

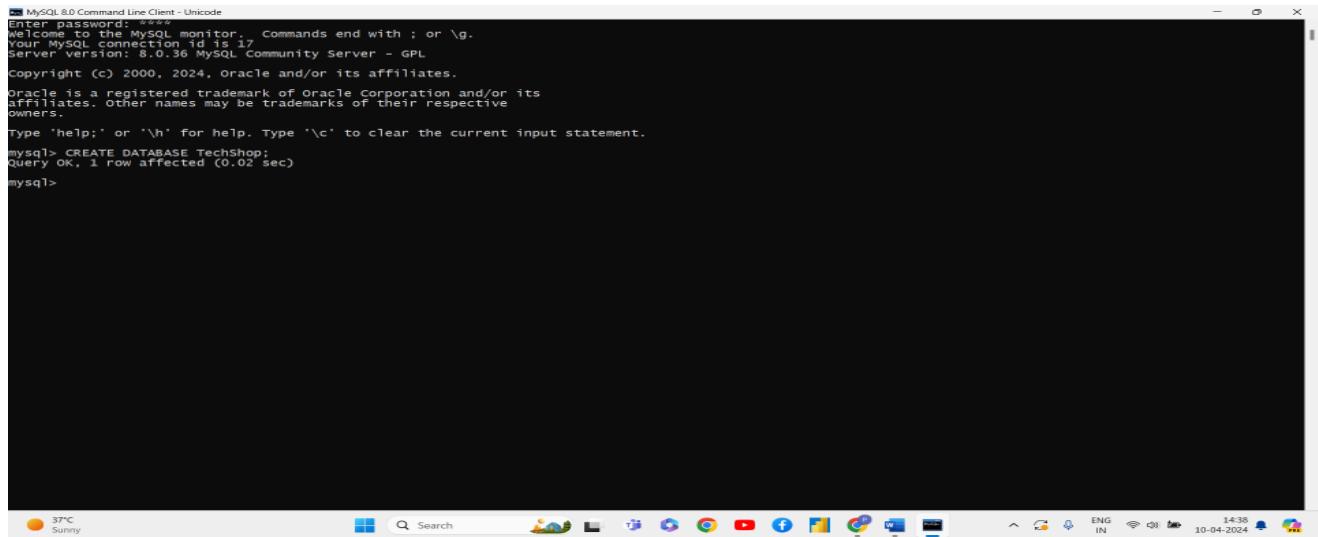
# **TECH SHOP – An Electronic gadgets shop**

"TechShop," sells electronic gadgets. TechShop maintains data related to their products, customers, and orders.

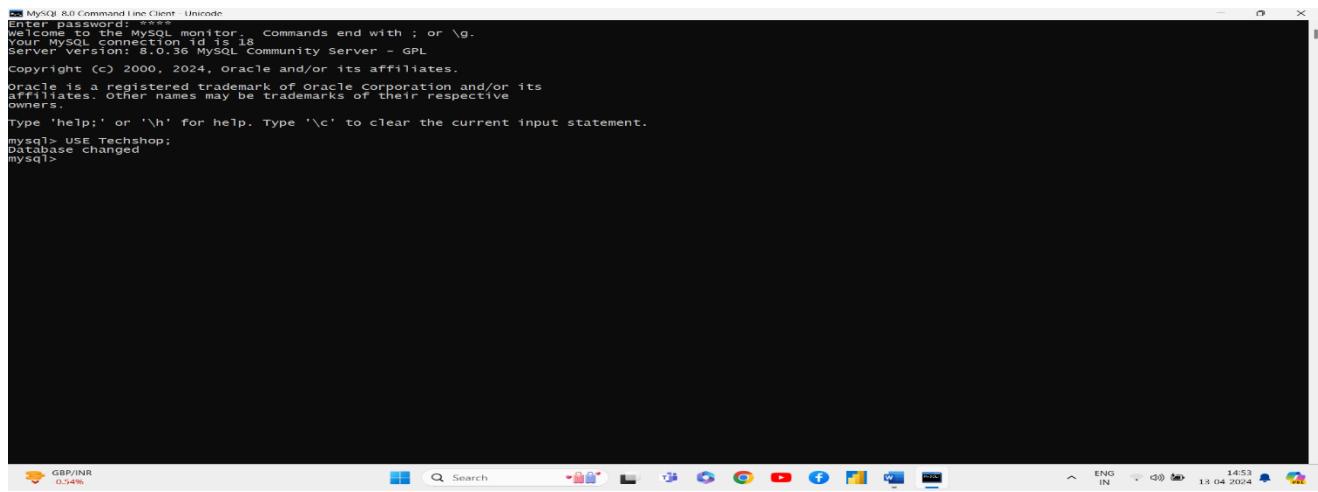
## Task:1. Database Design:

1. Create the database named "TechShop".

```
CREATE DATABASE TechShop;  
USE TechShop;
```



MySQL 8.0 Command Line Client - Unicode  
Enter password: \*\*\*\*  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 17  
Server version: 8.0.36 MySQL Community Server - GPL  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> CREATE DATABASE TechShop;  
Query OK, 1 row affected (0.02 sec)  
mysql>

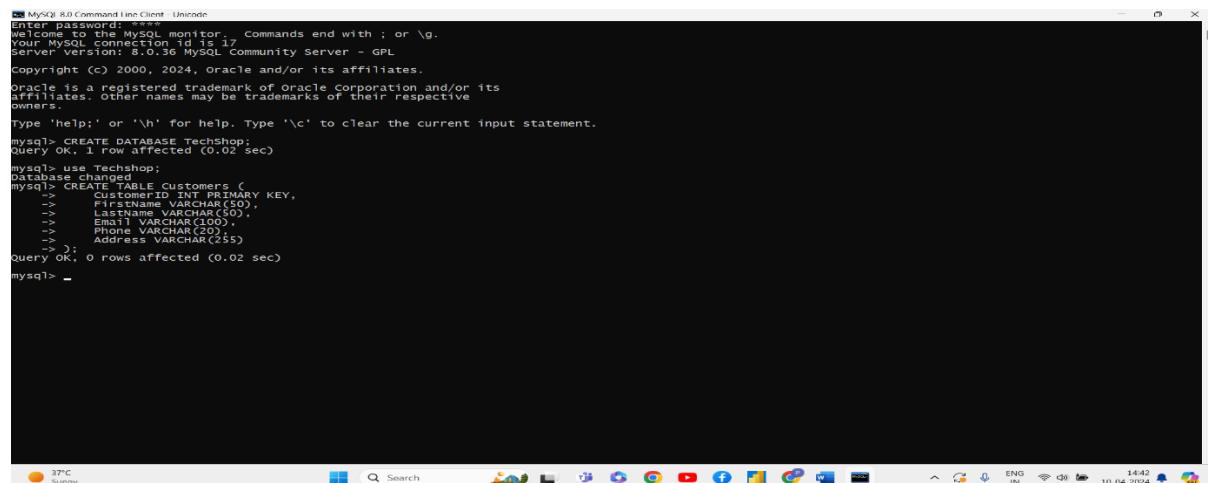


MySQL 8.0 Command Line Client - Unicode  
Enter password: \*\*\*\*  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 18  
Server version: 8.0.36 MySQL Community Server - GPL  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> USE TechShop;  
Database changed  
mysql>

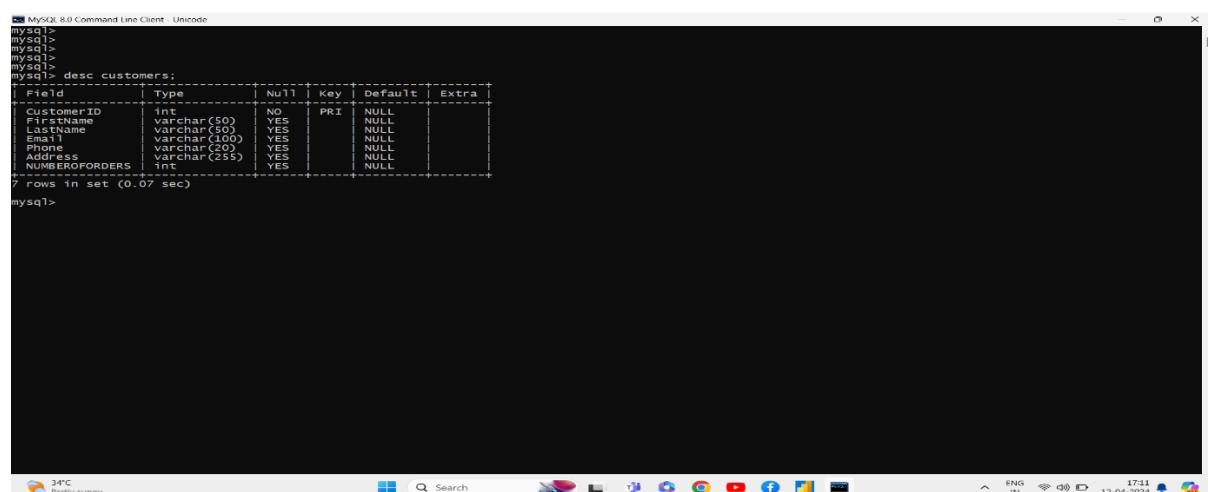
**2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.**

**2(a) Customers Table.**

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Phone VARCHAR(20),
    Address VARCHAR(255));
```



The screenshot shows the MySQL 8.0 Command Line Client window. The command `CREATE TABLE Customers (CustomerID INT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Email VARCHAR(100), Phone VARCHAR(20), Address VARCHAR(255));` is entered and executed. The output shows the table creation process, including the creation of the database `Techshop` and the table `Customers`. The table structure is defined with columns `CustomerID`, `FirstName`, `LastName`, `Email`, `Phone`, and `Address`.

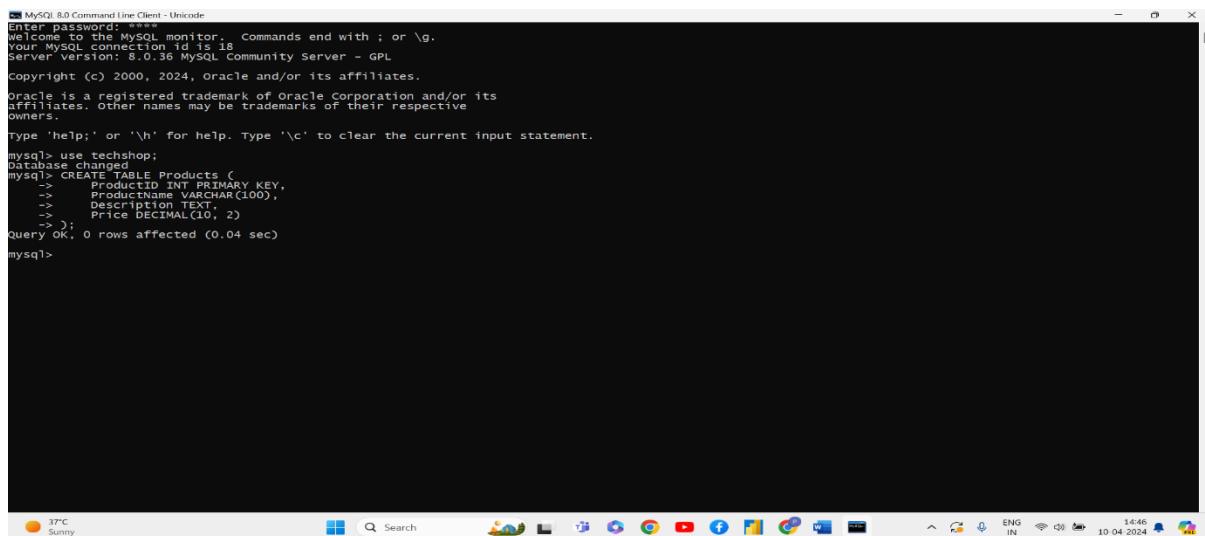


The screenshot shows the MySQL 8.0 Command Line Client window. The command `desc customers;` is entered to describe the `customers` table. The output displays the table structure with columns `CustomerID`, `FirstName`, `LastName`, `Email`, `Phone`, `Address`, and `NUMBEROFFORDERS`. The table has 7 rows in set (0.07 sec).

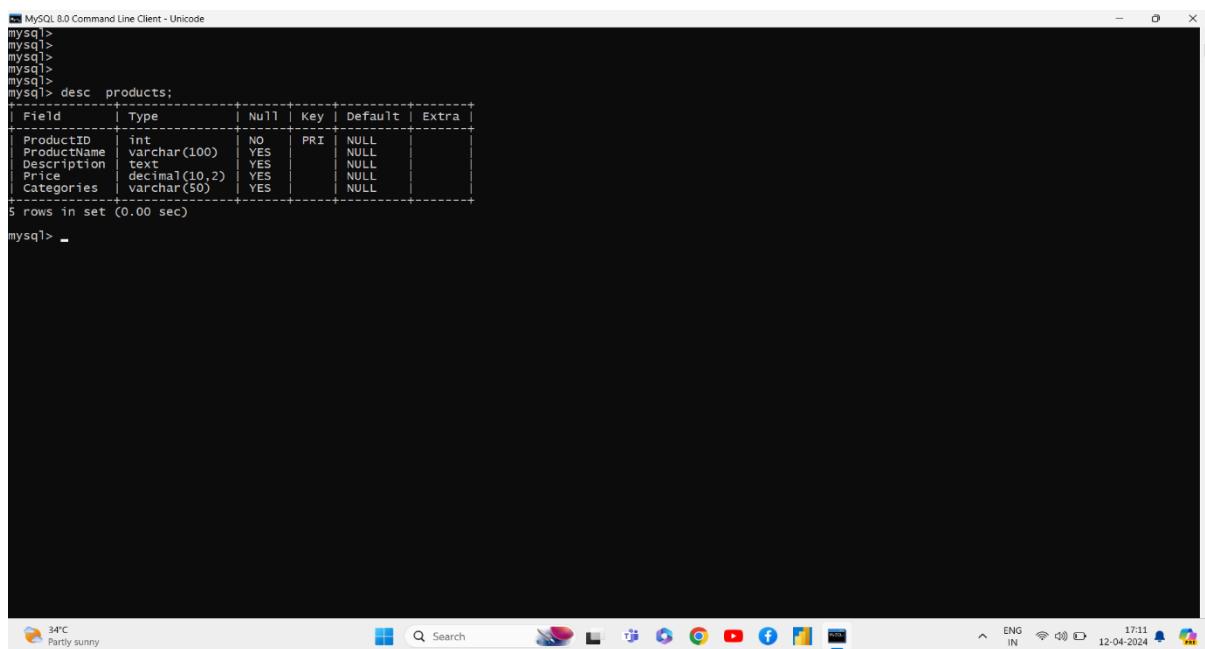
Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	
FirstName	varchar(50)	YES		NULL	
LastName	varchar(50)	YES		NULL	
Email	varchar(100)	YES		NULL	
Phone	varchar(20)	YES		NULL	
Address	varchar(255)	YES		NULL	
NUMBEROFFORDERS	int	YES		NULL	

## 2.(b).Products Table.

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Description TEXT,
    Price DECIMAL(10, 2));
```



