inventory;
+-------------+-----------+-----------------+-----------------+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-------------+-----------+-----------------+-----------------+
|        4001 |      1001 |              20 | 2025-06-10      |
|        4002 |      1002 |              50 | 2025-06-10      |
|        4003 |      1003 |              10 | 2025-06-10      |
|        4004 |      1004 |              30 | 2025-06-10      |
|        4005 |      1005 |              25 | 2025-06-10      |
|        4006 |      1006 |              15 | 2025-06-10      |
|        4007 |      1007 |              18 | 2025-06-10      |
|        4008 |      1008 |              12 | 2025-06-10      |
|        4009 |      1009 |              30 | 2025-06-10      |
|        4010 |      1010 |               7 | 2025-06-10      |
+-------------+-----------+-----------------+-----------------+
10 rows in set (0.01 sec)

mysql> 
```

Task:2 SELECT, WHERE, BETWEEN, LIKE:

Query Solutions:

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

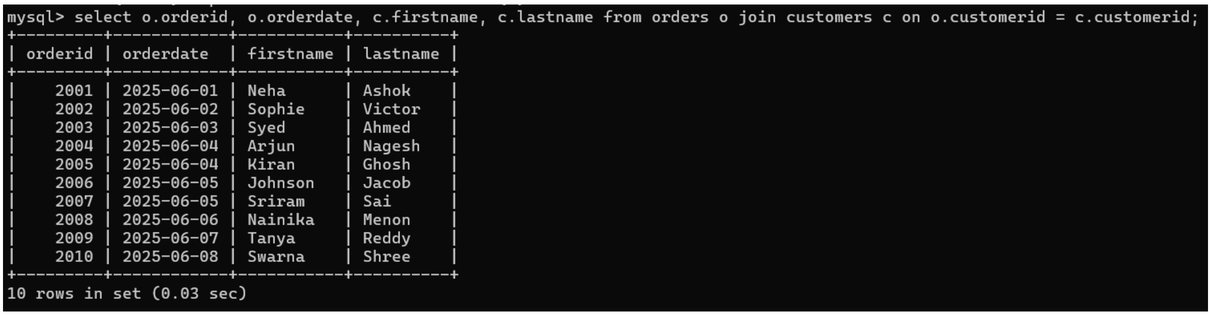
Query:
-> select FirstName, LastName, Email from customers;



```
mysql> select FirstName, LastName, Email from customers;
+-----------+----------+------------------------+
| FirstName | LastName | Email                  |
+-----------+----------+------------------------+
| Neha      | Ashok    | nehaashok@gmail.com    |
| Arjun     | Nagesh   | arjun2004@gmail.com    |
| Sophie    | Victor   | sophievic@gmail.com    |
| Syed      | Ahmed    | syedahmed@gmail.com    |
| Kiran     | Ghosh    | kiranghosh@gmail.com   |
| Johnson   | Jacob    | johnsonjacob@gmail.com |
| Sriram    | Sai      | sriramsai@gmail.com    |
| Nainika   | Menon    | nainikamenon@gmail.com |
| Tanya     | Reddy    | tanyareddy@gmail.com   |
| Swarna    | Shree    | swarnashree@gmail.com  |
+-----------+----------+------------------------+
10 rows in set (0.01 sec)
```

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

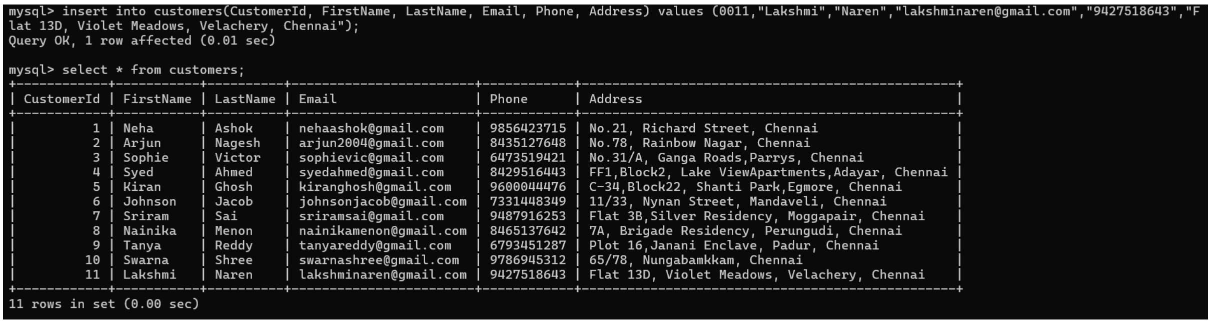
Query:
-> select o.orderid, o.orderdate, c.firstname, c.lastname from orders o join customers c on o.customerid = c.customerid;



```
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 |
+---------+------------+-----------+----------+
|    2001 | 2025-06-01 | Neha      | Ashok    |
|    2002 | 2025-06-02 | Sophie    | Victor   |
|    2003 | 2025-06-03 | Syed      | Ahmed    |
|    2004 | 2025-06-04 | Arjun     | Nagesh   |
|    2005 | 2025-06-04 | Kiran     | Ghosh    |
|    2006 | 2025-06-05 | Johnson   | Jacob    |
|    2007 | 2025-06-05 | Sriram    | Sai      |
|    2008 | 2025-06-06 | Nainika   | Menon    |
|    2009 | 2025-06-07 | Tanya     | Reddy    |
|    2010 | 2025-06-08 | Swarna    | Shree    |
+---------+------------+-----------+----------+
10 rows in set (0.03 sec)
```

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

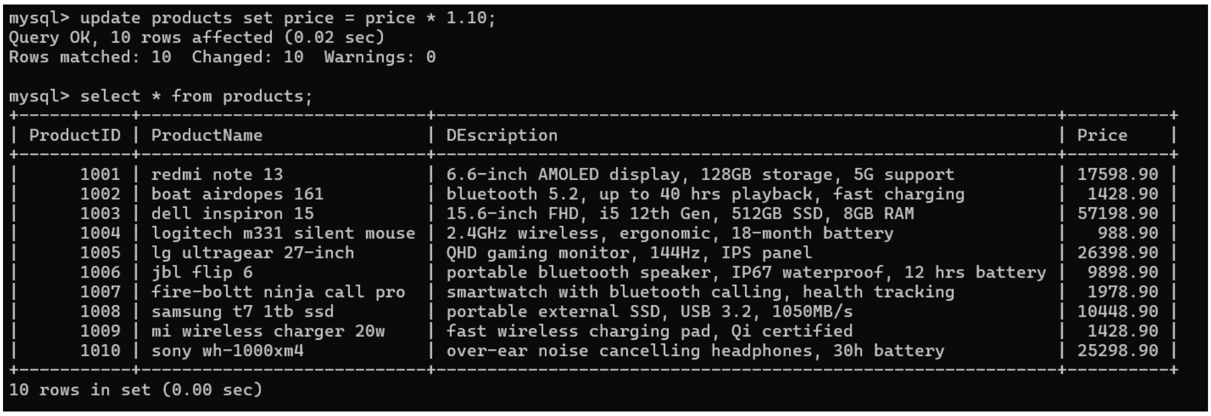
Query:
-> insert into customers(CustomerId, FirstName, LastName, Email, Phone, Address) values (0011,"Lakshmi","Naren","lakshminaren@gmail.com","9427518643","Flat 13D, Violet Meadows, Velachery, Chennai");



```
mysql> insert into customers(CustomerId, FirstName, LastName, Email, Phone, Address) values (0011,"Lakshmi","Naren","lakshminaren@gmail.com","9427518643","F
lat 13D, Violet Meadows, Velachery, Chennai");
Query OK, 1 row affected (0.01 sec)

mysql> select * from customers;
+------------+-----------+----------+------------------------+------------+-------------------------------------------------+
| CustomerId | FirstName | LastName | Email                  | Phone      | Address                                         |
+------------+-----------+----------+------------------------+------------+-------------------------------------------------+
|          1 | Neha      | Ashok    | nehaashok@gmail.com    | 9856423715 | No.21, Richard Street, Chennai                  |
|          2 | Arjun     | Nagesh   | arjun2004@gmail.com    | 8435127648 | No.78, Rainbow Nagar, Chennai                   |
|          3 | Sophie    | Victor   | sophievic@gmail.com    | 6473519421 | No.31/A, Ganga Roads,Parrys, Chennai            |
|          4 | Syed      | Ahmed    | syedahmed@gmail.com    | 8429516443 | FF1,Block2, Lake ViewApartments,Adayar, Chennai |
|          5 | Kiran     | Ghosh    | kiranghosh@gmail.com   | 9600044476 | C-34,Block22, Shanti Park,Egmore, Chennai       |
|          6 | Johnson   | Jacob    | johnsonjacob@gmail.com | 7331448349 | 11/33, Nynan Street, Mandaveli, Chennai         |
|          7 | Sriram    | Sai      | sriramsai@gmail.com    | 9487916253 | Flat 3B,Silver Residency, Moggapair, Chennai    |
|          8 | Nainika   | Menon    | nainikamenon@gmail.com | 8465137642 | 7A, Brigade Residency, Perungudi, Chennai       |
|          9 | Tanya     | Reddy    | tanyareddy@gmail.com   | 6793451287 | Plot 16,Janani Enclave, Padur, Chennai          |
|         10 | Swarna    | Shree    | swarnashree@gmail.com  | 9786945312 | 65/78, Nungabamkkam, Chennai                    |
|         11 | Lakshmi   | Naren    | lakshminaren@gmail.com | 9427518643 | Flat 13D, Violet Meadows, Velachery, Chennai    |
+------------+-----------+----------+------------------------+------------+-------------------------------------------------+
11 rows in set (0.00 sec)
```

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

Query:
-> update products set price = price * 1.10;



```
mysql> update products set price = price * 1.10;
Query OK, 10 rows affected (0.02 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from products;
+-----------+----------------------------+-------------------------------------------------------------+----------+
| ProductID | ProductName                | DEscription                                                 | Price    |
+-----------+----------------------------+-------------------------------------------------------------+----------+
|      1001 | redmi note 13              | 6.6-inch AMOLED display, 128GB storage, 5G support          | 17598.90 |
|      1002 | boat airdopes 161          | bluetooth 5.2, up to 40 hrs playback, fast charging         |  1428.90 |
|      1003 | dell inspiron 15           | 15.6-inch FHD, i5 12th Gen, 512GB SSD, 8GB RAM              | 57198.90 |
|      1004 | logitech m331 silent mouse | 2.4GHz wireless, ergonomic, 18-month battery                |   988.90 |
|      1005 | lg ultragear 27-inch       | QHD gaming monitor, 144Hz, IPS panel                        | 26398.90 |
|      1006 | jbl flip 6                 | portable bluetooth speaker, IP67 waterproof, 12 hrs battery |  9898.90 |
|      1007 | fire-boltt ninja call pro  | smartwatch with bluetooth calling, health tracking          |  1978.90 |
|      1008 | samsung t7 1tb ssd         | portable external SSD, USB 3.2, 1050MB/s                    | 10448.90 |
|      1009 | mi wireless charger 20w    | fast wireless charging pad, Qi certified                    |  1428.90 |
|      1010 | sony wh-1000xm4            | over-ear noise cancelling headphones, 30h battery           | 25298.90 |
+-----------+----------------------------+-------------------------------------------------------------+----------+
10 rows in set (0.00 sec)
```

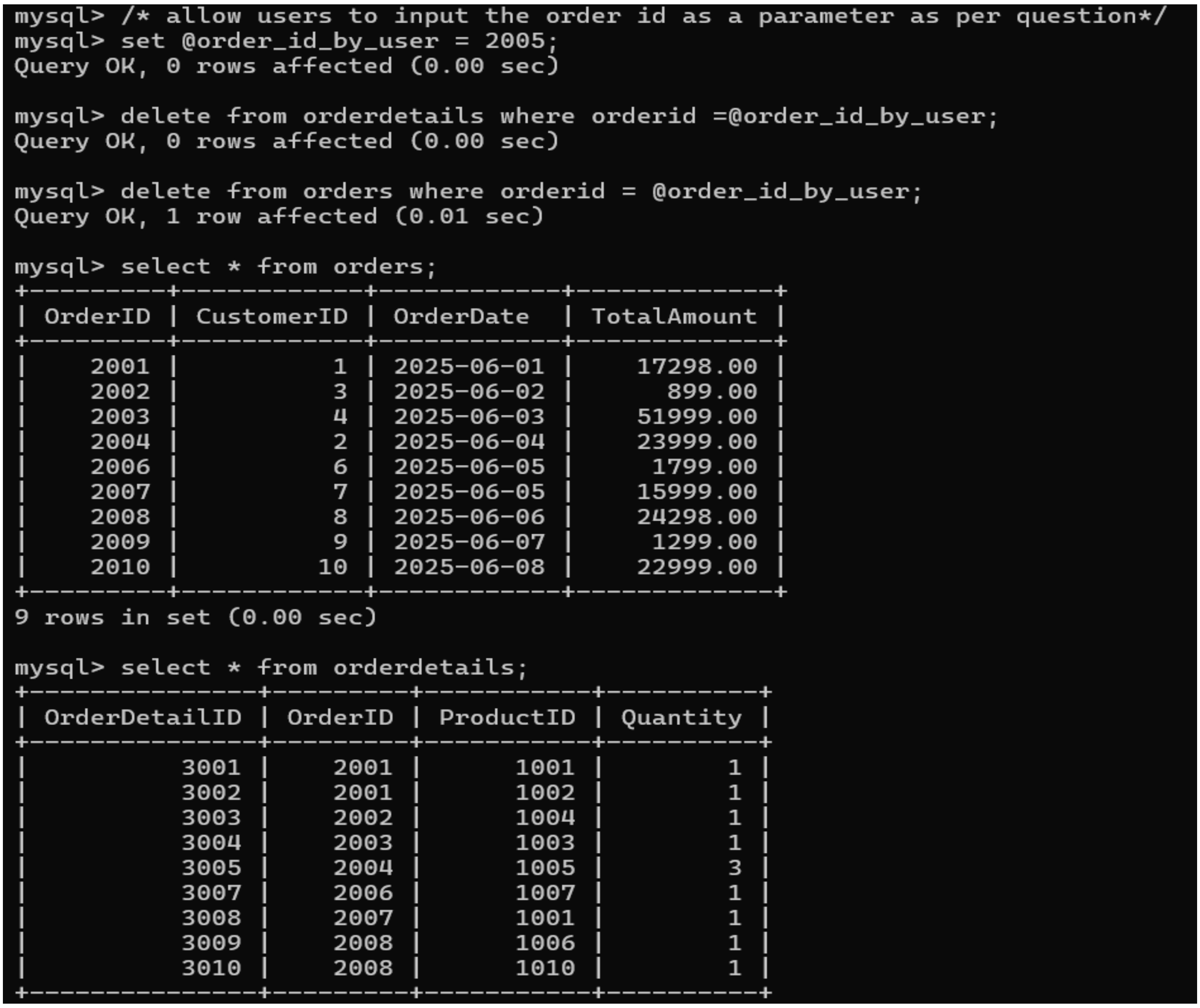
5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.

Query:
-> /* allow users to input the order id as a parameter as per question*/
-> set @order_id_by_user = 2005;

-> delete from orderdetails where orderid =@order_id_by_user;
Query OK, 0 rows affected (0.00 sec)

-> delete from orders where orderid = @order_id_by_user;
Query OK, 1 row affected (0.01 sec)

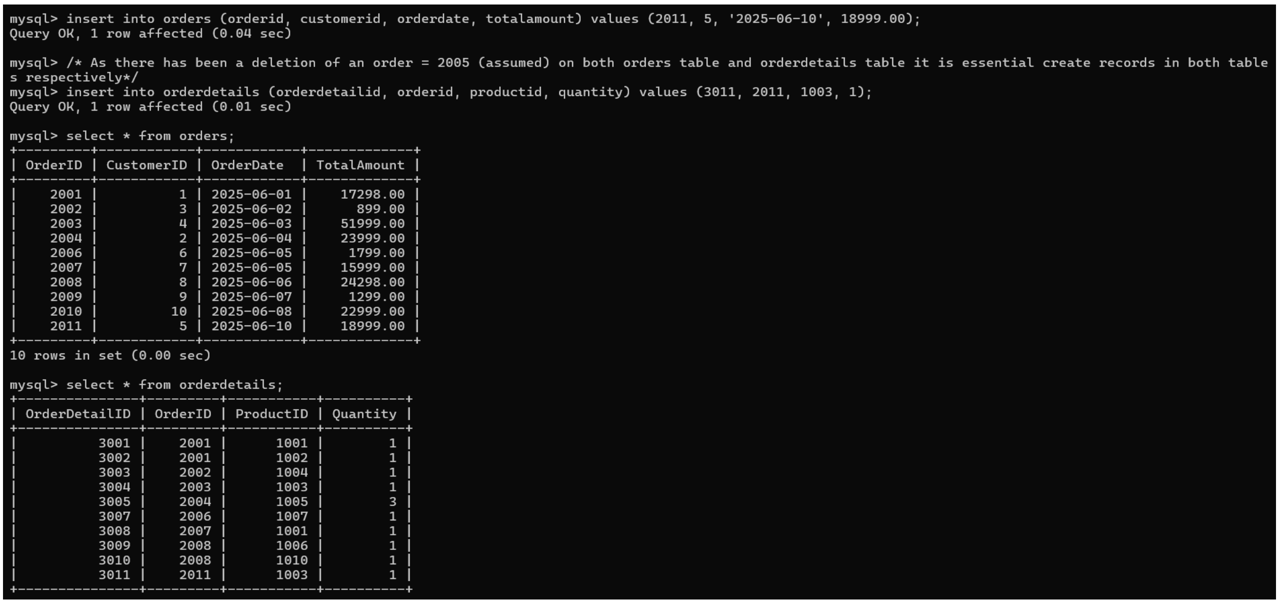
-> select * from orders;
-> select * from orderdetails;



```
mysql> /* allow users to input the order id as a parameter as per question*/
mysql> set @order_id_by_user = 2005;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from orderdetails where orderid =@order_id_by_user;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from orders where orderid = @order_id_by_user;
Query OK, 1 row affected (0.01 sec)

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|    2001 |          1 | 2025-06-01 |    17298.00 |
|    2002 |          3 | 2025-06-02 |      899.00 |
|    2003 |          4 | 2025-06-03 |    51999.00 |
|    2004 |          2 | 2025-06-04 |    23999.00 |
|    2006 |          6 | 2025-06-05 |     1799.00 |
|    2007 |          7 | 2025-06-05 |    15999.00 |
|    2008 |          8 | 2025-06-06 |    24298.00 |
|    2009 |          9 | 2025-06-07 |     1299.00 |
|    2010 |         10 | 2025-06-08 |    22999.00 |
+---------+------------+------------+-------------+
9 rows in set (0.00 sec)

mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|          3001 |    2001 |      1001 |        1 |
|          3002 |    2001 |      1002 |        1 |
|          3003 |    2002 |      1004 |        1 |
|          3004 |    2003 |      1003 |        1 |
|          3005 |    2004 |      1005 |        3 |
|          3007 |    2006 |      1007 |        1 |
|          3008 |    2007 |      1001 |        1 |
|          3009 |    2008 |      1006 |        1 |
|          3010 |    2008 |      1010 |        1 |
+---------------+---------+-----------+----------+
```

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.

Query:
-> insert into orders (orderid, customerid, orderdate, totalamount) values (2011, 5, '2025-06-10', 18999.00);
-> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3011, 2011, 1003, 1);



```
mysql> insert into orders (orderid, customerid, orderdate, totalamount) values (2011, 5, '2025-06-10', 18999.00);
Query OK, 1 row affected (0.04 sec)

mysql> /* As there has been a deletion of an order = 2005 (assumed) on both orders table and orderdetails table it is essential create records in both table
s respectively*/
mysql> insert into orderdetails (orderdetailid, orderid, productid, quantity) values (3011, 2011, 1003, 1);
Query OK, 1 row affected (0.01 sec)

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|    2001 |          1 | 2025-06-01 |    17298.00 |
|    2002 |          3 | 2025-06-02 |      899.00 |
|    2003 |          4 | 2025-06-03 |    51999.00 |
|    2004 |          2 | 2025-06-04 |    23999.00 |
|    2006 |          6 | 2025-06-05 |     1799.00 |
|    2007 |          7 | 2025-06-05 |    15999.00 |
|    2008 |          8 | 2025-06-06 |    24298.00 |
|    2009 |          9 | 2025-06-07 |     1299.00 |
|    2010 |         10 | 2025-06-08 |    22999.00 |
|    2011 |          5 | 2025-06-10 |    18999.00 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)

mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|          3001 |    2001 |      1001 |        1 |
|          3002 |    2001 |      1002 |        1 |
|          3003 |    2002 |      1004 |        1 |
|          3004 |    2003 |      1003 |        1 |
|          3005 |    2004 |      1005 |        3 |
|          3007 |    2006 |      1007 |        1 |
|          3008 |    2007 |      1001 |        1 |
|          3009 |    2008 |      1006 |        1 |
|          3010 |    2008 |      1010 |        1 |
|          3011 |    2011 |      1003 |        1 |
+---------------+---------+-----------+----------+
```

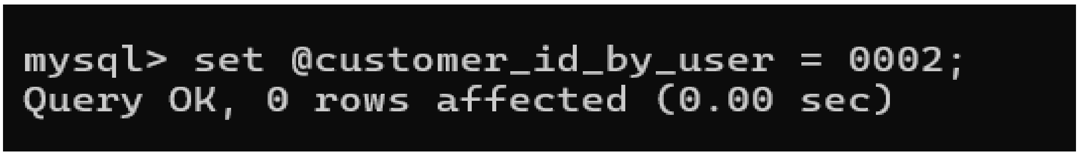
7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.

-> /* allow users to input the customer id as a parameter as per question*/
-> set @customer_id_by_user = 0002;

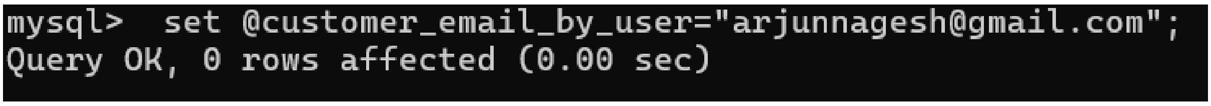
-> set @customer_email_by_user="arjunnagesh@gmail.com";

-> set @customer_address_by_user = "Plot 12/K, Anugraha Apartments, Anna Nagar, Chennai";

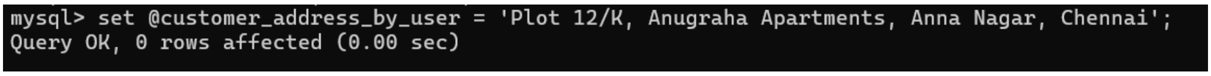
-> update customers set email = @customer_email_by_user, address = @customer_address_by_user where customerid = @customer_id_by_user;



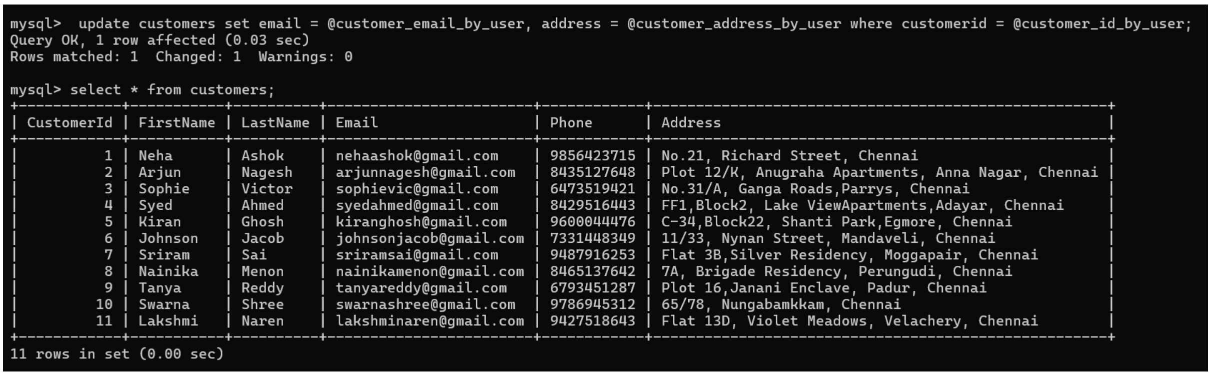
```
mysql> set @customer_id_by_user = 0002;
Query OK, 0 rows affected (0.00 sec)
```