This screenshot shows the MySQL 8.0 Command Line Client window. The command `CREATE TABLE Products (ProductID INT PRIMARY KEY, ProductName VARCHAR(100), Description TEXT, Price DECIMAL(10, 2));` has been entered and executed successfully, as indicated by the message "Query OK, 0 rows affected (0.04 sec)". The MySQL prompt is visible at the bottom of the window.

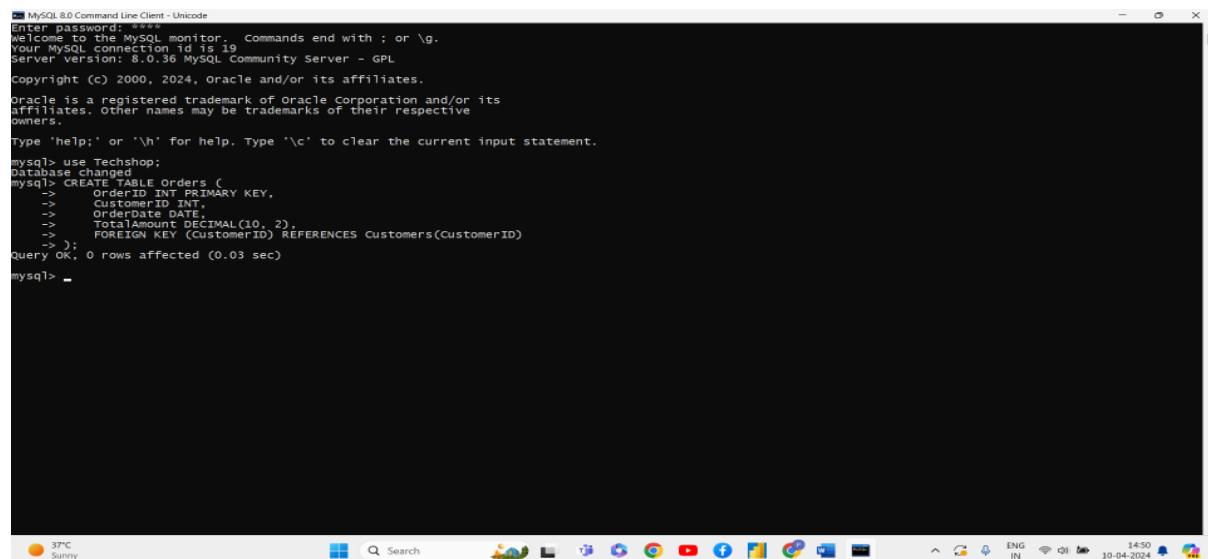


This screenshot shows the MySQL 8.0 Command Line Client window. The command `desc products;` has been entered and executed, displaying the structure of the `products` table. The output shows five columns: `ProductID`, `ProductName`, `Description`, `Price`, and `Categories`. The `ProductID` column is defined as an integer with a primary key constraint, while the other three columns are defined as strings. The `Price` column is defined as a decimal type with a scale of 2. The command `5 rows in set (0.00 sec)` is shown at the bottom, indicating the number of rows returned and the execution time.

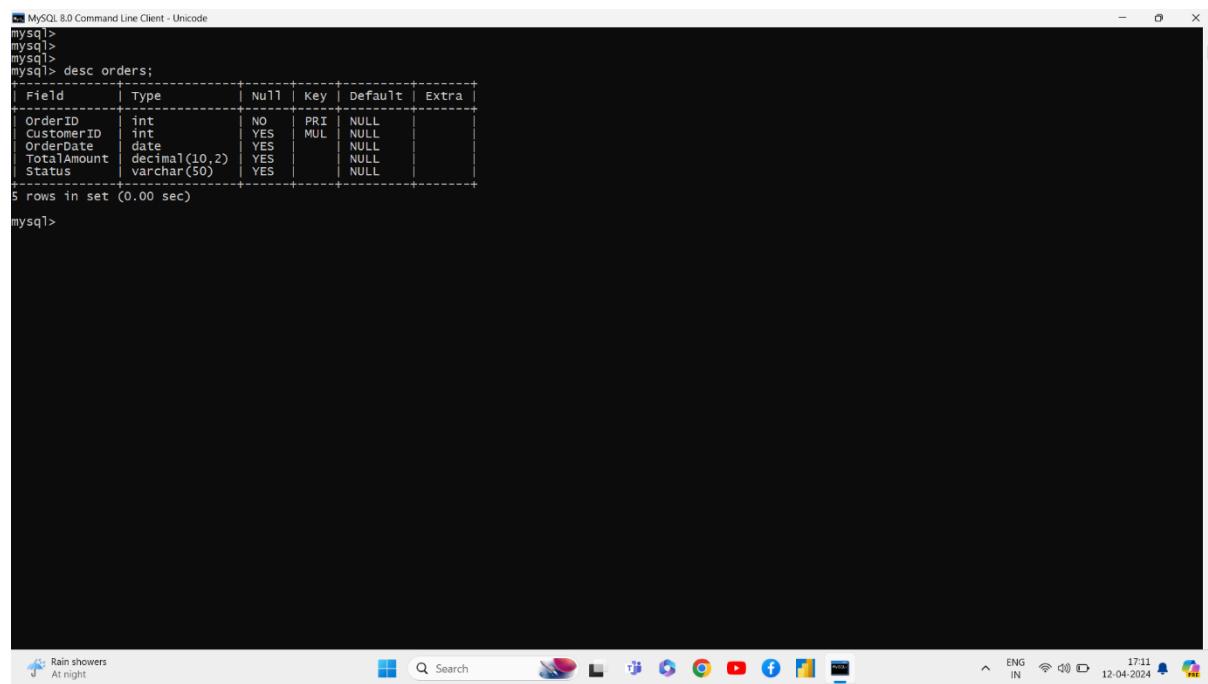
Field	Type	Null	Key	Default	Extra
ProductID	int	NO	PRI	NULL	
ProductName	varchar(100)	YES		NULL	
Description	text	YES		NULL	
Price	decimal(10,2)	YES		NULL	
Categories	varchar(50)	YES		NULL	

## 2.(c) Orders Table.

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));
```



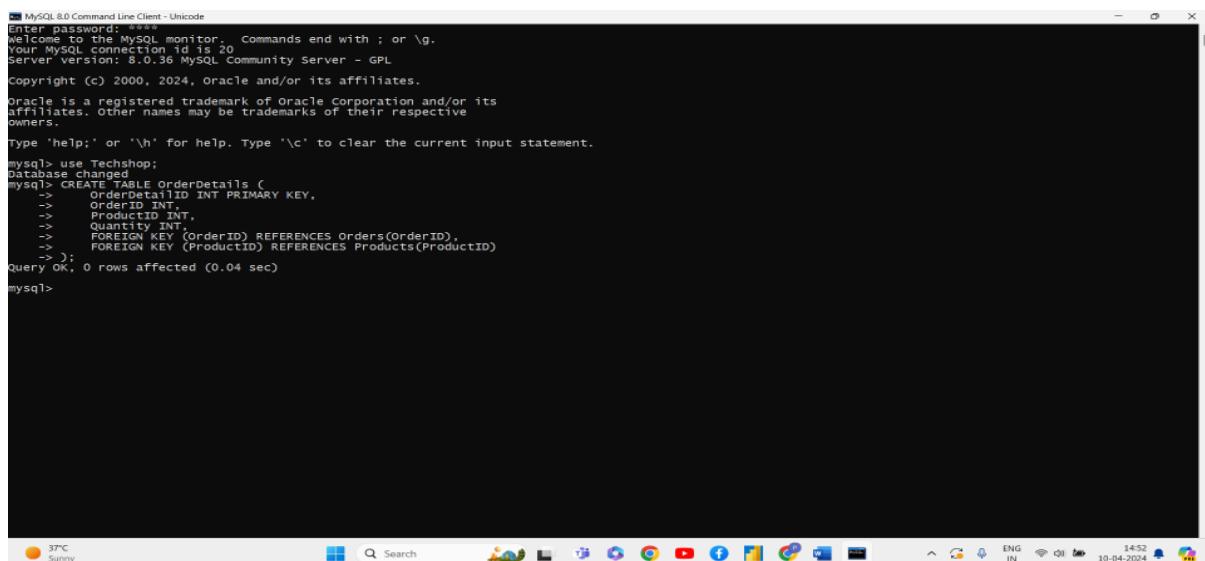
This screenshot shows the MySQL 8.0 Command Line Client window. The command `CREATE TABLE Orders` is being typed into the input field. The table structure includes columns for OrderID (INT, Primary Key), CustomerID (INT), OrderDate (DATE), and TotalAmount (DECIMAL(10, 2)). A foreign key constraint is defined for CustomerID, referencing the CustomerID column in the Customers table. The command is completed with a semicolon, and the response shows "Query OK, 0 rows affected (0.03 sec)". The MySQL monitor interface displays standard welcome messages and copyright information.



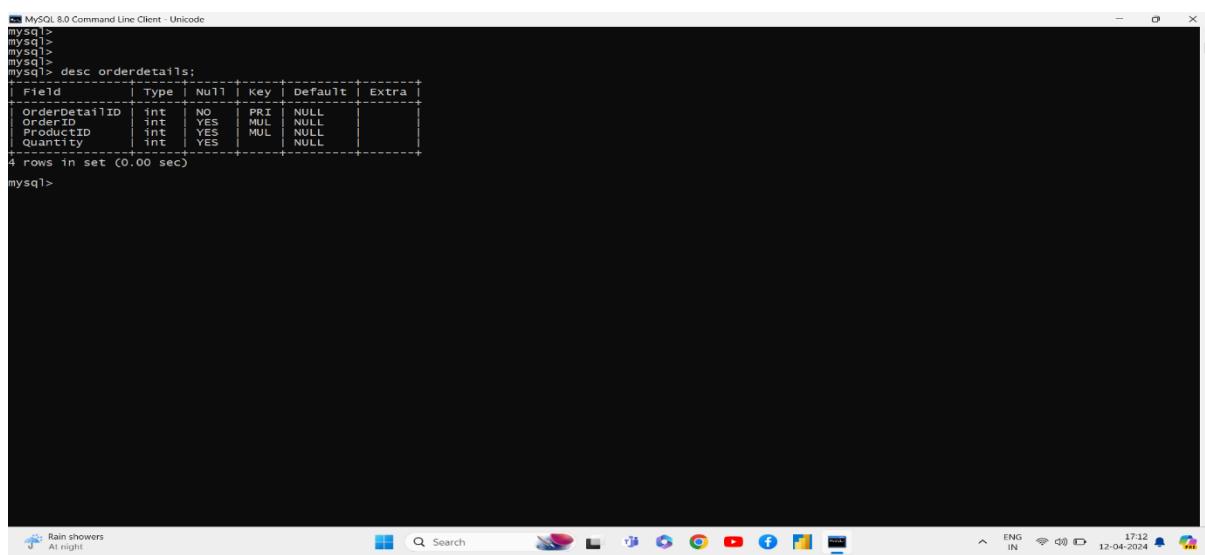
This screenshot shows the MySQL 8.0 Command Line Client window. The command `desc orders;` is being typed into the input field. The output displays the structure of the Orders table, listing five columns: OrderID (int, NO, PRI, NULL), CustomerID (int, YES, MUL, NULL), OrderDate (date, YES, NULL, NULL), TotalAmount (decimal(10,2), YES, NULL, NULL), and status (varchar(50), YES, NULL, NULL). The command is completed with a semicolon, and the response shows "5 rows in set (0.00 sec)". The MySQL monitor interface displays standard welcome messages and copyright information.

## 2.(d) Order details Table.