```
mysql>  set @customer_email_by_user="arjunnagesh@gmail.com";
Query OK, 0 rows affected (0.00 sec)
```



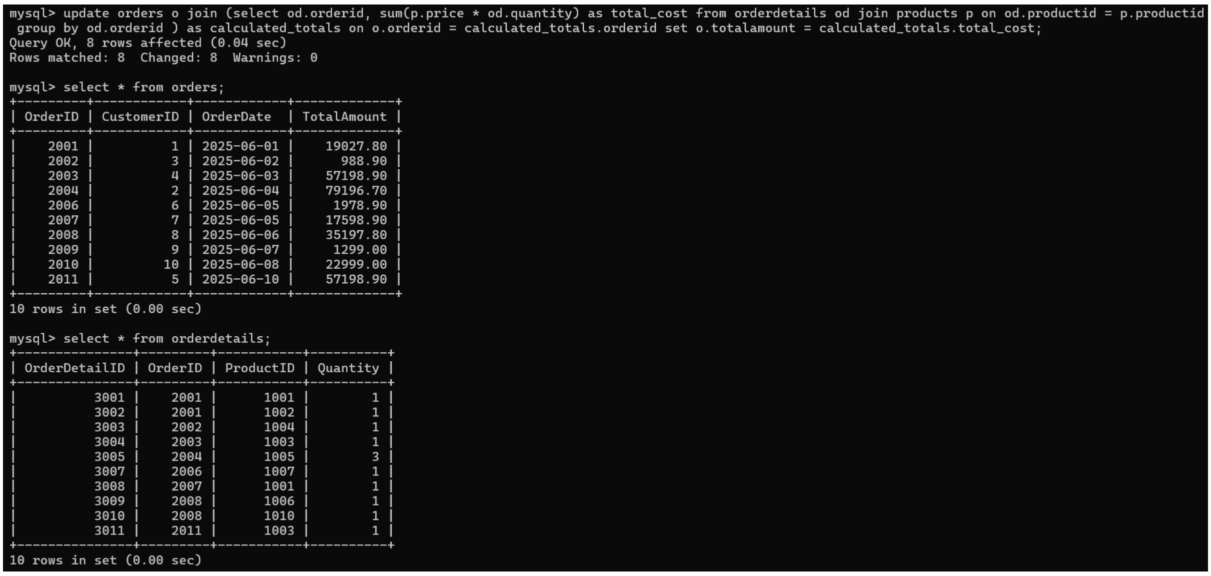
```
mysql> set @customer_address_by_user = 'Plot 12/K, Anugraha Apartments, Anna Nagar, Chennai';
Query OK, 0 rows affected (0.00 sec)
```



```
mysql>  update customers set email = @customer_email_by_user, address = @customer_address_by_user where customerid = @customer_id_by_user;
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                                             |
+------------+-----------+----------+------------------------+------------+-----------------------------------------------------+
|          1 | Neha      | Ashok    | nehaashok@gmail.com    | 9856423715 | No.21, Richard Street, Chennai                      |
|          2 | Arjun     | Nagesh   | arjunnagesh@gmail.com  | 8435127648 | Plot 12/K, Anugraha Apartments, Anna Nagar, Chennai |
|          3 | Sophie    | Victor   | sophievic@gmail.com    | 6473519421 | No.31/A, Ganga Roads,Parrys, Chennai                |
|          4 | Syed      | Ahmed    | syedahmed@gmail.com    | 8429516443 | FF1,Block2, Lake ViewApartments,Adayar, Chennai     |
|          5 | Kiran     | Ghosh    | kiranghosh@gmail.com   | 9600044476 | C-34,Block22, Shanti Park,Egmore, Chennai           |
|          6 | Johnson   | Jacob    | johnsonjacob@gmail.com | 7331448349 | 11/33, Nynan Street, Mandaveli, Chennai             |
|          7 | Sriram    | Sai      | sriramsai@gmail.com    | 9487916253 | Flat 3B,Silver Residency, Moggapair, Chennai        |
|          8 | Nainika   | Menon    | nainikamenon@gmail.com | 8465137642 | 7A, Brigade Residency, Perungudi, Chennai           |
|          9 | Tanya     | Reddy    | tanyareddy@gmail.com   | 6793451287 | Plot 16,Janani Enclave, Padur, Chennai              |
|         10 | Swarna    | Shree    | swarnashree@gmail.com  | 9786945312 | 65/78, Nungabamkkam, Chennai                        |
|         11 | Lakshmi   | Naren    | lakshminaren@gmail.com | 9427518643 | Flat 13D, Violet Meadows, Velachery, Chennai        |
+------------+-----------+----------+------------------------+------------+-----------------------------------------------------+
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.

Query:
-> update orders o join (select od.orderid, sum(p.price * od.quantity) as total_cost from orderdetails od join products p on od.productid = p.productid group by od.orderid) as calculated_totals on o.orderid = calculated_totals.orderid set o.totalamount = calculated_totals.total_cost;
Query OK, 8 rows affected (0.04 sec)



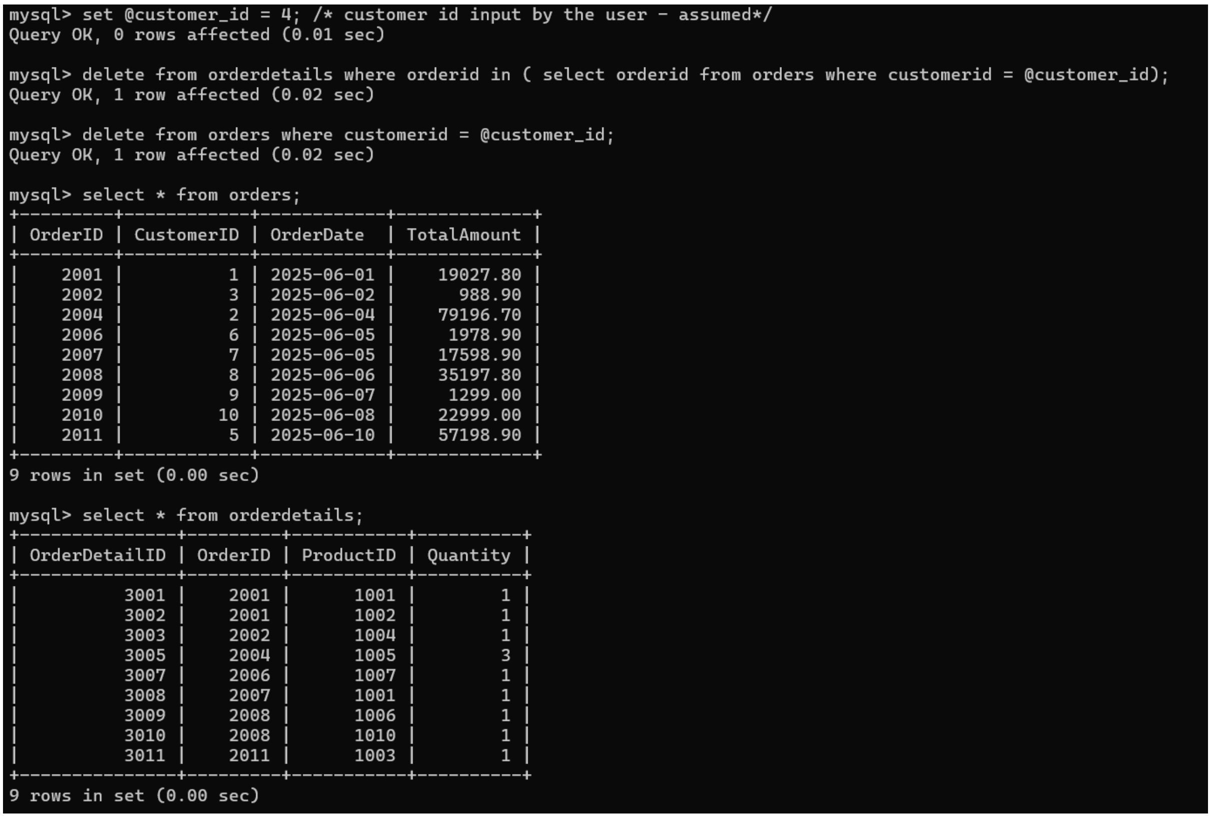
```
mysql> update orders o join (select od.orderid, sum(p.price * od.quantity) as total_cost from orderdetails od join products p on od.productid = p.productid
 group by od.orderid ) as calculated_totals on o.orderid = calculated_totals.orderid set o.totalamount = calculated_totals.total_cost;
Query OK, 8 rows affected (0.04 sec)
Rows matched: 8  Changed: 8  Warnings: 0

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|    2001 |          1 | 2025-06-01 |    19027.80 |
|    2002 |          3 | 2025-06-02 |      988.90 |
|    2003 |          4 | 2025-06-03 |    57198.90 |
|    2004 |          2 | 2025-06-04 |    79196.70 |
|    2006 |          6 | 2025-06-05 |     1978.90 |
|    2007 |          7 | 2025-06-05 |    17598.90 |
|    2008 |          8 | 2025-06-06 |    35197.80 |
|    2009 |          9 | 2025-06-07 |     1299.00 |
|    2010 |         10 | 2025-06-08 |    22999.00 |
|    2011 |          5 | 2025-06-10 |    57198.90 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)

mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|          3001 |    2001 |      1001 |        1 |
|          3002 |    2001 |      1002 |        1 |
|          3003 |    2002 |      1004 |        1 |
|          3004 |    2003 |      1003 |        1 |
|          3005 |    2004 |      1005 |        3 |
|          3007 |    2006 |      1007 |        1 |
|          3008 |    2007 |      1001 |        1 |
|          3009 |    2008 |      1006 |        1 |
|          3010 |    2008 |      1010 |        1 |
|          3011 |    2011 |      1003 |        1 |
+---------------+---------+-----------+----------+
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.

Query:
-> set @customer_id = 4; /* customer id input by the user - assumed*/
Query OK, 0 rows affected (0.01 sec)

mysql> delete from orderdetails where orderid in (select orderid from orders where customerid = @customer_id);
Query OK, 1 row affected (0.02 sec)

mysql> delete from orders where customerid = @customer_id;
Query OK, 1 row affected (0.02 sec)

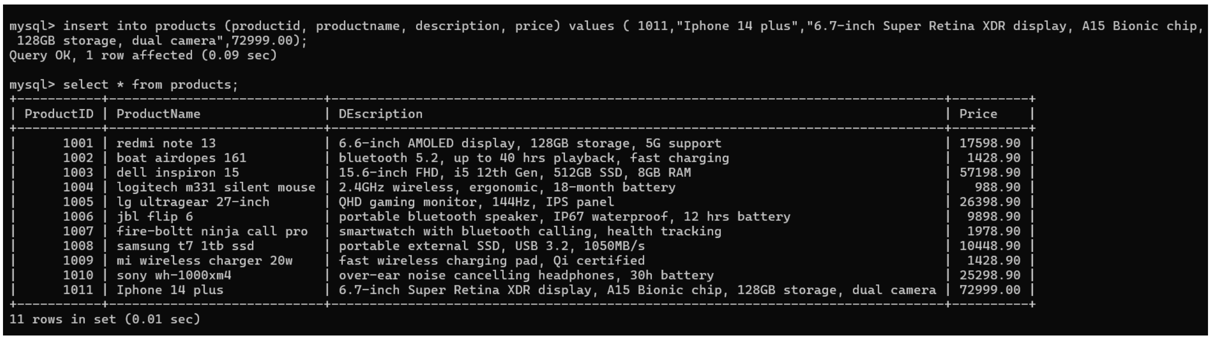


```
mysql> set @customer_id = 4; /* customer id input by the user - assumed*/
Query OK, 0 rows affected (0.01 sec)

mysql> delete from orderdetails where orderid in ( select orderid from orders where customerid = @customer_id);
Query OK, 1 row affected (0.02 sec)

mysql> delete from orders where customerid = @customer_id;
Query OK, 1 row affected (0.02 sec)

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|    2001 |          1 | 2025-06-01 |    19027.80 |
|    2002 |          3 | 2025-06-02 |      988.90 |
|    2004 |          2 | 2025-06-04 |    79196.70 |
|    2006 |          6 | 2025-06-05 |     1978.90 |
|    2007 |          7 | 2025-06-05 |    17598.90 |
|    2008 |          8 | 2025-06-06 |    35197.80 |
|    2009 |          9 | 2025-06-07 |     1299.00 |
|    2010 |         10 | 2025-06-08 |    22999.00 |
|    2011 |          5 | 2025-06-10 |    57198.90 |
+---------+------------+------------+-------------+
9 rows in set (0.00 sec)

mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|          3001 |    2001 |      1001 |        1 |
|          3002 |    2001 |      1002 |        1 |
|          3003 |    2002 |      1004 |        1 |
|          3005 |    2004 |      1005 |        3 |
|          3007 |    2006 |      1007 |        1 |
|          3008 |    2007 |      1001 |        1 |
|          3009 |    2008 |      1006 |        1 |
|          3010 |    2008 |      1010 |        1 |
|          3011 |    2011 |      1003 |        1 |
+---------------+---------+-----------+----------+
9 rows in set (0.00 sec)
```

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

Query:
->

```
mysql> insert into products (productid, productname, description, price) values ( 1011,"Iphone 14 plus","6.7-inch Super Retina XDR display, A15 Bionic chip, 128GB storage, dual camera",72999.00);
Query OK, 1 row affected (0.09 sec)
```

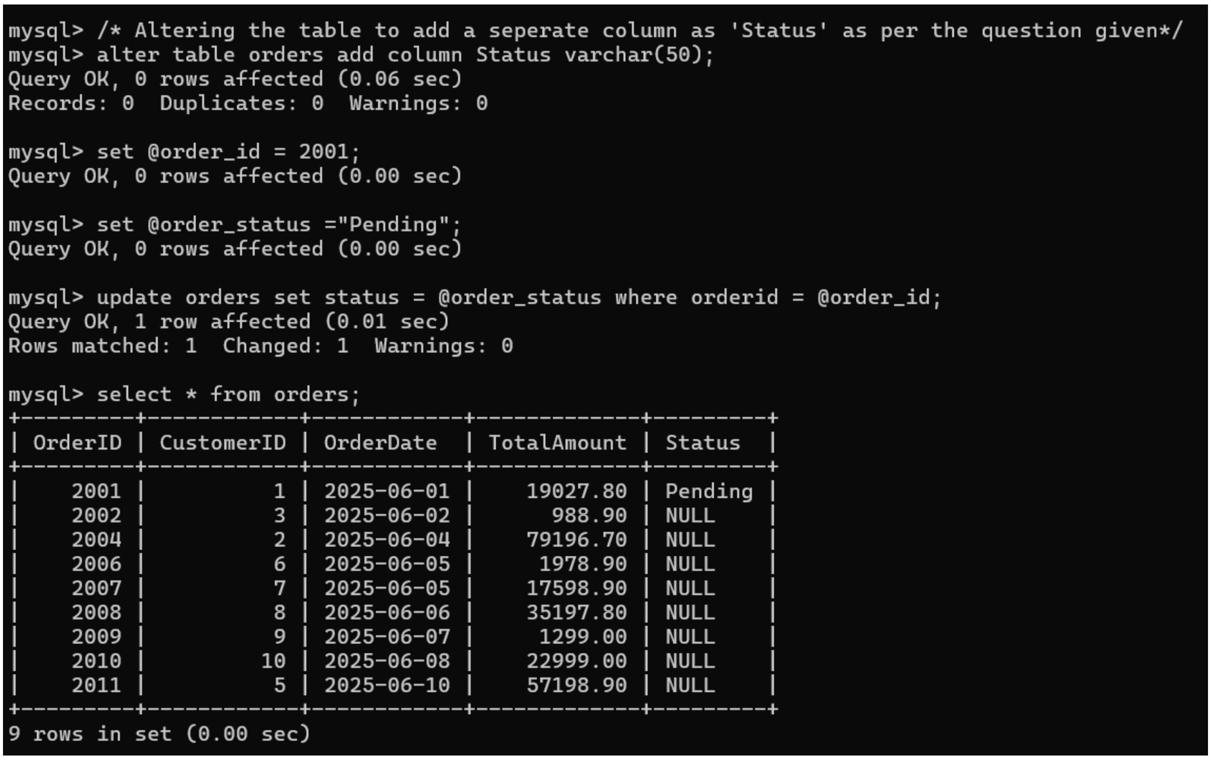


```
mysql> insert into products (productid, productname, description, price) values ( 1011,"Iphone 14 plus","6.7-inch Super Retina XDR display, A15 Bionic chip,
 128GB storage, dual camera",72999.00);
Query OK, 1 row affected (0.09 sec)

mysql> select * from products;
+-----------+----------------------------+--------------------------------------------------------------------------------+----------+
| ProductID | ProductName                | DEscription                                                                    | Price    |
+-----------+----------------------------+--------------------------------------------------------------------------------+----------+
|      1001 | redmi note 13              | 6.6-inch AMOLED display, 128GB storage, 5G support                             | 17598.90 |
|      1002 | boat airdopes 161          | bluetooth 5.2, up to 40 hrs playback, fast charging                            |  1428.90 |
|      1003 | dell inspiron 15           | 15.6-inch FHD, i5 12th Gen, 512GB SSD, 8GB RAM                                 | 57198.90 |
|      1004 | logitech m331 silent mouse | 2.4GHz wireless, ergonomic, 18-month battery                                   |   988.90 |
|      1005 | lg ultragear 27-inch       | QHD gaming monitor, 144Hz, IPS panel                                           | 26398.90 |
|      1006 | jbl flip 6                 | portable bluetooth speaker, IP67 waterproof, 12 hrs battery                    |  9898.90 |
|      1007 | fire-boltt ninja call pro  | smartwatch with bluetooth calling, health tracking                             |  1978.90 |
|      1008 | samsung t7 1tb ssd         | portable external SSD, USB 3.2, 1050MB/s                                       | 10448.90 |
|      1009 | mi wireless charger 20w    | fast wireless charging pad, Qi certified                                       |  1428.90 |
|      1010 | sony wh-1000xm4            | over-ear noise cancelling headphones, 30h battery                              | 25298.90 |
|      1011 | Iphone 14 plus             | 6.7-inch Super Retina XDR display, A15 Bionic chip, 128GB storage, dual camera | 72999.00 |
+-----------+----------------------------+--------------------------------------------------------------------------------+----------+
11 rows in set (0.01 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.

Query:

```
-> /* Altering the table to add a seperate column as 'Status' as per the question given*/
mysql> alter table orders add column Status varchar(50);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> set @order_id = 2001;
Query OK, 0 rows affected (0.00 sec)

mysql> set @order_status ="Pending";
Query OK, 0 rows affected (0.00 sec)

mysql> update orders set status = @order_status where orderid = @order_id;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> select * from orders;
```



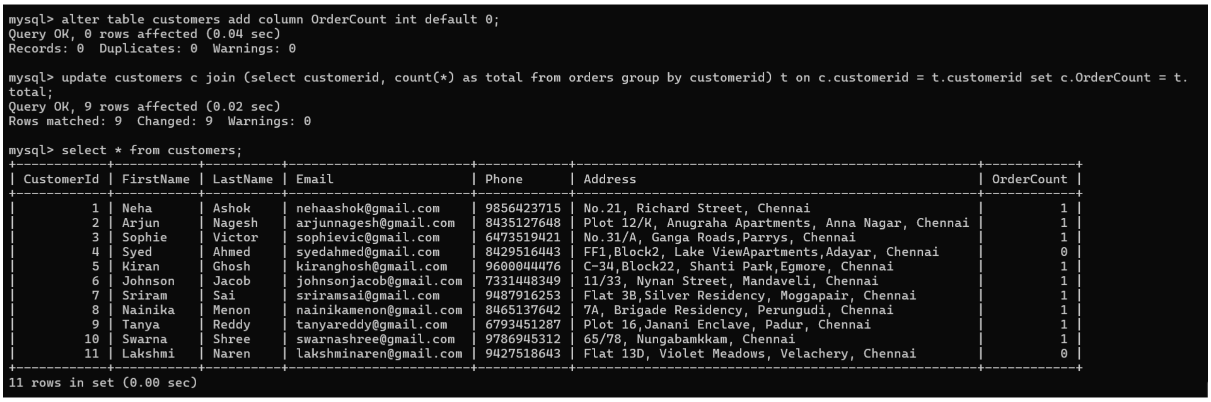
```
mysql> /* Altering the table to add a seperate column as 'Status' as per the question given*/
mysql> alter table orders add column Status varchar(50);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> set @order_id = 2001;
Query OK, 0 rows affected (0.00 sec)

mysql> set @order_status ="Pending";
Query OK, 0 rows affected (0.00 sec)

mysql> update orders set status = @order_status where orderid = @order_id;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from orders;
+---------+------------+------------+-------------+---------+
| OrderID | CustomerID | OrderDate  | TotalAmount | Status  |
+---------+------------+------------+-------------+---------+
|    2001 |          1 | 2025-06-01 |    19027.80 | Pending |
|    2002 |          3 | 2025-06-02 |      988.90 | NULL    |
|    2004 |          2 | 2025-06-04 |    79196.70 | NULL    |
|    2006 |          6 | 2025-06-05 |     1978.90 | NULL    |
|    2007 |          7 | 2025-06-05 |    17598.90 | NULL    |
|    2008 |          8 | 2025-06-06 |    35197.80 | NULL    |
|    2009 |          9 | 2025-06-07 |     1299.00 | NULL    |
|    2010 |         10 | 2025-06-08 |    22999.00 | NULL    |
|    2011 |          5 | 2025-06-10 |    57198.90 | 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.

Query:
-> alter table customers add column OrderCount int default 0;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> update customers c join (select customerid, count(*) as total from orders group by customerid) t on c.customerid = t.customerid set c.OrderCount = t.total;
Query OK, 9 rows affected (0.02 sec)
Rows matched: 9 Changed: 9 Warnings: 0

mysql> select * from customers;



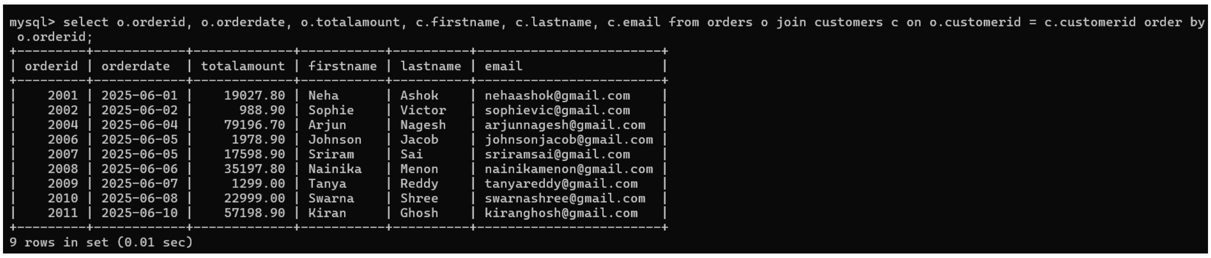
```
mysql> alter table customers add column OrderCount int default 0;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> update customers c join (select customerid, count(*) as total from orders group by customerid) t on c.customerid = t.customerid set c.OrderCount = t.
total;
Query OK, 9 rows affected (0.02 sec)
Rows matched: 9  Changed: 9  Warnings: 0