```
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));
```



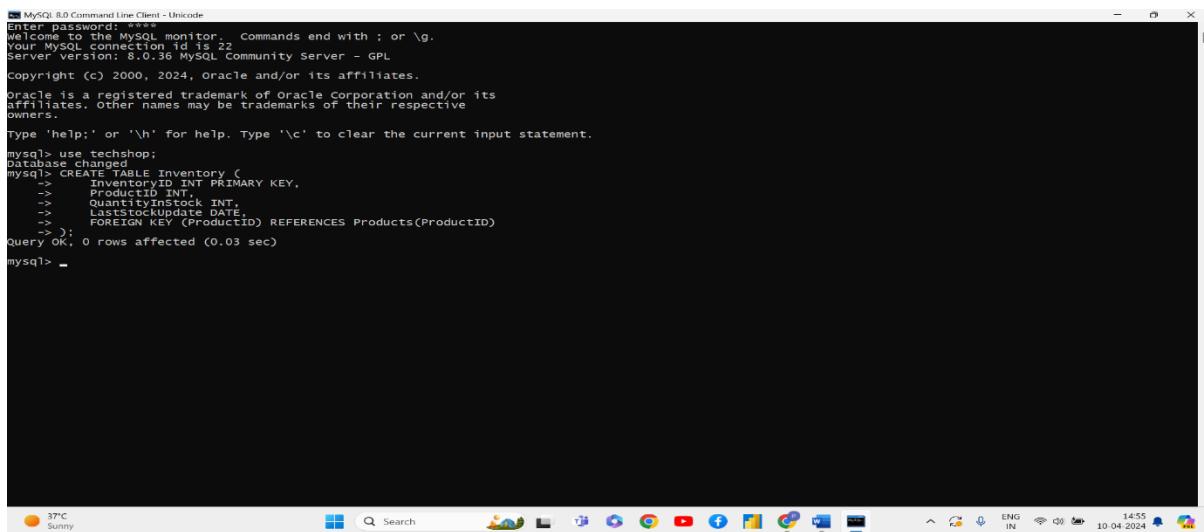
MySQL 8.0 Command Line Client - Unicode  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 20  
Server version: 8.0.36 MySQL Community Server - GPL  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> use Techshop;  
Database changed  
mysql> CREATE TABLE OrderDetails (  
 > OrderDetailID INT PRIMARY KEY,  
 > OrderID INT,  
 > ProductID INT,  
 > Quantity INT,  
 > FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
 > FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
 > );  
Query OK, 0 rows affected (0.04 sec)  
mysql>



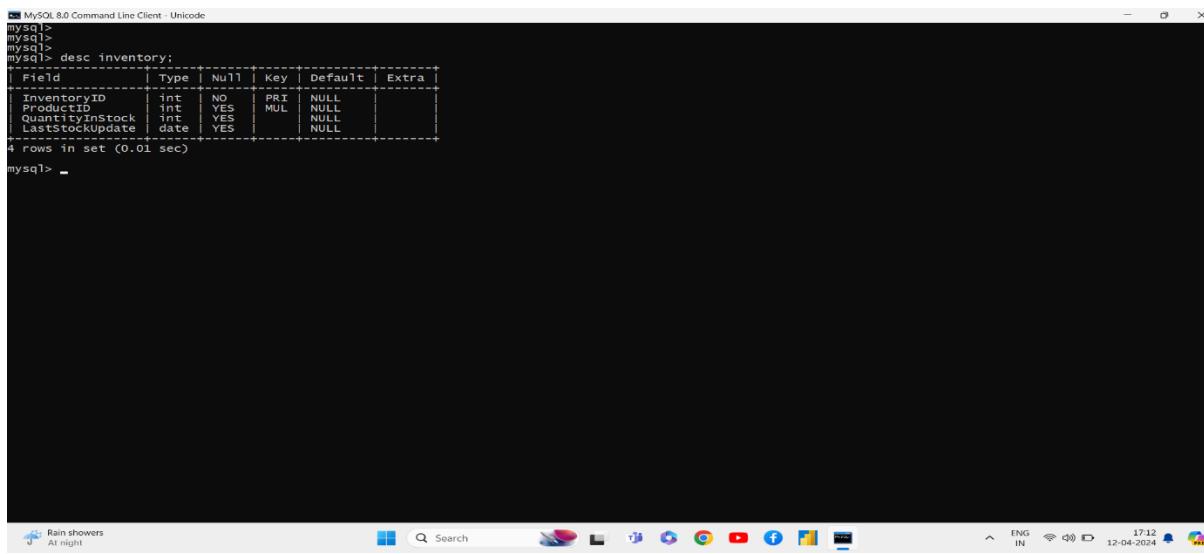
MySQL 8.0 Command Line Client - Unicode  
mysql>  
mysql>  
mysql>  
mysql> desc orderdetails;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
OrderDetailID	int	NO	PRI	NULL
OrderID	int	YES	MUL	NULL
ProductID	int	YES	MUL	NULL
quantity	int	YES	NULL	NULL
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
mysql>

## 2.(e) Inventory Table.

```
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate DATETIME,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID));
```



This screenshot shows the MySQL 8.0 Command Line Client window. The command `CREATE TABLE Inventory (InventoryID INT PRIMARY KEY, ProductID INT, QuantityInStock INT, LastStockUpdate DATETIME, FOREIGN KEY (ProductID) REFERENCES Products(ProductID));` is being entered into the command line. The output shows the table creation process, including the creation of the table and the addition of the foreign key constraint. The MySQL monitor displays standard copyright and help information at the top.

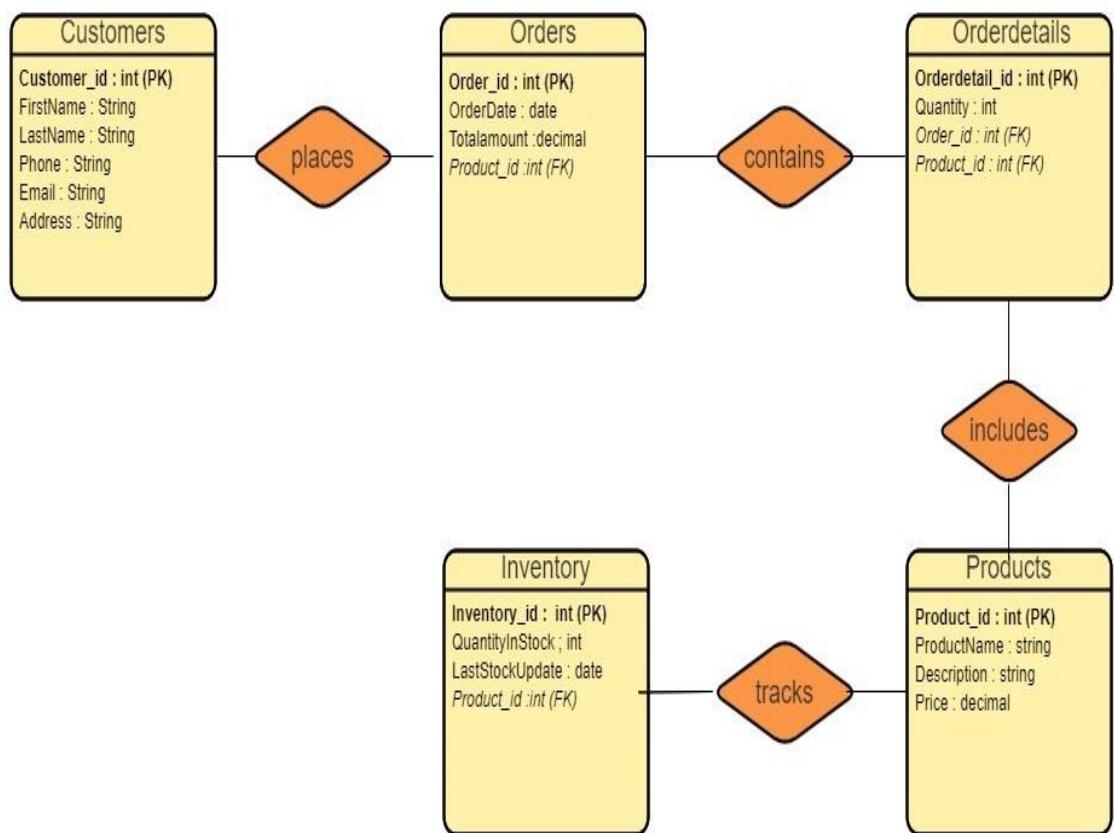


This screenshot shows the MySQL 8.0 Command Line Client window after the table has been created. The command `desc inventory;` is run to describe the structure of the `Inventory` table. The output displays the table structure with four columns: `InventoryID`, `ProductID`, `QuantityInStock`, and `LastStockUpdate`. The `InventoryID` column is defined as an integer primary key, while the other three columns are integers. The `LastStockUpdate` column is defined as a date type.

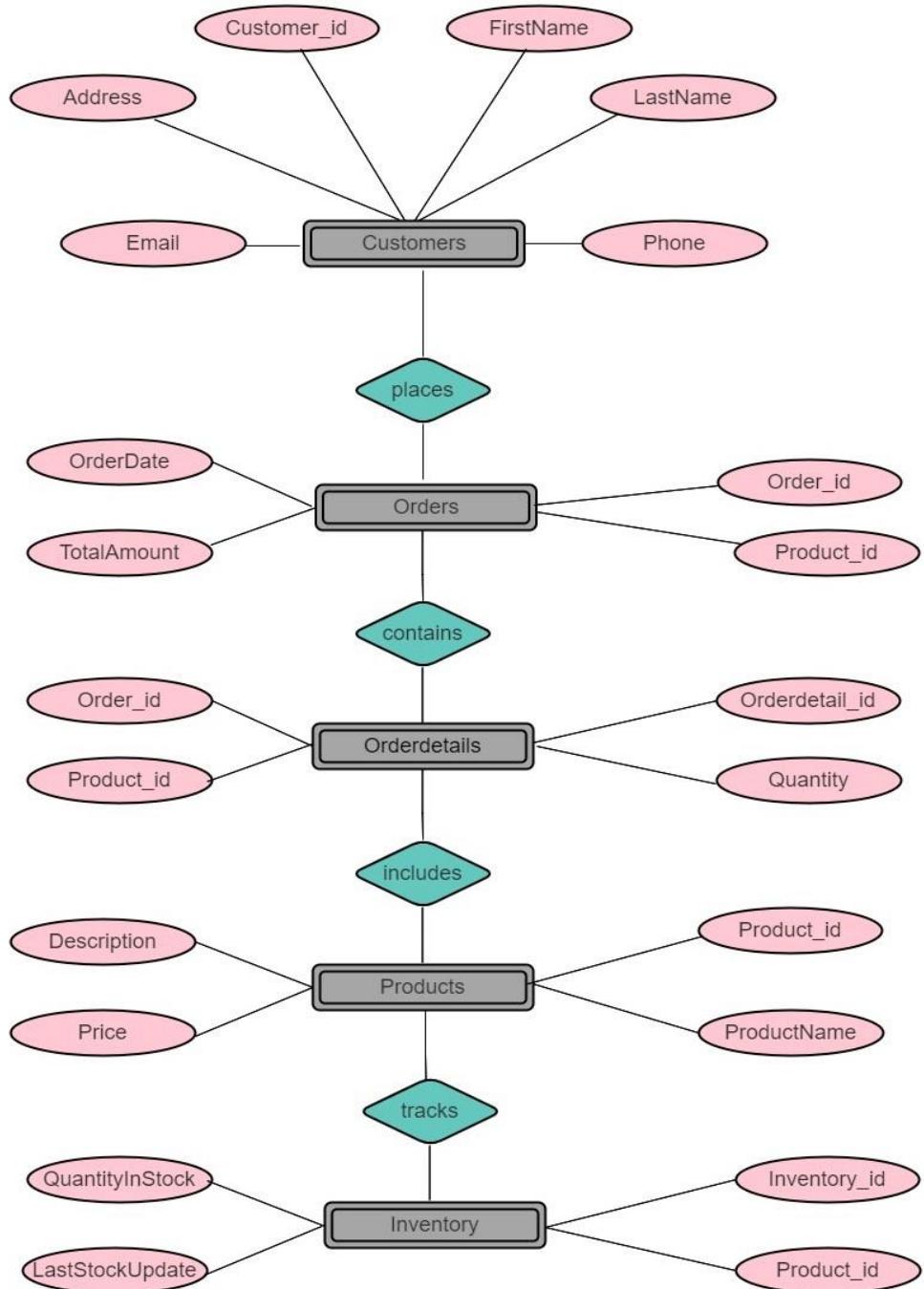
Field	Type	Null	Key	Default	Extra
InventoryID	int	NO	PRI	NULL	
ProductID	int	YES	MUL	NULL	
QuantityInStock	int	YES		NULL	
LastStockUpdate	date	YES		NULL	

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

TECHSHOP



## TECHSHOP



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

**4.(a) Primary keys for the tables :**

These keys are already defined in the database 'Techshop'.

TABLE NAME	PRIMARY KEY
Customers	Customer_id
Products	Product_id
Orders	Order_id
OrderDetails	OrderDetails_id
Inventory	Inventory_id

**4.(b) Foreign keys for referential constraints for the tables:**

FOREIGN KEY	ORIGINAL TABLE	REFERENCED TABLE
Customer_id	Customers	Orders
Order_id	Orders	OrderDetails
Product_id	Products	OrderDetails , Inventory

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

**5.(a)Customers**

```
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
```

```
VALUES
```

```
(1, 'Damon','Salvatore', 'damonsalvatore@gmail.com', '1234567890' '23 Mystic Falls')