mysql> select * from customers;
+------------+-----------+----------+------------------------+------------+-----------------------------------------------------+------------+
| CustomerId | FirstName | LastName | Email                  | Phone      | Address                                             | OrderCount |
+------------+-----------+----------+------------------------+------------+-----------------------------------------------------+------------+
|          1 | Neha      | Ashok    | nehaashok@gmail.com    | 9856423715 | No.21, Richard Street, Chennai                      |          1 |
|          2 | Arjun     | Nagesh   | arjunnagesh@gmail.com  | 8435127648 | Plot 12/K, Anugraha Apartments, Anna Nagar, Chennai |          1 |
|          3 | Sophie    | Victor   | sophievic@gmail.com    | 6473519421 | No.31/A, Ganga Roads,Parrys, Chennai                |          1 |
|          4 | Syed      | Ahmed    | syedahmed@gmail.com    | 8429516443 | FF1,Block2, Lake ViewApartments,Adayar, Chennai     |          0 |
|          5 | Kiran     | Ghosh    | kiranghosh@gmail.com   | 9600044476 | C-34,Block22, Shanti Park,Egmore, Chennai           |          1 |
|          6 | Johnson   | Jacob    | johnsonjacob@gmail.com | 7331448349 | 11/33, Nynan Street, Mandaveli, Chennai             |          1 |
|          7 | Sriram    | Sai      | sriramsai@gmail.com    | 9487916253 | Flat 3B,Silver Residency, Moggapair, Chennai        |          1 |
|          8 | Nainika   | Menon    | nainikamenon@gmail.com | 8465137642 | 7A, Brigade Residency, Perungudi, Chennai           |          1 |
|          9 | Tanya     | Reddy    | tanyareddy@gmail.com   | 6793451287 | Plot 16,Janani Enclave, Padur, Chennai              |          1 |
|         10 | Swarna    | Shree    | swarnashree@gmail.com  | 9786945312 | 65/78, Nungabamkkam, Chennai                        |          1 |
|         11 | Lakshmi   | Naren    | lakshminaren@gmail.com | 9427518643 | Flat 13D, Violet Meadows, Velachery, Chennai        |          0 |
+------------+-----------+----------+------------------------+------------+-----------------------------------------------------+------------+
11 rows in set (0.00 sec)
```

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

Query Solutions:

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

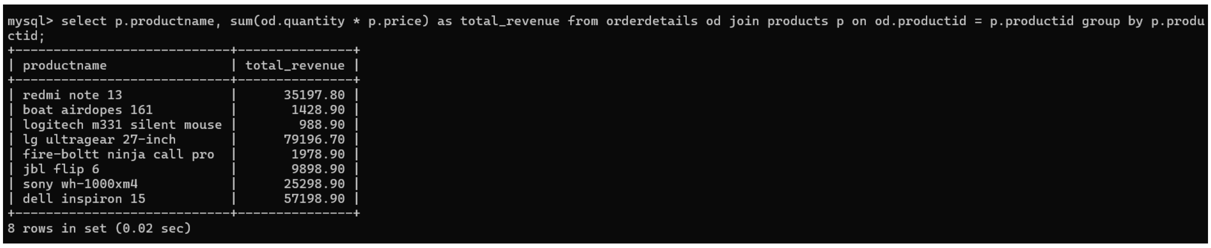
Query:
->mysql> select o.orderid, o.orderdate, o.totalamount, c.firstname, c.lastname, c.email from orders o join customers c on o.customerid = c.customerid order by o.orderid;



```
mysql> select o.orderid, o.orderdate, o.totalamount, c.firstname, c.lastname, c.email from orders o join customers c on o.customerid = c.customerid order by
 o.orderid;
+---------+------------+-------------+-----------+----------+------------------------+
| orderid | orderdate  | totalamount | firstname | lastname | email                  |
+---------+------------+-------------+-----------+----------+------------------------+
|    2001 | 2025-06-01 |    19027.80 | Neha      | Ashok    | nehaashok@gmail.com    |
|    2002 | 2025-06-02 |      988.90 | Sophie    | Victor   | sophievic@gmail.com    |
|    2004 | 2025-06-04 |    79196.70 | Arjun     | Nagesh   | arjunnagesh@gmail.com  |
|    2006 | 2025-06-05 |     1978.90 | Johnson   | Jacob    | johnsonjacob@gmail.com |
|    2007 | 2025-06-05 |    17598.90 | Sriram    | Sai      | sriramsai@gmail.com    |
|    2008 | 2025-06-06 |    35197.80 | Nainika   | Menon    | nainikamenon@gmail.com |
|    2009 | 2025-06-07 |     1299.00 | Tanya     | Reddy    | tanyareddy@gmail.com   |
|    2010 | 2025-06-08 |    22999.00 | Swarna    | Shree    | swarnashree@gmail.com  |
|    2011 | 2025-06-10 |    57198.90 | Kiran     | Ghosh    | kiranghosh@gmail.com   |
+---------+------------+-------------+-----------+----------+------------------------+
9 rows in set (0.01 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.

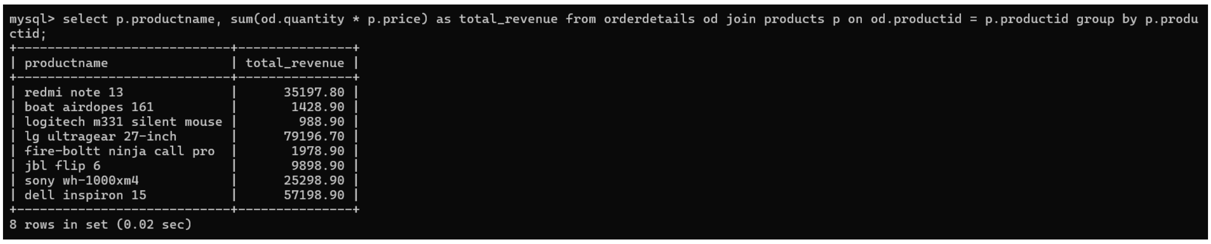
Query:
-> select p.productname, sum(od.quantity * p.price) as total_revenue from orderdetails od join products p on od.productid = p.productid group by p.productid;



```
mysql> select p.productname, sum(od.quantity * p.price) as total_revenue from orderdetails od join products p on od.productid = p.productid group by p.produ
ctid;
+----------------------------+---------------+
| productname                | total_revenue |
+----------------------------+---------------+
| redmi note 13              |      35197.80 |
| boat airdopes 161          |       1428.90 |
| logitech m331 silent mouse |        988.90 |
| lg ultragear 27-inch       |      79196.70 |
| fire-boltt ninja call pro  |       1978.90 |
| jbl flip 6                 |       9898.90 |
| sony wh-1000xm4            |      25298.90 |
| dell inspiron 15           |      57198.90 |
+----------------------------+---------------+
8 rows in set (0.02 sec)
```

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

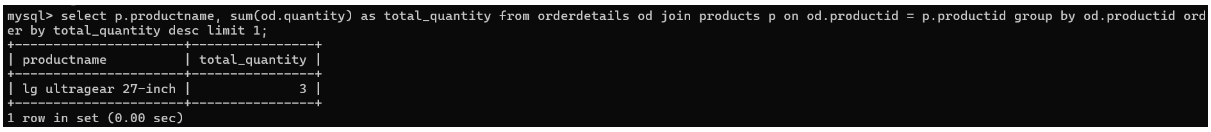
Query:
-> select p.productname, sum(od.quantity * p.price) as total_revenue from orderdetails od join products p on od.productid = p.productid group by p.productid;



```
mysql> select p.productname, sum(od.quantity * p.price) as total_revenue from orderdetails od join products p on od.productid = p.productid group by p.produ
ctid;
+----------------------------+---------------+
| productname                | total_revenue |
+----------------------------+---------------+
| redmi note 13              |      35197.80 |
| boat airdopes 161          |       1428.90 |
| logitech m331 silent mouse |        988.90 |
| lg ultragear 27-inch       |      79196.70 |
| fire-boltt ninja call pro  |       1978.90 |
| jbl flip 6                 |       9898.90 |
| sony wh-1000xm4            |      25298.90 |
| dell inspiron 15           |      57198.90 |
+----------------------------+---------------+
8 rows in set (0.02 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.

Query:
-> select p.productname, sum(od.quantity) as total_quantity from orderdetails od join products p on od.productid = p.productid group by od.productid order by total_quantity desc limit 1;