(2,'stefan', 'Salvatore' stefansalvatore@gmail.com', '2134567890', '25 New York'),

(3, 'Elena','Gilbert', 'elenagilbert@gmail.com', '3124567890' '54 London'),

(4, 'Caroline', 'Forbes' 'carolineforbes@gmail.com' '7612345890' '90 San Francisco'),

(5, 'Bonny' , 'Bennet' 'bonnybennet@gmail.com' 4123567890', '07 Los Angeles'),

(6, 'Jeremy' , 'Gilbert' jeremygilbert@gmail.com' '5123467890', '55 Vatican'),

(7, 'Klaus','Michaelson 'klausmichaelson@gmail.com' '6123457890' '34 New York'),

(8, 'Elijah 'Michaelson elijahmichaelson@gmail.com '7123456890' '35 New York),

(9, 'Katherine' 'Pierce' 'katherinepierce@gmail.com' 8123456790', 89 San Francisco),

(10, 'Alaric', 'Saltzman', 'alaricsaltzman@gmail.com' '9123456780', '23 Vatican');
```

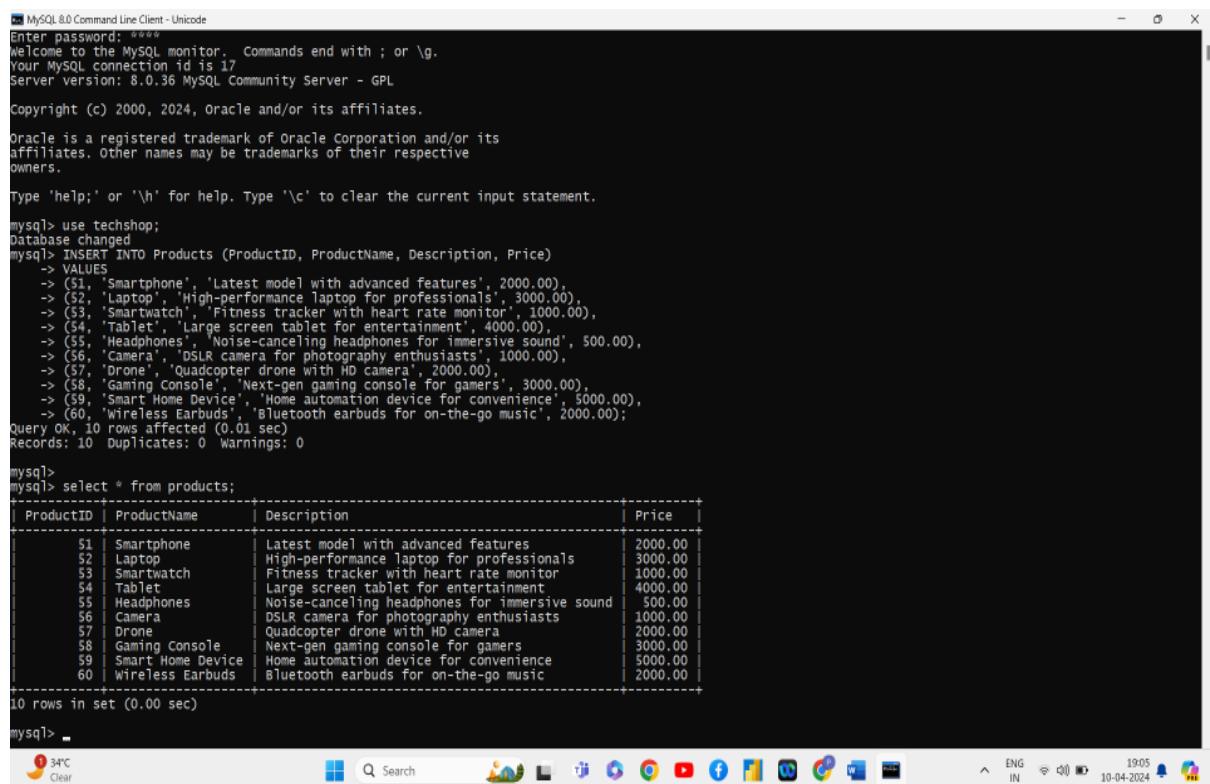
The screenshot shows a terminal window for MySQL 8.0 Command Line Client - Unicode. The user has switched to the 'techshop' database. An 'INSERT' statement is executed, inserting 10 rows into the 'Customers' table. The inserted data matches the values listed in the previous code block. After the insert, a 'select \* from customers;' query is run, displaying the 10 inserted rows. The table output is as follows:

CustomerID	FirstName	LastName	Email	Phone	Address
1	Damon	Salvatore	damonsalvatore@gmail.com	1234567890	23 Mystic Falls
2	Stefan	Salvatore	stefansalvatore@gmail.com	2134567890	25 New York
3	Elena	Gilbert	elenagilbert@gmail.com	3124567890	54 London
4	Caroline	Forbes	carolineforbes@gmail.com	7612345890	90 San Francisco
5	Bonny	Bennet	bonnybennet@gmail.com	4123567890	07 Los Angeles
6	Jeremy	Gilbert	jeremygilbert@gmail.com	5123467890	55 Vatican
7	Klaus	Michaelson	klausmichaelson@gmail.com	6123457890	34 New York
8	Elijah	Michaelson	elijahmichaelson@gmail.com	7123456890	35 New York
9	Katherine	Pierce	katherinepierce@gmail.com	8123456790	89 San Francisco
10	Alaric	Saltzman	alaricsaltzman@gmail.com	9123456780	23 Vatican

At the bottom of the terminal, system status information is visible, including battery level (0.54%), language (ENG IN), and date/time (10-04-2024 18:39).

## 5.(b) Products

```
INSERT INTO Products (ProductID, ProductName, Description, Price)
VALUES
(51,'Smartphone', 'Latest model with advanced features', 2000.00),
(52, 'Laptop' 'High-performance laptop for professionals', 3000.00),
(53, 'Smartwatch', 'Fitness tracker with heart rate monitor', 1000.00),
(54, 'Tablet' 'Large screen tablet for entertainment', 4000.00),
(55, 'Headphones', Noise-canceling headphones for immersive sound', 500.00),
(56, 'Camera' 'DSLR camera for photography enthusiasts', 1000.00),
(57, 'Drone' 'Quadcopter drone with HD camera', 2000.00),
(58, 'Gaming Console' 'Next-gen gaming console for gamers', 3000.00),
(59, 'Smart Home Device', 'Home automation device for convenience'5000.00),
(60, 'wireless Earbuds', 'Bluetooth earbuds for on-the-go music', 2000.00);
```



The screenshot shows the MySQL 8.0 Command Line Client interface. The command line shows the execution of an `INSERT` statement into the `Products` table, followed by a `SELECT * FROM products;` query which displays the inserted data in a tabular format.

```
MySQL 8.0 Command Line Client - Unicode
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

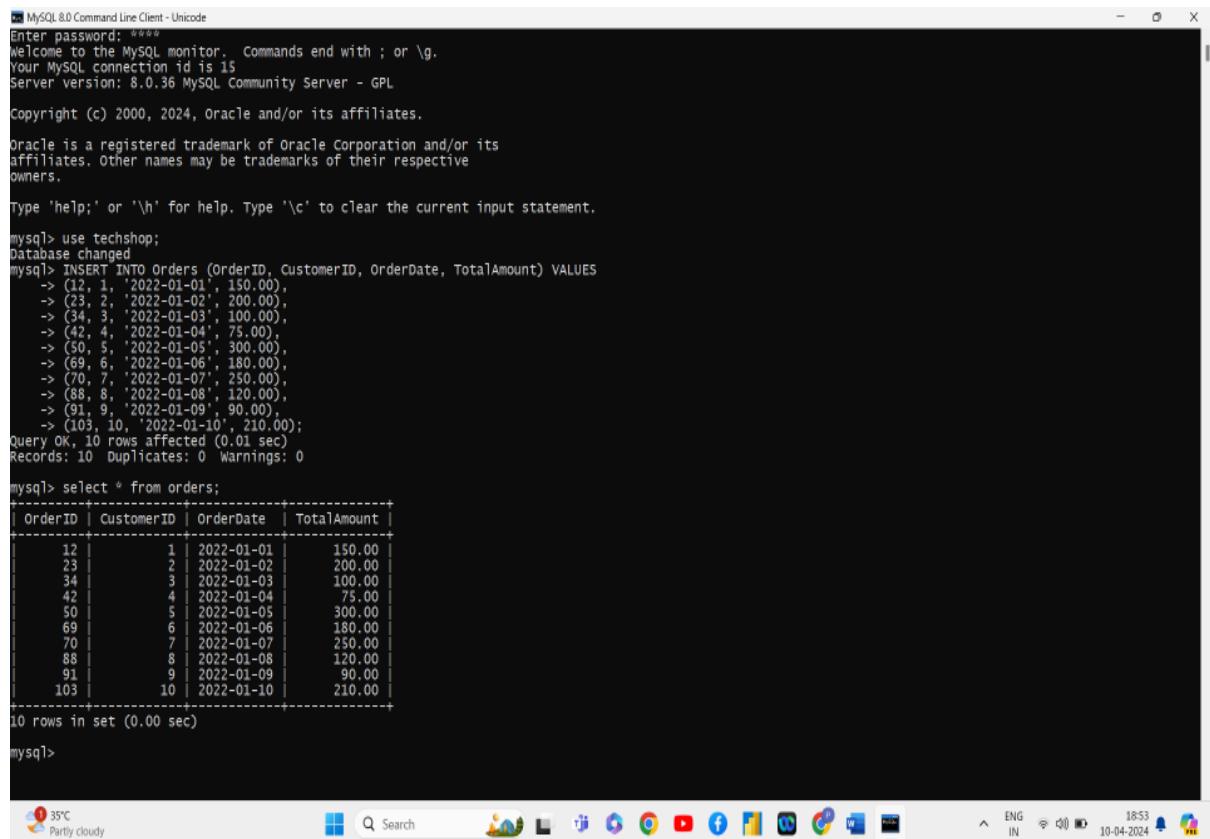
mysql> use techshop;
Database changed
mysql> INSERT INTO Products (ProductID, ProductName, Description, Price)
-> VALUES
-> ($1, 'Smartphone', 'Latest model with advanced features', 2000.00),
-> ($2, 'Laptop', 'High-performance laptop for professionals', 3000.00),
-> ($3, 'Smartwatch', 'Fitness tracker with heart rate monitor', 1000.00),
-> ($4, 'Tablet', 'Large screen tablet for entertainment', 4000.00),
-> ($5, 'Headphones', 'Noise-canceling headphones for immersive sound', 500.00),
-> ($6, 'Camera', 'DSLR camera for photography enthusiasts', 1000.00),
-> ($7, 'Drone', 'Quadcopter drone with HD camera', 2000.00),
-> ($8, 'Gaming Console', 'Next-gen gaming console for gamers', 3000.00),
-> ($9, 'Smart Home Device', 'Home automation device for convenience', 5000.00),
-> ($60, 'Wireless Earbuds', 'Bluetooth earbuds for on-the-go music', 2000.00);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
mysql> select * from products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| S1 | Smartphone | Latest model with advanced features | 2000.00 |
| S2 | Laptop | High-performance laptop for professionals | 3000.00 |
| S3 | Smartwatch | Fitness tracker with heart rate monitor | 1000.00 |
| S4 | Tablet | Large screen tablet for entertainment | 4000.00 |
| S5 | Headphones | Noise-canceling headphones for immersive sound | 500.00 |
| S6 | Camera | DSLR camera for photography enthusiasts | 1000.00 |
| S7 | Drone | Quadcopter drone with HD camera | 2000.00 |
| S8 | Gaming Console | Next-gen gaming console for gamers | 3000.00 |
| S9 | Smart Home Device | Home automation device for convenience | 5000.00 |
| S60 | Wireless Earbuds | Bluetooth earbuds for on-the-go music | 2000.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

### 5.(c) Orders

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
(12, 1, '2022-01-01' 150.00),
(23, 2, '2022-01-02', 200.00),
(34, 3, '2022-01-03', 100.00),
(42, 4, '2022-01-04', 75.00),
(50, 5, '2022-01-05' ,300.00) ,
(69, 6,'2022-01-06', 180.00),
(70, 7, '2022-01-07' ,250.00),
(88, 8, '2022-01-08', 120.00),
(91, 9, '2022-01-09' ,90.00),
(103, 10, '2022-01-10' ,210.00);
```



```
MySQL 8.0 Command Line Client - Unicode
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

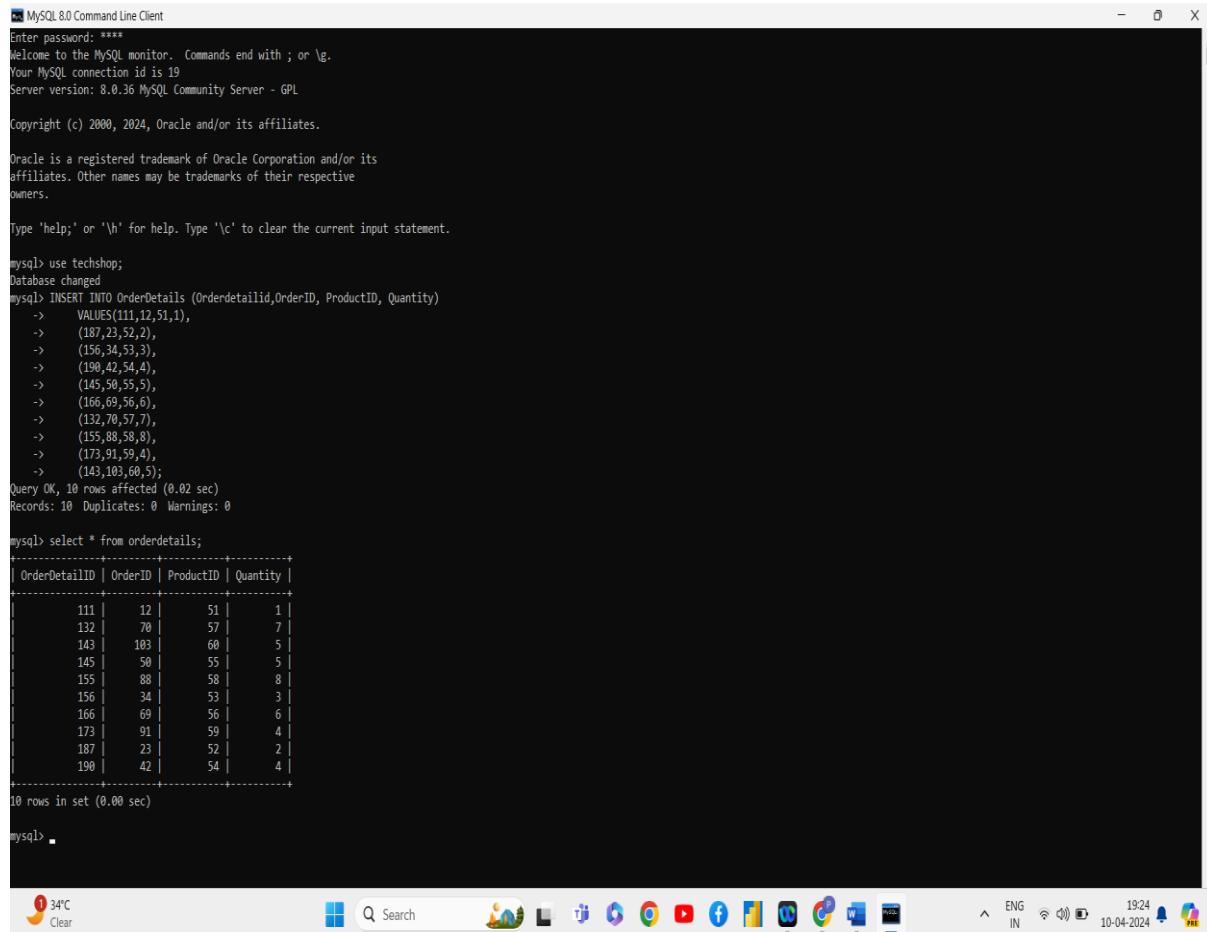
mysql> use techshop;
Database changed
mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES
-> (12, 1, '2022-01-01', 150.00),
-> (23, 2, '2022-01-02', 200.00),
-> (34, 3, '2022-01-03', 100.00),
-> (42, 4, '2022-01-04', 75.00),
-> (50, 5, '2022-01-05' ,300.00),
-> (69, 6, '2022-01-06', 180.00),
-> (70, 7, '2022-01-07' ,250.00),
-> (88, 8, '2022-01-08', 120.00),
-> (91, 9, '2022-01-09' ,90.00),
-> (103, 10, '2022-01-10' ,210.00);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
|      12 |          1 | 2022-01-01 |      150.00 |
|      23 |          2 | 2022-01-02 |      200.00 |
|      34 |          3 | 2022-01-03 |      100.00 |
|      42 |          4 | 2022-01-04 |       75.00 |
|      50 |          5 | 2022-01-05 |     300.00 |
|      69 |          6 | 2022-01-06 |      180.00 |
|      70 |          7 | 2022-01-07 |     250.00 |
|      88 |          8 | 2022-01-08 |      120.00 |
|      91 |          9 | 2022-01-09 |      90.00 |
|     103 |         10 | 2022-01-10 |     210.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

## 5.(d) OrderDetails

```
INSERT INTO OrderDetails (OrderDetailId, OrderID, ProductID, Quantity)
VALUES (111,12,51,1),
(187,23,52,2),
(156,34,53,3),
(190,42,54,4),
(145,50,55,5),
(166,69,56,6),
(132,70,57,7),
(155,88,58,8),
(173,91,59,4),
(143,103,60,5);
```



The screenshot shows a Windows taskbar at the bottom with various icons including Start, Search, File Explorer, Task View, Edge, YouTube, Facebook, and others. The system tray shows the date as 10-04-2024 and the time as 19:24.

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

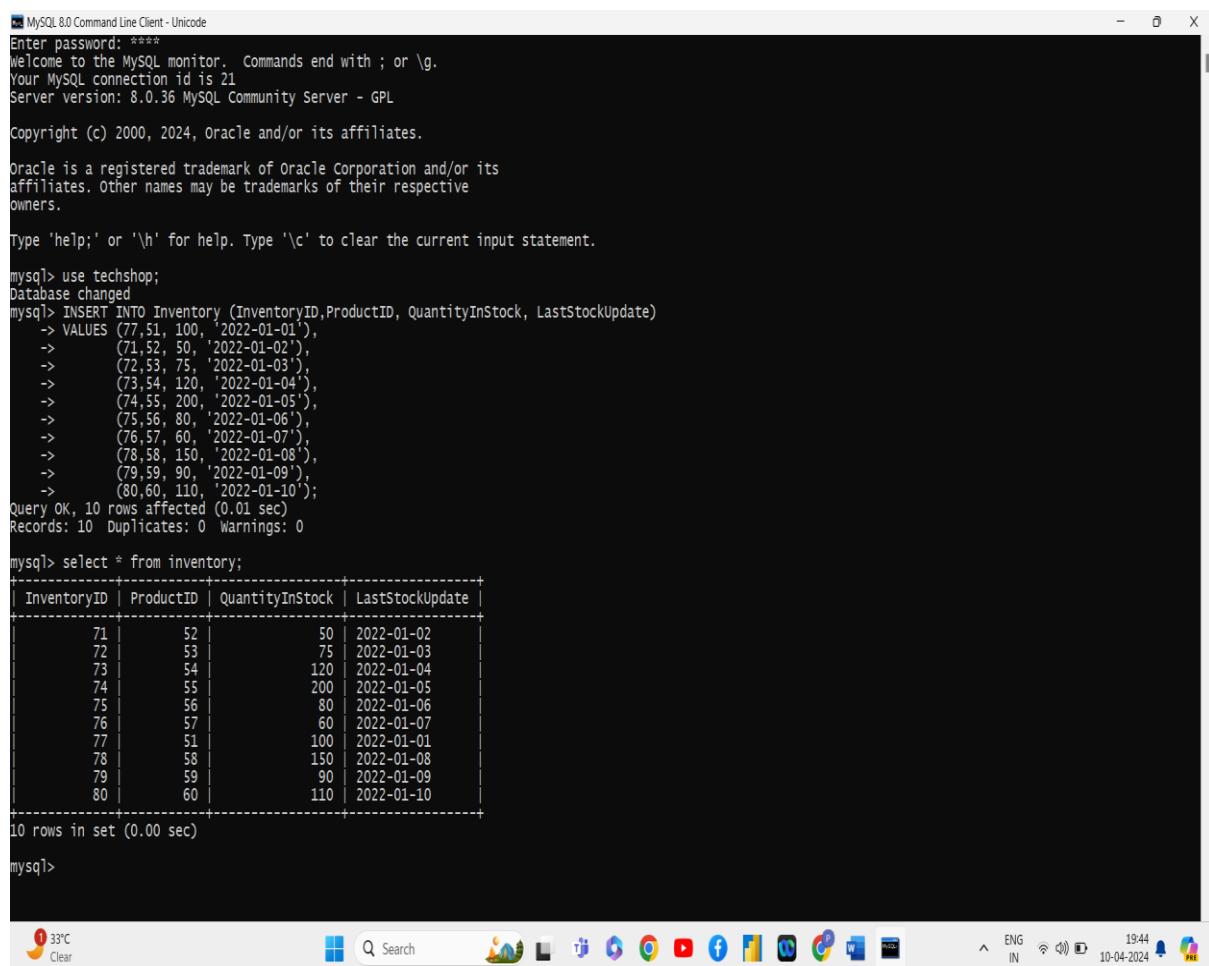
mysql> use techshop;
Database changed
mysql> INSERT INTO OrderDetails (OrderDetailId, OrderID, ProductID, Quantity)
->     VALUES(111,12,51,1),
->     (187,23,52,2),
->     (156,34,53,3),
->     (190,42,54,4),
->     (145,50,55,5),
->     (166,69,56,6),
->     (132,70,57,7),
->     (155,88,58,8),
->     (173,91,59,4),
->     (143,103,60,5);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from orderdetails;
+-----+-----+-----+-----+
| OrderDetailID | OrderID | ProductID | Quantity |
+-----+-----+-----+-----+
|      111 |      12 |       51 |       1 |
|      132 |      70 |       57 |       7 |
|      143 |     103 |       60 |       5 |
|      145 |      50 |       55 |       5 |
|      155 |      88 |       58 |       8 |
|      156 |      34 |       53 |       3 |
|      166 |      69 |       56 |       6 |
|      173 |      91 |       59 |       4 |
|      187 |      23 |       52 |       2 |
|      190 |      42 |       54 |       4 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

## 5.(e) Inventory

```
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)
VALUES
(77,51, 100, '2022-01-01'),
(71,52, 50, '2022-01-02'),
(72,53, 75, '2022-01-03'),
(73,54, 120, '2022-01-04') ,
(74,55, 200, '2022-01-05') ,
(75,56, 80, '2022-01-06'),
(76,57, 60, '2022-01-07'),
(78,58, 150, '2022-01-08'),
(79,59, 90, '2022-01-09'),
(80,60, 110, '2022-01-10');
```



The screenshot shows the MySQL 8.0 Command Line Client interface. The command line shows the execution of an `INSERT` statement into the `Inventory` table, followed by a `SELECT` statement to verify the data.

```
MySQL 8.0 Command Line Client - Unicode
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use techshop;
Database changed
mysql> INSERT INTO Inventory (InventoryID,ProductID, QuantityInStock, LastStockUpdate)
-> VALUES (77,51, 100, '2022-01-01'),
-> (71,52, 50, '2022-01-02'),
-> (72,53, 75, '2022-01-03'),
-> (73,54, 120, '2022-01-04'),
-> (74,55, 200, '2022-01-05'),
-> (75,56, 80, '2022-01-06'),
-> (76,57, 60, '2022-01-07'),
-> (78,58, 150, '2022-01-08'),
-> (79,59, 90, '2022-01-09'),
-> (80,60, 110, '2022-01-10');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from inventory;
+-----+-----+-----+-----+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-----+-----+-----+-----+
|      71 |       52 |         50 | 2022-01-02 |
|      72 |       53 |         75 | 2022-01-03 |
|      73 |       54 |        120 | 2022-01-04 |
|      74 |       55 |        200 | 2022-01-05 |
|      75 |       56 |         80 | 2022-01-06 |
|      76 |       57 |         60 | 2022-01-07 |
|      77 |       51 |        100 | 2022-01-01 |
|      78 |       58 |        150 | 2022-01-08 |
|      79 |       59 |         90 | 2022-01-09 |
|      80 |       60 |        110 | 2022-01-10 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

## 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;
```

The screenshot shows a terminal window for MySQL 8.0 Command Line Client - Unicode. The command `use techshop;` has been run, changing the database to `techshop`. The query `SELECT FirstName, LastName, Email FROM Customers;` is then executed, resulting in the following output:

FirstName	LastName	Email
Damon	Salvatore	damonsalvatore@gmail.com
Stefan	Salvatore	stefansalvatore@gmail.com
Elena	Gilbert	elenagilbert@gmail.com
Caroline	Forbes	carolineforbes@gmail.com
Bonny	Bennet	bonnybennet@gmail.com
Jeremy	Gilbert	jeremygilbert@gmail.com
Klaus	Michaelson	klausmichaelson@gmail.com
Elijah	Michaelson	elijahmichaelson@gmail.com
Katherine	Pierce	katherinepierce@gmail.com
Alaric	Saltzman	alaricsaltzman@gmail.com