```
mysql> select p.productname, sum(od.quantity) as total_quantity from orderdetails od join products p on od.productid = p.productid group by od.productid ord
er by total_quantity desc limit 1;
+----------------------+----------------+
| productname          | total_quantity |
+----------------------+----------------+
| lg ultragear 27-inch |              3 |
+----------------------+----------------+
1 row in set (0.00 sec)
```

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

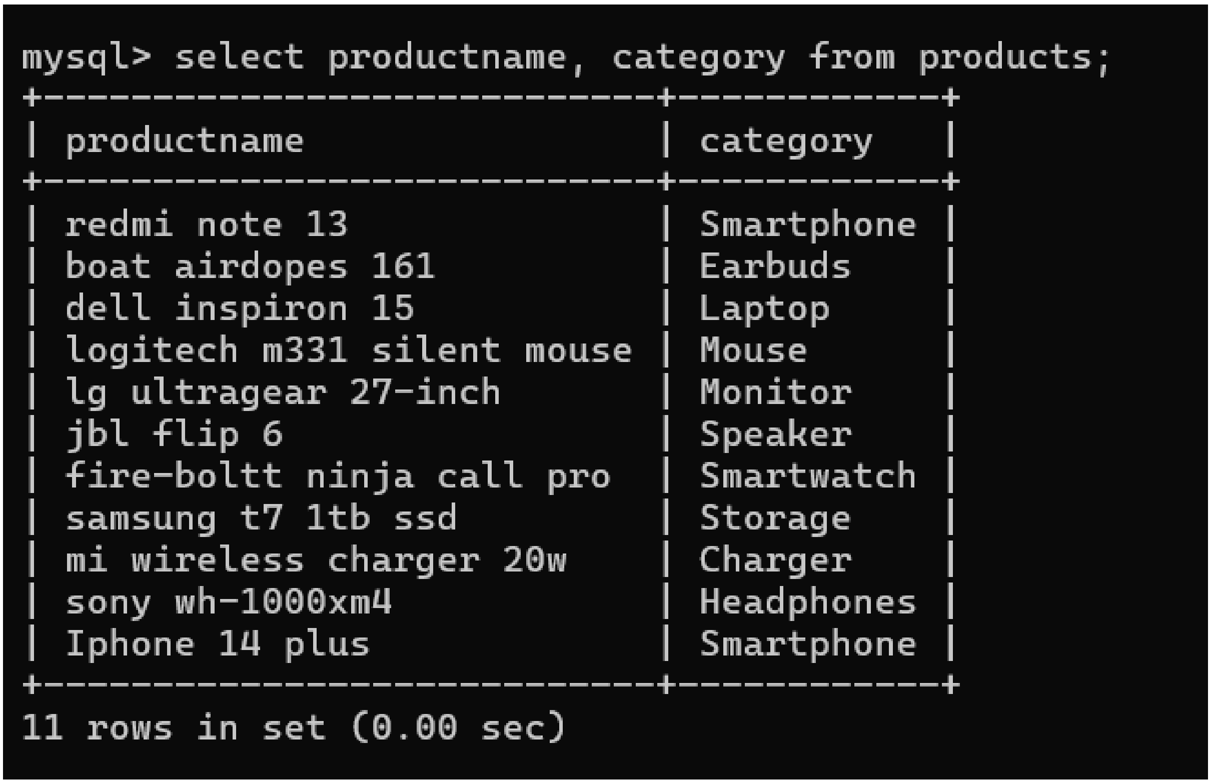
Query:
-> /* Altering the table to add a seperate column as 'Category' as per the question given*/
mysql> alter table products add column category varchar(50);
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> update products set category = 'Smartphone' where productname = 'redmi note 13';

mysql> update products set category = 'Earbuds' where productname = 'boat airdopes 161';

mysql> update products set category = 'Laptop' where productname = 'dell inspiron 15';

mysql> update products set category = 'Mouse' where productname = 'logitech m331 silent mouse';

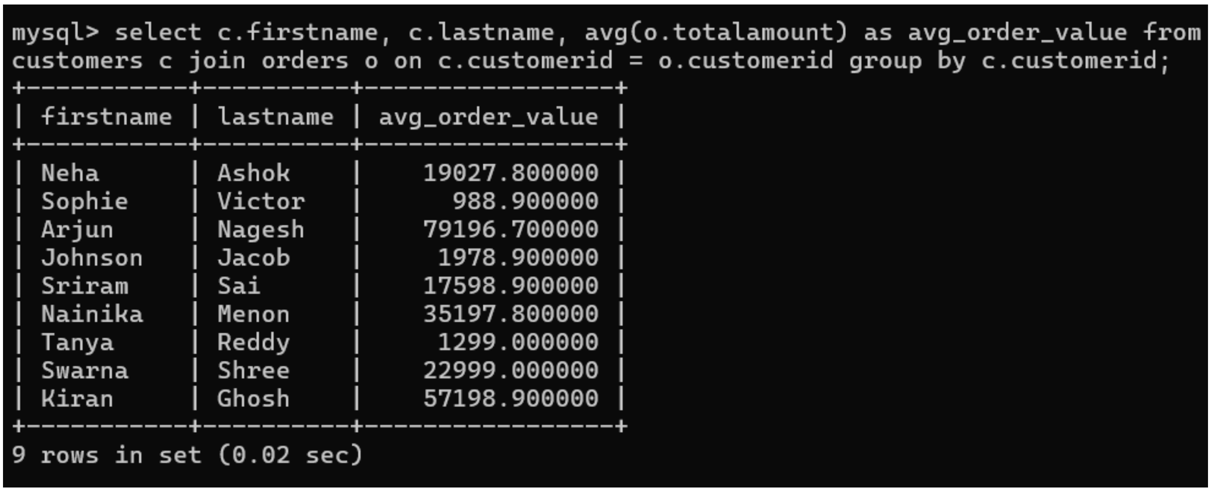
```
mysql> update products set category = 'Monitor' where productname = 'lg ultragear 27-inch';

mysql> update products set category = 'Speaker' where productname = 'jbl flip 6';

mysql> update products set category = 'Smartwatch' where productname = 'fire-boltt ninja call pro';

mysql> update products set category = 'Storage' where productname = 'samsung t7 1tb ssd';

mysql> update products set category = 'Charger' where productname = 'mi wireless charger 20w';

mysql> update products set category = 'Headphones' where productname = 'sony wh-1000xm4';

mysql> update products set category = 'Smartphone' where productname = 'Iphone 14 plus';


mysql> select * from products;
```



```
mysql> select productname, category from products;
+-----------------------------+------------+
| productname                 | category   |
+-----------------------------+------------+
| redmi note 13               | Smartphone |
| boat airdopes 161           | Earbuds    |
| dell inspiron 15            | Laptop     |
| logitech m331 silent mouse  | Mouse      |
| lg ultragear 27-inch        | Monitor    |
| jbl flip 6                  | Speaker    |
| fire-boltt ninja call pro   | Smartwatch |
| samsung t7 1tb ssd          | Storage    |
| mi wireless charger 20w     | Charger    |
| sony wh-1000xm4             | Headphones |
| Iphone 14 plus              | Smartphone |
+-----------------------------+------------+
11 rows in set (0.00 sec)
```

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

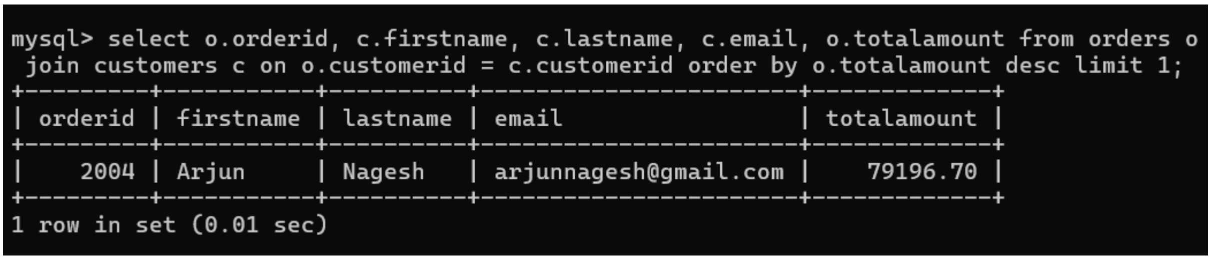
Query:
-> select c.firstname, c.lastname, avg(o.totalamount) as avg_order_value from customers c join orders o on c.customerid = o.customerid group by c.customerid;



```
mysql> select c.firstname, c.lastname, avg(o.totalamount) as avg_order_value from
customers c join orders o on c.customerid = o.customerid group by c.customerid;
+-----------+----------+-----------------+
| firstname | lastname | avg_order_value |
+-----------+----------+-----------------+
| Neha      | Ashok    |    19027.800000 |
| Sophie    | Victor   |      988.900000 |
| Arjun     | Nagesh   |    79196.700000 |
| Johnson   | Jacob    |     1978.900000 |
| Sriram    | Sai      |    17598.900000 |
| Nainika   | Menon    |    35197.800000 |
| Tanya     | Reddy    |     1299.000000 |
| Swarna    | Shree    |    22999.000000 |
| Kiran     | Ghosh    |    57198.900000 |
+-----------+----------+-----------------+
9 rows in set (0.02 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.

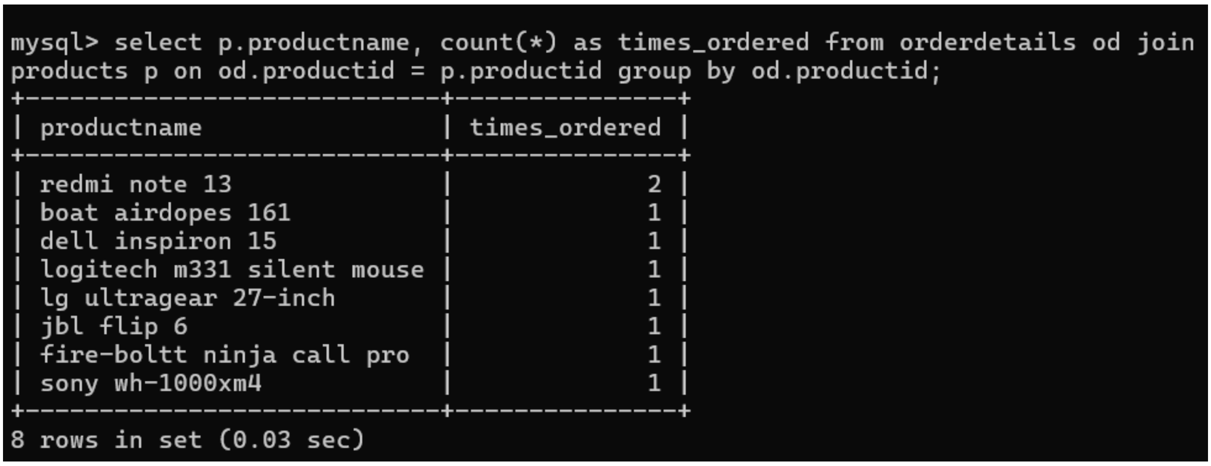
Query:
-> select o.orderid, c.firstname, c.lastname, c.email, o.totalamount from orders o join customers c on o.customerid = c.customerid order by o.totalamount desc limit 1;



```
mysql> select o.orderid, c.firstname, c.lastname, c.email, o.totalamount from orders o
 join customers c on o.customerid = c.customerid order by o.totalamount desc limit 1;
+---------+-----------+----------+-----------------------+-------------+
| orderid | firstname | lastname | email                 | totalamount |
+---------+-----------+----------+-----------------------+-------------+
|    2004 | Arjun     | Nagesh   | arjunnagesh@gmail.com |    79196.70 |
+---------+-----------+----------+-----------------------+-------------+
1 row in set (0.01 sec)
```