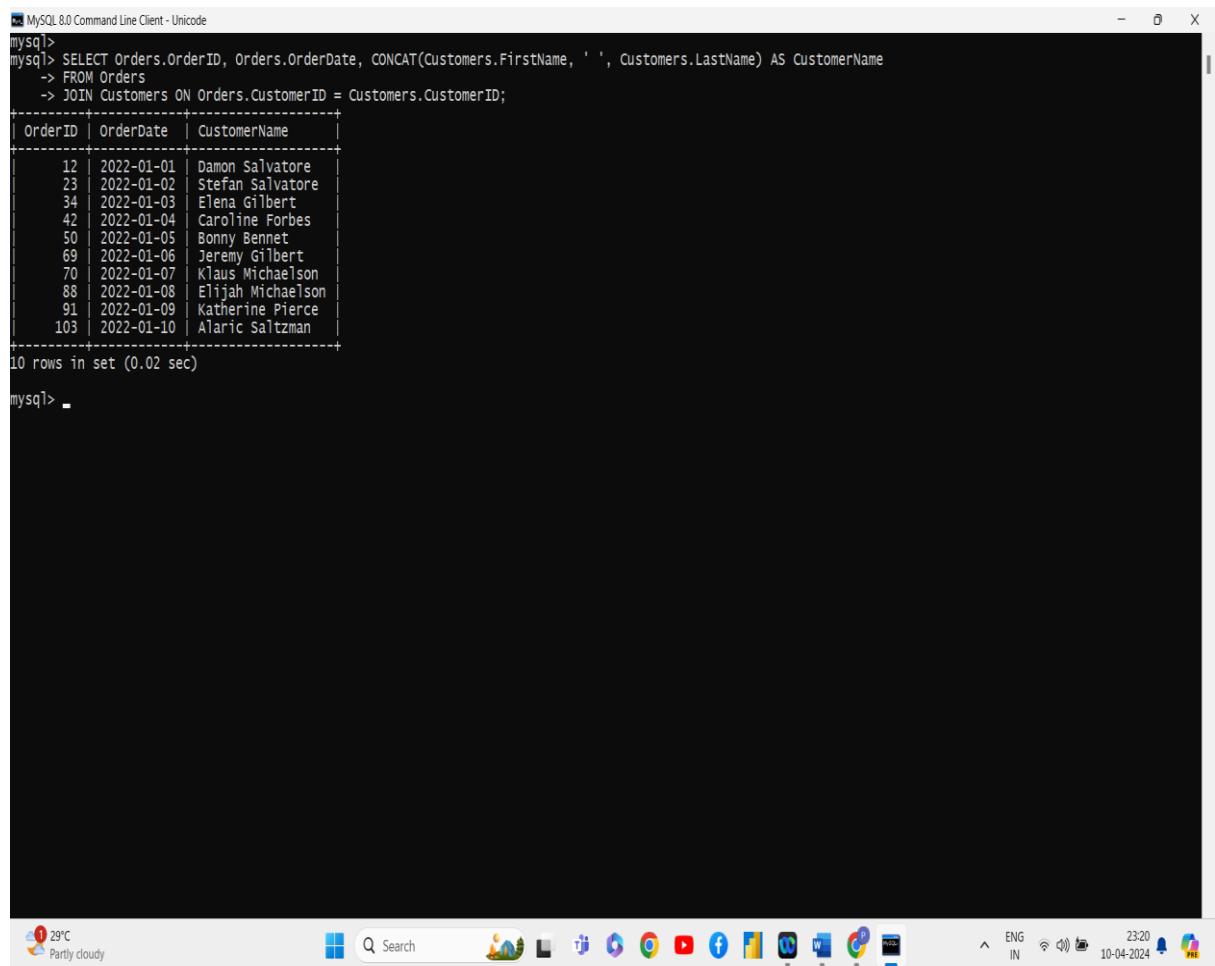
10 rows in set (0.01 sec)

mysql>

The system tray at the bottom of the window shows the date and time as 10-04-2024 23:14, along with icons for battery, signal, and other system status.

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

```
SELECT Orders.OrderID, Orders.OrderDate, CONCAT(Customers.FirstName, ' ',  
Customers.LastName) AS CustomerName  
FROM Orders  
JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```



The screenshot shows a terminal window for MySQL 8.0 Command Line Client - Unicode. The user has run the following SQL query:

```
mysql> SELECT Orders.OrderID, Orders.OrderDate, CONCAT(Customers.FirstName, ' ', Customers.LastName) AS CustomerName  
    -> FROM Orders  
    -> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

The result is a table with three columns: OrderID, OrderDate, and CustomerName. The data is as follows:

OrderID	OrderDate	CustomerName
12	2022-01-01	Damon Salvatore
23	2022-01-02	Stefan Salvatore
34	2022-01-03	Elena Gilbert
42	2022-01-04	Caroline Forbes
50	2022-01-05	Bonny Bennet
69	2022-01-06	Jeremy Gilbert
70	2022-01-07	Klaus Michaelson
88	2022-01-08	Elijah Michaelson
91	2022-01-09	Katherine Pierce
103	2022-01-10	Alaric Saltzman

10 rows in set (0.02 sec)

At the bottom of the terminal window, there is a status bar showing system information: 29°C, Partly cloudy, Search, Taskbar icons (including File Explorer, Edge, Google Chrome, YouTube, Facebook, Mail, Word, Excel, Powerpoint), ENG IN, 23:20, 10-04-2024, and a battery icon.

**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, phone, Address)
```

```
VALUES
```

```
(11, 'Rebekah', 'Michaelson','rebekahmichaelson@gmail.com', '9871234560','12 London');
```

The screenshot shows a terminal window for MySQL 8.0 Command Line Client - Unicode. The user has run an INSERT query to add a new customer record (CustomerID 11) with the name 'Rebekah Michaelson' and email 'rebekahmichaelson@gmail.com'. After the insertion, a SELECT \* query is run to display all records from the 'customers' table, which now includes the new row.

```
mysql>
mysql>
mysql> INSERT INTO CUSTOMERS(Customerid,FirstName,LastName,Email,phone,Address)
-> VALUES(11,'Rebekah','Michaelson','rebekahmichaelson@gmail.com','9871234560','12 London');
Query OK, 1 row affected (0.03 sec)

mysql> select* from customers;
+-----+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address |
+-----+-----+-----+-----+-----+-----+
| 1 | Damon | Salvatore | damonsalvatore@gmail.com | 1234567890 | 23 Mystic Falls |
| 2 | Stefan | Salvatore | stefansalvatore@gmail.com | 2134567890 | 25 New York |
| 3 | Elena | Gilbert | elenagilbert@gmail.com | 3124567890 | 54 London |
| 4 | Caroline | Forbes | carolineforbes@gmail.com | 7612345890 | 90 San Francisco |
| 5 | Bonny | Bennet | bonnybennet@gmail.com | 4123567890 | 07 Los Angeles |
| 6 | Jeremy | Gilbert | jeremygilbert@gmail.com | 5123467890 | 55 Vatican |
| 7 | Klaus | Michaelson | klausmichaelson@gmail.com | 6123457890 | 34 New York |
| 8 | Elijah | Michaelson | elijahmichaelson@gmail.com | 7123456890 | 35 New York |
| 9 | Katherine | Pierce | katherinepierce@gmail.com | 8123456790 | 89 San Francisco |
| 10 | Alaric | Saltzman | alaricsaltzman@gmail.com | 9123456780 | 23 Vatican |
| 11 | Rebekah | Michaelson | rebekahmichaelson@gmail.com | 9871234560 | 12 London |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

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

```
UPDATE Products
```

```
SET Price = Price* 1.10;
```

```
MySQL 8.0 Command Line Client - Unicode
mysql>
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 |
+-----+-----+-----+-----+
| 51 | Smartphone | Latest model with advanced features | 2200.00 |
| 52 | Laptop | High-performance laptop for professionals | 3300.00 |
| 53 | Smartwatch | Fitness tracker with heart rate monitor | 1100.00 |
| 54 | Tablet | Large screen tablet for entertainment | 4400.00 |
| 55 | Headphones | Noise-canceling headphones for immersive sound | 550.00 |
| 56 | Camera | DSLR camera for photography enthusiasts | 1100.00 |
| 57 | Drone | Quadcopter drone with HD camera | 2200.00 |
| 58 | Gaming Console | Next-gen gaming console for gamers | 3300.00 |
| 59 | Smart Home Device | Home automation device for convenience | 5500.00 |
| 60 | Wireless Earbuds | Bluetooth earbuds for on-the-go music | 2200.00 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql>
```



Search



ENG IN 23:33 10-04-2024

**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 ORDERS
```

```
WHERE Orderid=34;
```

The screenshot shows a terminal window for MySQL 8.0 Command Line Client - Unicode. The user has run the following commands:

```
mysql>
mysql>
mysql>
mysql> DELETE FROM OrderDetails
-> WHERE OrderID = (SELECT OrderID FROM Orders WHERE OrderID = 34);
Query OK, 1 row affected (0.02 sec)

mysql>
mysql> DELETE FROM Orders
-> WHERE OrderID = 34;
Query OK, 1 row affected (0.00 sec)

mysql> select * from orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
|      12 |         1 | 2022-01-01 |     150.00 |
|      23 |         2 | 2022-01-02 |     200.00 |
|      42 |         4 | 2022-01-04 |      75.00 |
|      50 |         5 | 2022-01-05 |     300.00 |
|      69 |         6 | 2022-01-06 |     180.00 |
|      70 |         7 | 2022-01-07 |     250.00 |
|      88 |         8 | 2022-01-08 |     120.00 |
|      91 |         9 | 2022-01-09 |      90.00 |
|     103 |        10 | 2022-01-10 |     210.00 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql> select * from orderdetails;
+-----+-----+-----+-----+
| OrderDetailID | OrderID | ProductID | Quantity |
+-----+-----+-----+-----+
|      111 |      12 |       51 |       1 |
|      132 |      70 |       57 |       7 |
|      143 |     103 |       60 |       5 |
|      145 |      50 |       55 |       5 |
|      155 |      88 |       58 |       8 |
|      166 |      69 |       56 |       6 |
|      173 |      91 |       59 |       4 |
|      187 |      23 |       52 |       2 |
|      190 |      42 |       54 |       4 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
```

The terminal also displays system status at the bottom, including weather (29°C Partly cloudy), search bar, and various application icons.

**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(od.Quantity * p.Price) AS AverageOrderValue  
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 GROUP BY c.CustomerID;
```

The screenshot shows the MySQL 8.0 Command Line Client window. The command line displays the following SQL query:

```
SELECT c.FirstName, c.LastName, AVG(od.Quantity * p.Price) AS AverageOrderValue  
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 GROUP BY c.CustomerID;
```

Below the command line, the results are displayed in a table:

FirstName	LastName	AverageOrderValue
Damon	Salvatore	2200.00000
Stefan	Salvatore	6600.00000
Caroline	Forbes	17600.00000
Bonny	Bennet	2750.00000
Jeremy	Gilbert	6600.00000
Klaus	Michaelson	15400.00000
Elijah	Michaelson	26400.00000
Katherine	Pierce	22000.00000
Alaric	Saltzman	11000.00000

At the bottom of the command line, it says "9 rows in set (0.05 sec)".

The taskbar at the bottom of the screen shows various icons for Windows applications like File Explorer, Edge, and others, along with system status indicators for weather (37°C), search, and connectivity.

**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 = 'piercekatherina@gmail.com', Address = '29 Paris'
```

```
WHERE CustomerID = 9;
```

```
MySQL 8.0 Command Line Client - Unicode
mysql>
mysql>
mysql> UPDATE Customers
-> SET Email = 'piercekatherina@gmail.com', Address = '29 Paris'
-> WHERE CustomerID = 9;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

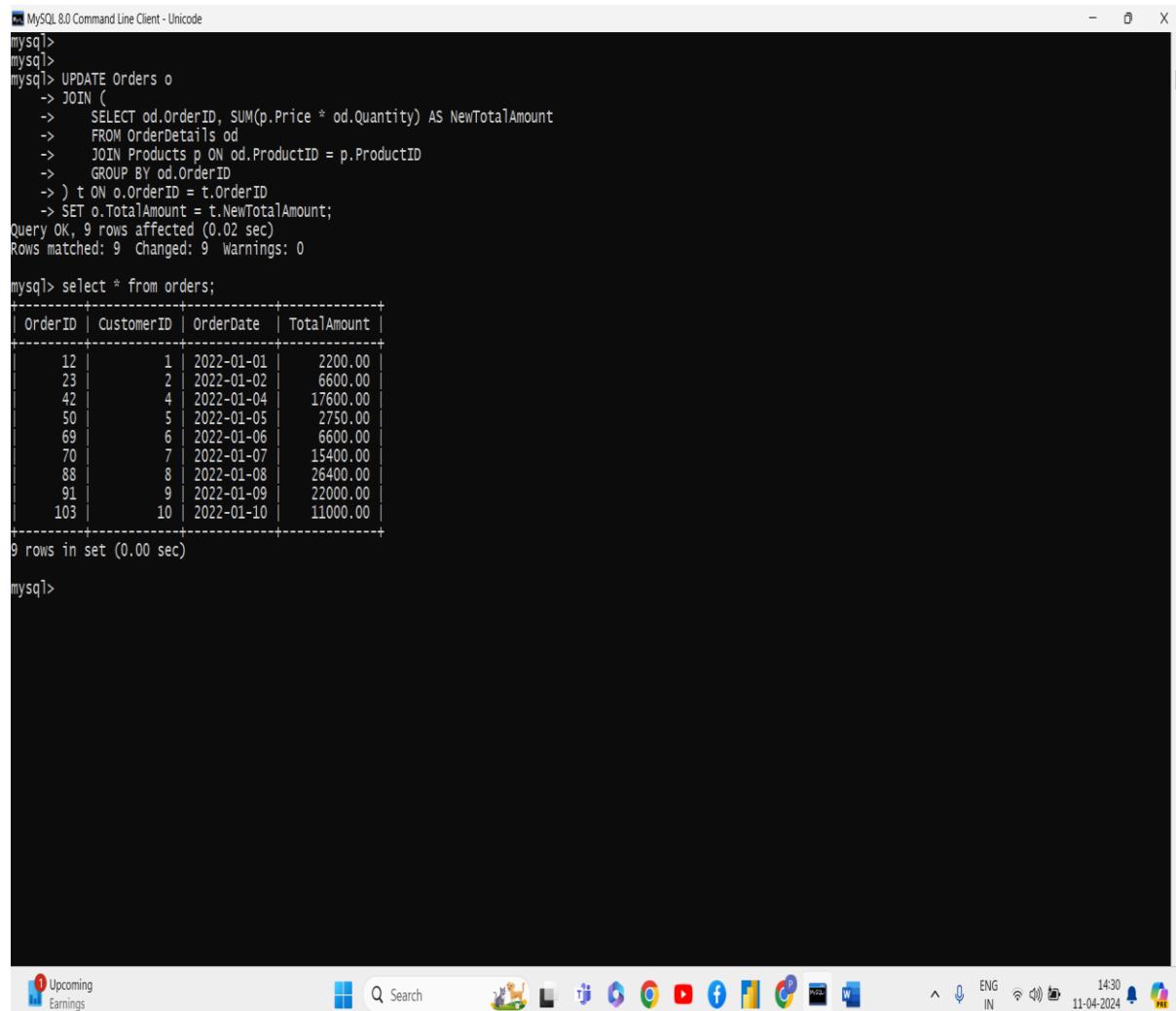
mysql> select * from customers;
+-----+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address |
+-----+-----+-----+-----+-----+-----+
| 1 | Damon | Salvatore | damonsalvatore@gmail.com | 1234567890 | 23 Mystic Falls |
| 2 | Stefan | Salvatore | stefansalvatore@gmail.com | 2134567890 | 25 New York |
| 3 | Elena | Gilbert | elenagilbert@gmail.com | 3124567890 | 54 London |
| 4 | Caroline | Forbes | carolineforbes@gmail.com | 7612345890 | 90 San Francisco |
| 5 | Bonny | Bennet | bonnybennet@gmail.com | 4123567890 | 07 Los Angeles |
| 6 | Jeremy | Gilbert | jeremygilbert@gmail.com | 5123467890 | 55 Vatican |
| 7 | Klaus | Michaelson | klausmichaelson@gmail.com | 6123457890 | 34 New York |
| 8 | Elijah | Michaelson | elijahmichaelson@gmail.com | 7123456890 | 35 New York |
| 9 | Katherine | Pierce | piercekatherina@gmail.com | 8123456790 | 29 Paris |
| 10 | Alaric | Saltzman | alaricsaltzman@gmail.com | 9123456780 | 23 Vatican |
| 11 | Rebekah | Michaelson | rebekahmichaelson@gmail.com | 9871234560 | 12 London |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> -
```



**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 o
JOIN (
  SELECT od.OrderID, SUM(p.Price * od.Quantity) AS NewTotalAmount
  FROM OrderDetails od
  JOIN Products p ON od.ProductID = p.ProductID
  GROUP BY od.OrderID
) t ON o.OrderID = t.OrderID
SET o.TotalAmount = t.NewTotalAmount;
```