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

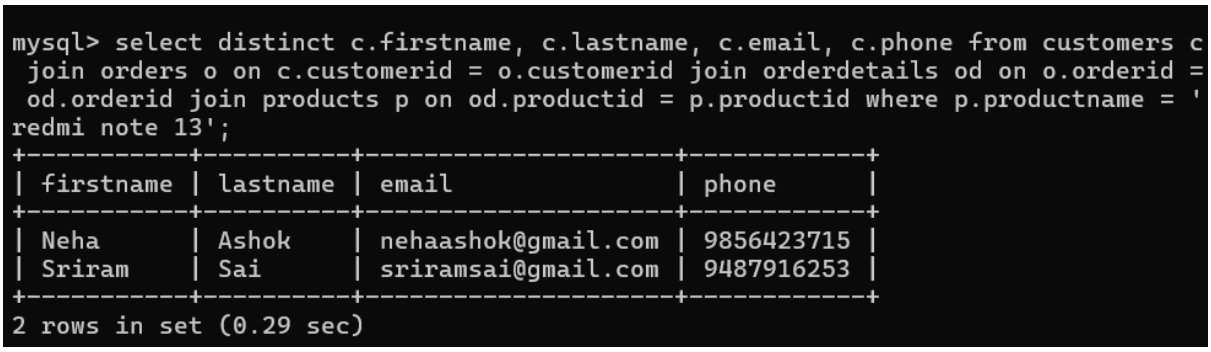
Query:
-> select p.productname, count(*) as times_ordered from orderdetails od join products p on od.productid = p.productid group by od.productid;



```
mysql> select p.productname, count(*) as times_ordered from orderdetails od join
products p on od.productid = p.productid group by od.productid;
+----------------------------+---------------+
| productname                | times_ordered |
+----------------------------+---------------+
| redmi note 13              |             2 |
| boat airdopes 161          |             1 |
| dell inspiron 15           |             1 |
| logitech m331 silent mouse |             1 |
| lg ultragear 27-inch       |             1 |
| jbl flip 6                 |             1 |
| fire-boltt ninja call pro  |             1 |
| sony wh-1000xm4            |             1 |
+----------------------------+---------------+
8 rows in set (0.03 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.

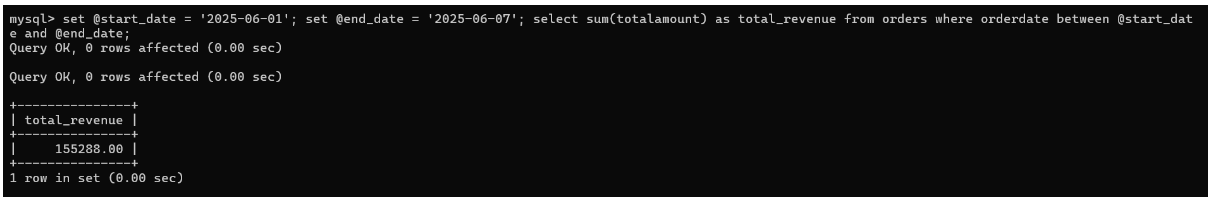
Query:
-> select distinct c.firstname, c.lastname, c.email, c.phone from customers c join orders o on c.customerid = o.customerid join orderdetails od on o.orderid = od.orderid join products p on od.productid = p.productid where p.productname = 'redmi note 13';



```
mysql> select distinct c.firstname, c.lastname, c.email, c.phone from customers c
 join orders o on c.customerid = o.customerid join orderdetails od on o.orderid =
 od.orderid join products p on od.productid = p.productid where p.productname = '
redmi note 13';
+-----------+----------+---------------------+------------+
| firstname | lastname | email               | phone      |
+-----------+----------+---------------------+------------+
| Neha      | Ashok    | nehaashok@gmail.com | 9856423715 |
| Sriram    | Sai      | sriramsai@gmail.com | 9487916253 |
+-----------+----------+---------------------+------------+
2 rows in set (0.29 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.

Query:
-> set @start_date = '2025-06-01'; set @end_date = '2025-06-07'; select sum(totalamount) as total_revenue from orders where orderdate between @start_date and @end_date;

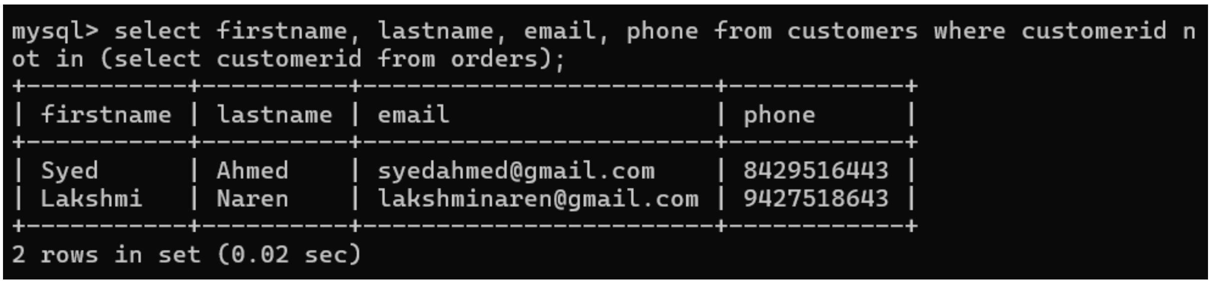


```
mysql> set @start_date = '2025-06-01'; set @end_date = '2025-06-07'; select sum(totalamount) as total_revenue from orders where orderdate between @start_dat
e and @end_date;
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+---------------+
| total_revenue |
+---------------+
|     155288.00 |
+---------------+
1 row in set (0.00 sec)
```

<u>Task:4   Subquery and its type:</u>

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

Query:
-> select firstname, lastname, email, phone from customers where customerid not in (select customerid from orders);

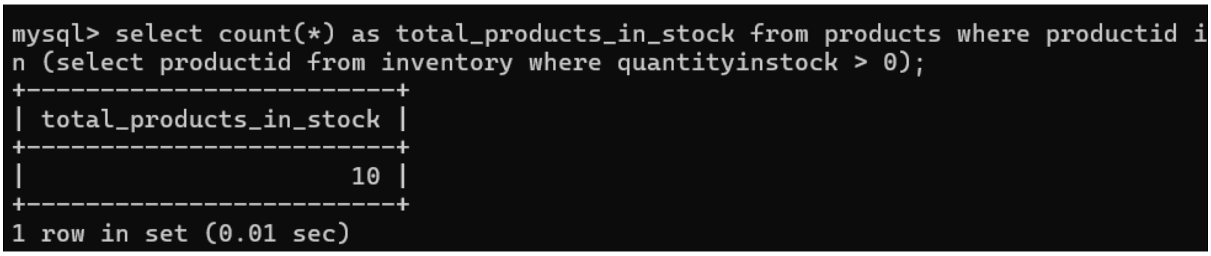


```
mysql> select firstname, lastname, email, phone from customers where customerid n
ot in (select customerid from orders);
+-----------+----------+------------------------+------------+
| firstname | lastname | email                  | phone      |
+-----------+----------+------------------------+------------+
| Syed      | Ahmed    | syedahmed@gmail.com    | 8429516443 |
| Lakshmi   | Naren    | lakshminaren@gmail.com | 9427518643 |
+-----------+----------+------------------------+------------+
2 rows in set (0.02 sec)
```

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

Query:

->select count(*) as total_products_in_stock from products where productid in (select productid from inventory where quantityinstock > 0);

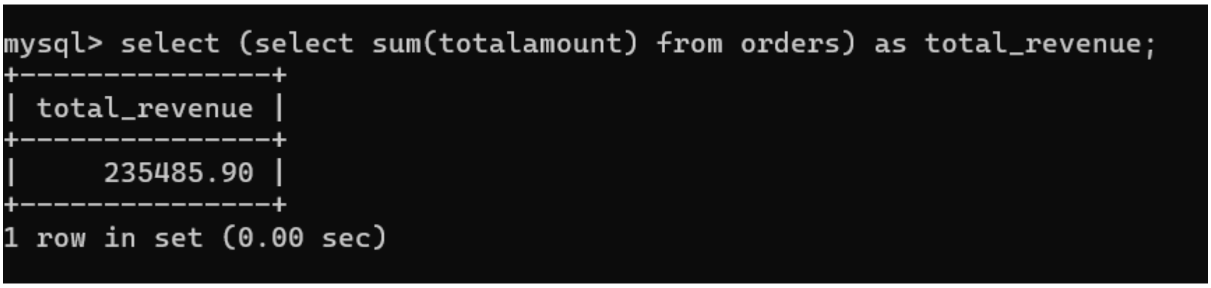


```
mysql> select count(*) as total_products_in_stock from products where productid i
n (select productid from inventory where quantityinstock > 0);
+-------------------------+
| total_products_in_stock |
+-------------------------+
|                      10 |
+-------------------------+
1 row in set (0.01 sec)
```

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

Query:
-> select (select sum(totalamount) from orders) as total_revenue;

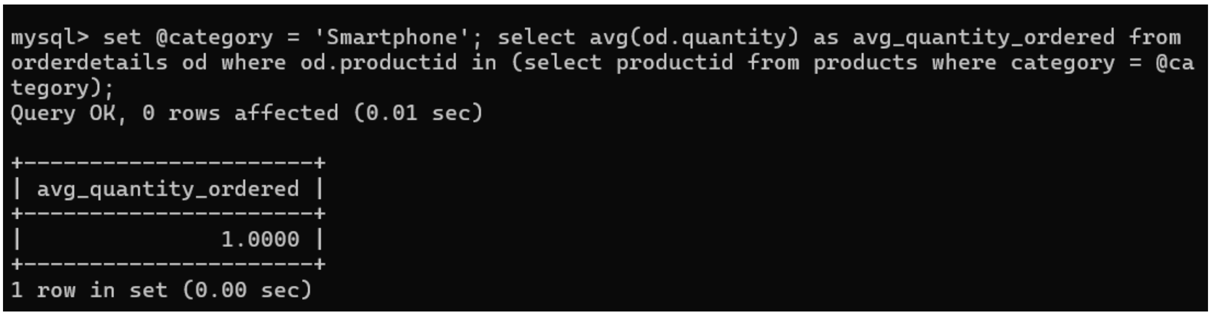
Using subqueries this query is done.



```
mysql> select (select sum(totalamount) from orders) as total_revenue;
+---------------+
| total_revenue |
+---------------+
|     235485.90 |
+---------------+
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.

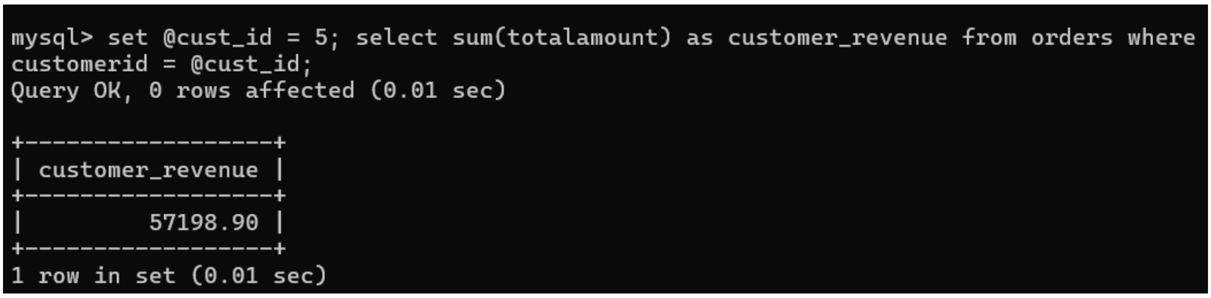
Query:
-> set @category = 'Smartphone'; select avg(od.quantity) as avg_quantity_ordered from orderdetails od where od.productid in (select productid from products where category = @category);



```
mysql> set @category = 'Smartphone'; select avg(od.quantity) as avg_quantity_ordered from
orderdetails od where od.productid in (select productid from products where category = @ca
tegory);
Query OK, 0 rows affected (0.01 sec)