The screenshot shows a terminal window for MySQL 8.0 Command Line Client - Unicode. The user has run an UPDATE query to calculate new total amounts for orders based on their details. After the update, a SELECT \* from orders command is run to verify the changes. The results show nine rows of order information, including OrderID, CustomerID, OrderDate, and the newly updated TotalAmount.

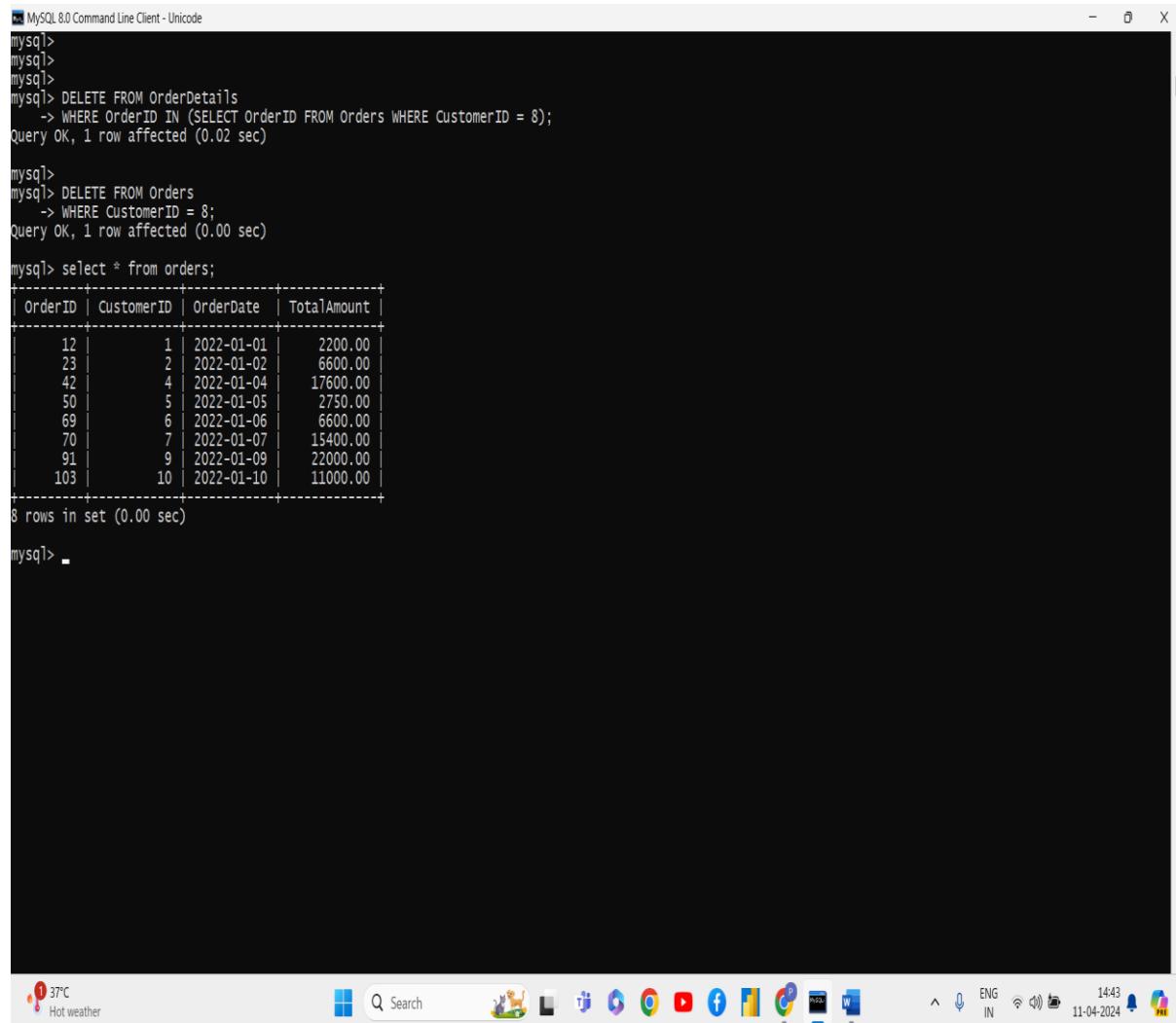
```
mysql>
mysql>
mysql> UPDATE Orders o
-> JOIN (
->   SELECT od.OrderID, SUM(p.Price * od.Quantity) AS NewTotalAmount
->   FROM OrderDetails od
->   JOIN Products p ON od.ProductID = p.ProductID
->   GROUP BY od.OrderID
-> ) t ON o.OrderID = t.OrderID
-> SET o.TotalAmount = t.NewTotalAmount;
Query OK, 9 rows affected (0.02 sec)
Rows matched: 9  Changed: 9  Warnings: 0

mysql> select * from orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
| 12 | 1 | 2022-01-01 | 2200.00 |
| 23 | 2 | 2022-01-02 | 6600.00 |
| 42 | 4 | 2022-01-04 | 17600.00 |
| 50 | 5 | 2022-01-05 | 2750.00 |
| 69 | 6 | 2022-01-06 | 6600.00 |
| 70 | 7 | 2022-01-07 | 15400.00 |
| 88 | 8 | 2022-01-08 | 26400.00 |
| 91 | 9 | 2022-01-09 | 22000.00 |
| 103 | 10 | 2022-01-10 | 11000.00 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
```

**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 Order ID FROM Orders WHERE CustomerID = 8);  
  
DELETE FROM Orders  
  
WHERE CustomerID = 8;
```



The screenshot shows a terminal window for MySQL 8.0 Command Line Client - Unicode. The user has run the following commands:

```
mysql>  
mysql>  
mysql>  
mysql> DELETE FROM OrderDetails  
-> WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = 8);  
Query OK, 1 row affected (0.02 sec)  
  
mysql>  
mysql> DELETE FROM Orders  
-> WHERE CustomerID = 8;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> select * from orders;  
+----+----+----+----+  
| OrderID | CustomerID | OrderDate | TotalAmount |  
+----+----+----+----+  
| 12 | 1 | 2022-01-01 | 2200.00 |  
| 23 | 2 | 2022-01-02 | 6600.00 |  
| 42 | 4 | 2022-01-04 | 17600.00 |  
| 50 | 5 | 2022-01-05 | 2750.00 |  
| 69 | 6 | 2022-01-06 | 6600.00 |  
| 70 | 7 | 2022-01-07 | 15400.00 |  
| 91 | 9 | 2022-01-09 | 22000.00 |  
| 103 | 10 | 2022-01-10 | 11000.00 |  
+----+----+----+----+  
8 rows in set (0.00 sec)  
  
mysql> -
```

The terminal shows the results of the SELECT query, which returns 8 rows. The system tray at the bottom indicates a weather icon for 'Hot weather' at 37°C, and the date and time as 11-04-2024 14:43.

**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 (Productid, ProductName, Description, Price)
VALUES (50, 'Smartphone X', 'Latest model with advanced features', 1000.00);
```

The screenshot shows a terminal window for MySQL 8.0 Command Line Client - Unicode. The user has run an INSERT query to add a new product (Smartphone X) with a price of 1000.00. After the insertion, a SELECT \* query is run to display all rows from the Products table, which now includes the new entry.

```
mysql>
mysql>
mysql>
mysql> INSERT INTO Products (Productid,ProductName, Description, Price)
-> VALUES (50,'Smartphone X', 'Latest model with advanced features', 1000.00);
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> select* from products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 50 | Smartphone X | Latest model with advanced features | 1000.00 |
| 51 | Smartphone | Latest model with advanced features | 2200.00 |
| 52 | Laptop | High-performance laptop for professionals | 3300.00 |
| 53 | Smartwatch | Fitness tracker with heart rate monitor | 1100.00 |
| 54 | Tablet | Large screen tablet for entertainment | 4400.00 |
| 55 | Headphones | Noise-canceling headphones for immersive sound | 550.00 |
| 56 | Camera | DSLR camera for photography enthusiasts | 1100.00 |
| 57 | Drone | Quadcopter drone with HD camera | 2200.00 |
| 58 | Gaming Console | Next-gen gaming console for gamers | 3300.00 |
| 59 | Smart Home Device | Home automation device for convenience | 5500.00 |
| 60 | Wireless Earbuds | Bluetooth earbuds for on-the-go music | 2200.00 |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)

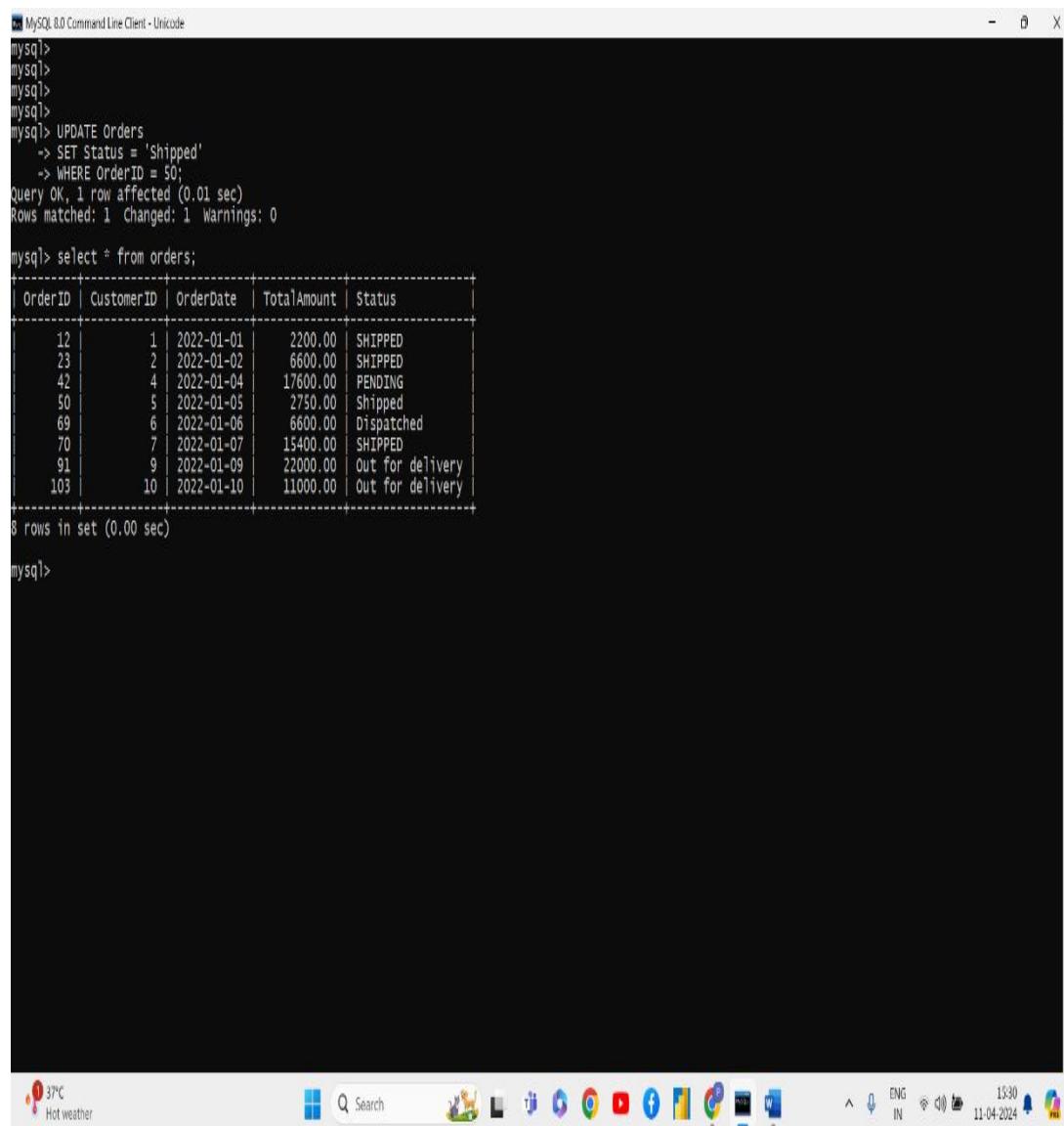
mysql>
```