+----------------------+
| avg_quantity_ordered |
+----------------------+
|               1.0000 |
+----------------------+
1 row in set (0.00 sec)
```

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

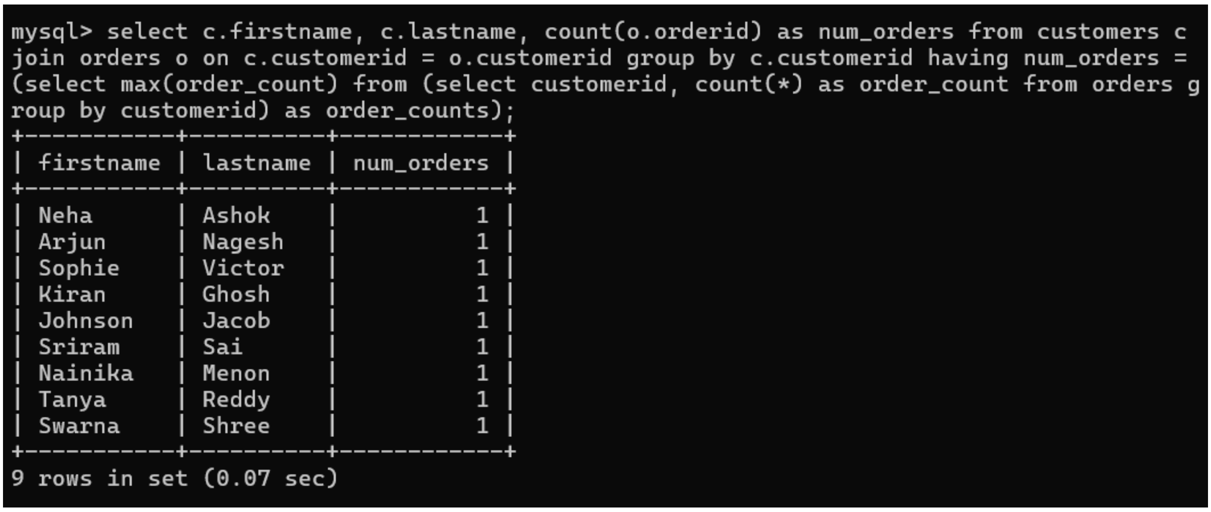
Query:
-> set @cust_id = 5; select sum(totalamount) as customer_revenue from orders where customerid = @cust_id;



```
mysql> set @cust_id = 5; select sum(totalamount) as customer_revenue from orders where
customerid = @cust_id;
Query OK, 0 rows affected (0.01 sec)

+------------------+
| customer_revenue |
+------------------+
|         57198.90 |
+------------------+
1 row in set (0.01 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.

Query:
->select c.firstname, c.lastname, count(o.orderid) as num_orders from customers c join orders o on c.customerid = o.customerid group by c.customerid having num_orders = (select max(order_count) from (select customerid, count(*) as order_count from orders group by customerid) as order_counts);

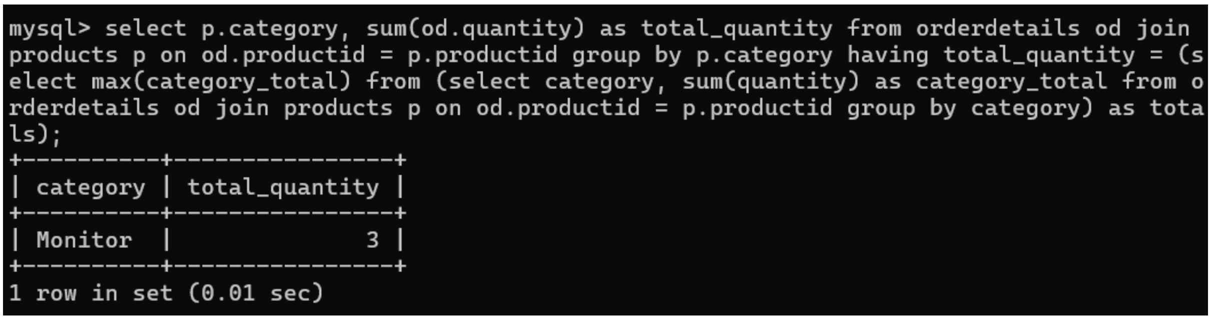


```
mysql> select c.firstname, c.lastname, count(o.orderid) as num_orders from customers c
join orders o on c.customerid = o.customerid group by c.customerid having num_orders =
(select max(order_count) from (select customerid, count(*) as order_count from orders g
roup by customerid) as order_counts);
+-----------+----------+------------+
| firstname | lastname | num_orders |
+-----------+----------+------------+
| Neha      | Ashok    |          1 |
| Arjun     | Nagesh   |          1 |
| Sophie    | Victor   |          1 |
| Kiran     | Ghosh    |          1 |
| Johnson   | Jacob    |          1 |
| Sriram    | Sai      |          1 |
| Nainika   | Menon    |          1 |
| Tanya     | Reddy    |          1 |
| Swarna    | Shree    |          1 |
+-----------+----------+------------+
9 rows in set (0.07 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.

Query:

-> select p.category, sum(od.quantity) as total_quantity from orderdetails od join products p on od.productid = p.productid group by p.category having total_quantity = (select max(category_total) from (select category, sum(quantity) as category_total from orderdetails od join products p on od.productid = p.productid group by category) as totals);

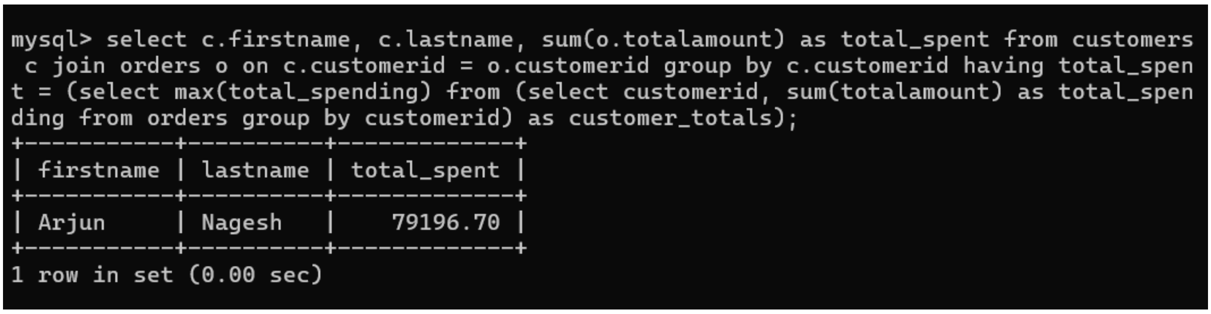


```
mysql> select p.category, sum(od.quantity) as total_quantity from orderdetails od join
products p on od.productid = p.productid group by p.category having total_quantity = (s
elect max(category_total) from (select category, sum(quantity) as category_total from o
rderdetails od join products p on od.productid = p.productid group by category) as tota
ls);
+----------+----------------+
| category | total_quantity |
+----------+----------------+
| Monitor  |              3 |
+----------+----------------+
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.

Query:

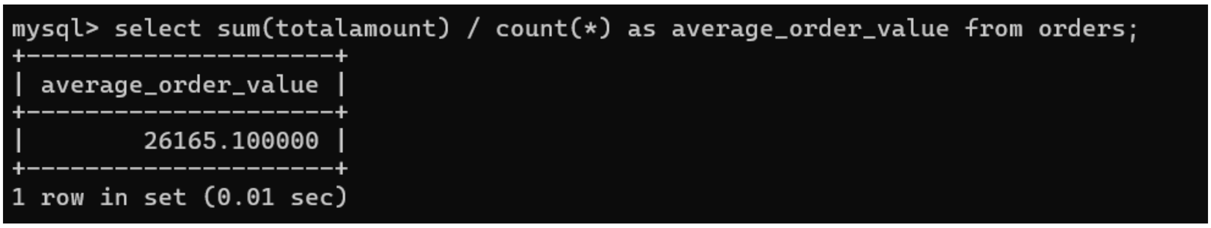
->select c.firstname, c.lastname, sum(o.totalamount) as total_spent from customers c join orders o on c.customerid = o.customerid group by c.customerid having total_spent = (select max(total_spending) from (select customerid, sum(totalamount) as total_spending from orders group by customerid) as customer_totals);



```
mysql> select c.firstname, c.lastname, sum(o.totalamount) as total_spent from customers
 c join orders o on c.customerid = o.customerid group by c.customerid having total_spen
t = (select max(total_spending) from (select customerid, sum(totalamount) as total_spen
ding from orders group by customerid) as customer_totals);
+-----------+----------+-------------+
| firstname | lastname | total_spent |
+-----------+----------+-------------+
| Arjun     | Nagesh   |    79196.70 |
+-----------+----------+-------------+
1 row in set (0.00 sec)
```

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

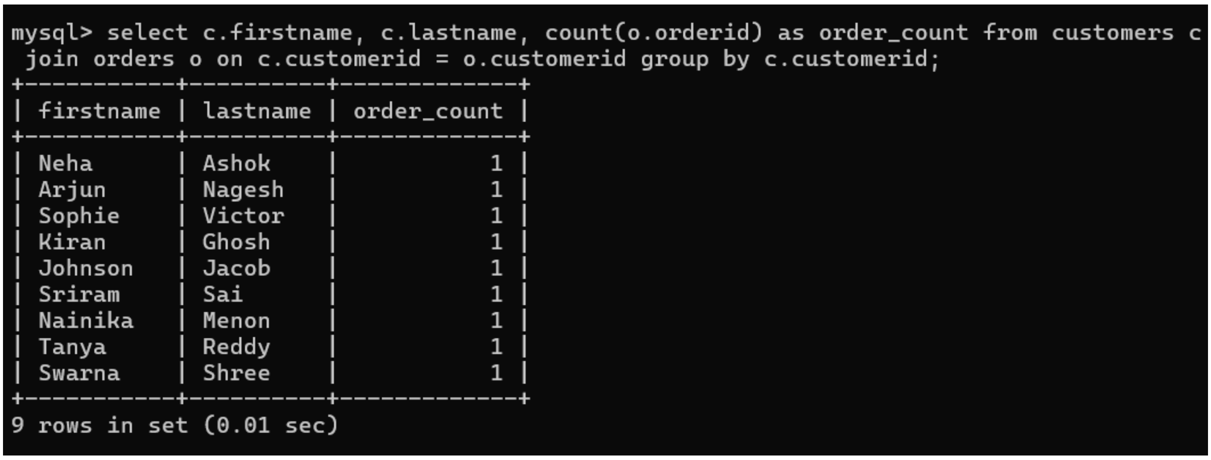
Query:
->select sum(totalamount) / count(*) as average_order_value from orders;



```
mysql> select sum(totalamount) / count(*) as average_order_value from orders;
+---------------------+
| average_order_value |
+---------------------+
|        26165.100000 |
+---------------------+
1 row in set (0.01 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.

Query:
-> select c.firstname, c.lastname, count(o.orderid) as order_count from customers c join orders o on c.customerid = o.customerid group by c.customerid;



```
mysql> select c.firstname, c.lastname, count(o.orderid) as order_count from customers c
 join orders o on c.customerid = o.customerid group by c.customerid;
+-----------+----------+-------------+
| firstname | lastname | order_count |
+-----------+----------+-------------+
| Neha      | Ashok    |           1 |
| Arjun     | Nagesh   |           1 |
| Sophie    | Victor   |           1 |
| Kiran     | Ghosh    |           1 |
| Johnson   | Jacob    |           1 |
| Sriram    | Sai      |           1 |
| Nainika   | Menon    |           1 |
| Tanya     | Reddy    |           1 |
| Swarna    | Shree    |           1 |
+-----------+----------+-------------+
9 rows in set (0.01 sec)
```