**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.**

```
UPDATE ORDERS
```

```
SET Status = 'Shipped'
```

```
WHERE Orderid =50;
```



MySQL 8.0 Command Line Client - Unicode

```
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE Orders
-> SET Status = 'Shipped'
-> WHERE OrderID = 50;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from orders;
+-----+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount | Status |
+-----+-----+-----+-----+-----+
|      12 |         1 | 2022-01-01 |     2200.00 | SHIPPED |
|      23 |         2 | 2022-01-02 |     6600.00 | SHIPPED |
|      42 |         4 | 2022-01-04 |    17600.00 | PENDING  |
|      50 |         5 | 2022-01-05 |     2750.00 | Shipped  |
|      69 |         6 | 2022-01-06 |     6600.00 | Dispatched |
|      70 |         7 | 2022-01-07 |    15400.00 | SHIPPED  |
|      91 |         9 | 2022-01-09 |    22000.00 | Out for delivery |
|     103 |        10 | 2022-01-10 |    11000.00 | Out for delivery |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

The screenshot shows a terminal window for MySQL 8.0 Command Line Client. It displays an UPDATE query that changes the status of order ID 50 to 'Shipped'. After executing the query, it shows the results of a SELECT query that retrieves all columns from the 'orders' table. The table has 8 rows, with the 5th row being updated. The status column shows 'Shipped' for the 5th row instead of 'Pending'. The terminal window is running on a Windows operating system, as indicated by the taskbar icons at the bottom.

**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.**

```
UPDATE CUSTOMERS
SET NumberOfOrders =
SELECT COUNT(*) FROM ORDERS o
WHERE o.CustomerId = c.CustomerId);
```

```

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> UPDATE Customers c
-> SET NumberOfOrders =
->     (SELECT COUNT(*)
->      FROM Orders o
->     WHERE o.CustomerID = c.CustomerID
-> );
Query OK, 11 rows affected (0.02 sec)
Rows matched: 11  Changed: 11  Warnings: 0

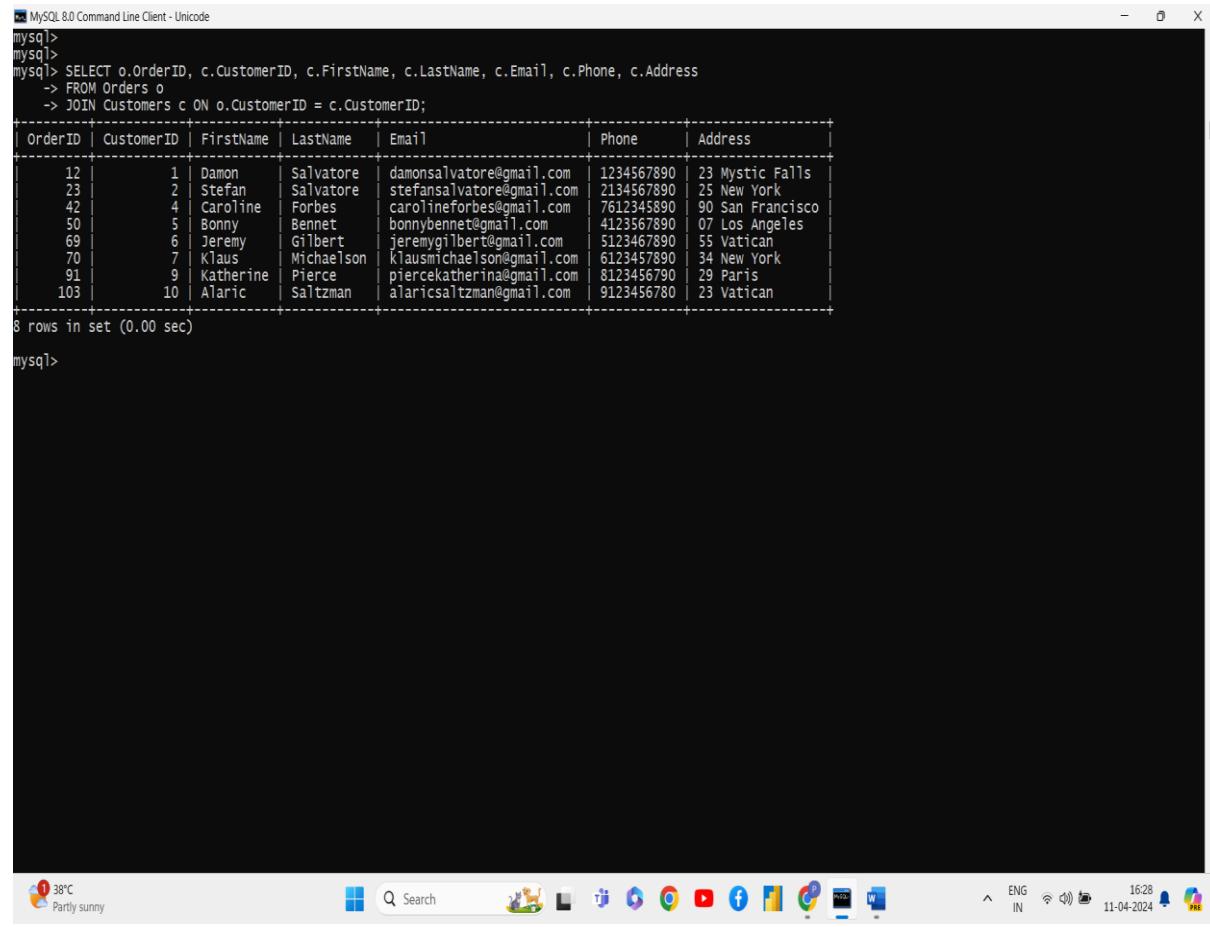
mysql> SELECT * FROM CUSTOMERS;
+-----+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | Phone | Address | NUMBEROFORDERS |
+-----+-----+-----+-----+-----+-----+
| 1 | Damon | Salvatore | damonsalvatore@gmail.com | 1234567890 | 23 Mystic Falls | 1 |
| 2 | Stefan | Salvatore | stefansalvatore@gmail.com | 2134567890 | 25 New York | 1 |
| 3 | Elena | Gilbert | elenagilbert@gmail.com | 3124567890 | 54 London | 0 |
| 4 | Caroline | Forbes | carolineforbes@gmail.com | 7612345890 | 90 San Francisco | 1 |
| 5 | Bonny | Bennet | bonnybennet@gmail.com | 41234567890 | 07 Los Angeles | 1 |
| 6 | Jeremy | Gilbert | jeremygilbert@gmail.com | 5123467890 | 55 Vatican | 1 |
| 7 | Klaus | Michaelson | klausmichaelson@gmail.com | 6123457890 | 34 New York | 1 |
| 8 | Elijah | Michaelson | elijahmichaelson@gmail.com | 7123456890 | 35 New York | 0 |
| 9 | Katherine | Pierce | piercekatherine@gmail.com | 8123456790 | 29 Paris | 1 |
| 10 | Alaric | Saltzman | alaricsaltzman@gmail.com | 9123456780 | 23 Vatican | 1 |
| 11 | Rebekah | Michaelson | rebekahmichaelson@gmail.com | 9871234560 | 12 London | 0 |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> SELECT * FROM ORDERS;
+-----+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount | Status |
+-----+-----+-----+-----+-----+
| 12 | 1 | 2022-01-01 | 2200.00 | SHIPPED |
| 23 | 2 | 2022-01-02 | 6600.00 | SHIPPED |
| 42 | 4 | 2022-01-04 | 17600.00 | PENDING |
| 50 | 5 | 2022-01-05 | 2750.00 | Shipped |
| 69 | 6 | 2022-01-06 | 6600.00 | Dispatched |
| 70 | 7 | 2022-01-07 | 15400.00 | SHIPPED |
| 91 | 9 | 2022-01-09 | 22000.00 | Out for delivery |
| 103 | 10 | 2022-01-10 | 11000.00 | Out for delivery |
+-----+-----+-----+-----+-----+
```

### 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, c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone, c.Address  
FROM Orders o  
JOIN Customers c ON o.CustomerID = c.CustomerID;
```



The screenshot shows the MySQL 8.0 Command Line Client interface. The command line shows the following SQL query:

```
mysql>  
mysql>  
mysql> SELECT o.OrderID, c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone, c.Address  
-> FROM Orders o  
-> JOIN Customers c ON o.CustomerID = c.CustomerID;
```

The result set is displayed as a table:

OrderID	CustomerID	FirstName	LastName	Email	Phone	Address
12	1	Damon	Salvatore	damonsalvatore@gmail.com	1234567890	23 Mystic Falls
23	2	Stefan	Salvatore	stefansalvatore@gmail.com	2134567890	25 New York
42	4	Caroline	Forbes	carolineforbes@gmail.com	7612345890	90 San Francisco
50	5	Bonny	Bennet	bonnybennet@gmail.com	4123567890	07 Los Angeles
69	6	Jeremy	Gilbert	jeremygilbert@gmail.com	5123467890	55 Vatican
70	7	Klaus	Michaelson	klausmichaelson@gmail.com	6123457890	34 New York
91	9	Katherine	Pierce	piercekatherina@gmail.com	8123456790	29 Paris
103	10	Alaric	Saltzman	alaricsaltzman@gmail.com	9123456780	23 Vatican

8 rows in set (0.00 sec)

```
mysql>
```

**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 TotalRevenue
FROM Products p
INNER JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductName;
```

The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". It displays the MySQL welcome message, connection details, and copyright information. The user then runs an SQL query to calculate the total revenue for each product in the "Products" table, joining it with the "OrderDetails" table based on ProductID. The results are displayed in a tabular format:

ProductName	TotalRevenue
Smartphone	2200.00
Drone	15400.00
Wireless Earbuds	11000.00
Headphones	2750.00
Camera	6600.00
Smart Home Device	22000.00
Laptop	6600.00
Tablet	17600.00

At the bottom of the terminal, the system tray shows weather information (31°C Haze), a search bar, and various application icons. The taskbar at the very bottom includes the Start button, a search bar, and icons for File Explorer, Task View, Edge, Google Chrome, YouTube, Facebook, and others. System status icons for battery, signal, and volume are also present.

```
mysql> use techshop;
Database changed
mysql> SELECT p.ProductName, SUM(od.Quantity * p.Price) AS TotalRevenue
-> FROM Products p
-> JOIN OrderDetails od ON p.ProductID = od.ProductID
-> GROUP BY p.ProductName;
+-----+-----+
| ProductName | TotalRevenue |
+-----+-----+
| Smartphone   |      2200.00 |
| Drone        |     15400.00 |
| Wireless Earbuds | 11000.00 |
| Headphones    |      2750.00 |
| Camera        |      6600.00 |
| Smart Home Device | 22000.00 |
| Laptop         |      6600.00 |
| Tablet         |     17600.00 |
+-----+-----+
8 rows in set (0.04 sec)

mysql>
```

**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.Email, c.Phone, c.Address  
FROM Customers c  
WHERE c.CustomerID IN (SELECT DISTINCT o.CustomerID FROM Orders o);
```

The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The command entered is:

```
mysql>  
mysql>  
mysql> SELECT c.FirstName, c.LastName, c.Email, c.Phone, c.Address  
    -> FROM Customers c  
    -> WHERE c.CustomerID IN (SELECT DISTINCT o.CustomerID FROM Orders o);
```

The result is a table with the following data:

FirstName	LastName	Email	Phone	Address
Damon	Salvatore	damonsalvatore@gmail.com	1234567890	23 Mystic Falls
Stefan	Salvatore	stefansalvatore@gmail.com	2134567890	25 New York
Caroline	Forbes	carolineforbes@gmail.com	7612345890	90 San Francisco
Bonny	Bennet	bonnybennet@gmail.com	4123567890	07 Los Angeles
Jeremy	Gilbert	jeremygilbert@gmail.com	5123467890	55 Vatican
Klaus	Michaelson	klausmichaelson@gmail.com	6123457890	34 New York
Katherine	Pierce	piercekatherina@gmail.com	8123456790	29 Paris
Alaric	Saltzman	alaricsaltzman@gmail.com	9123456780	23 Vatican

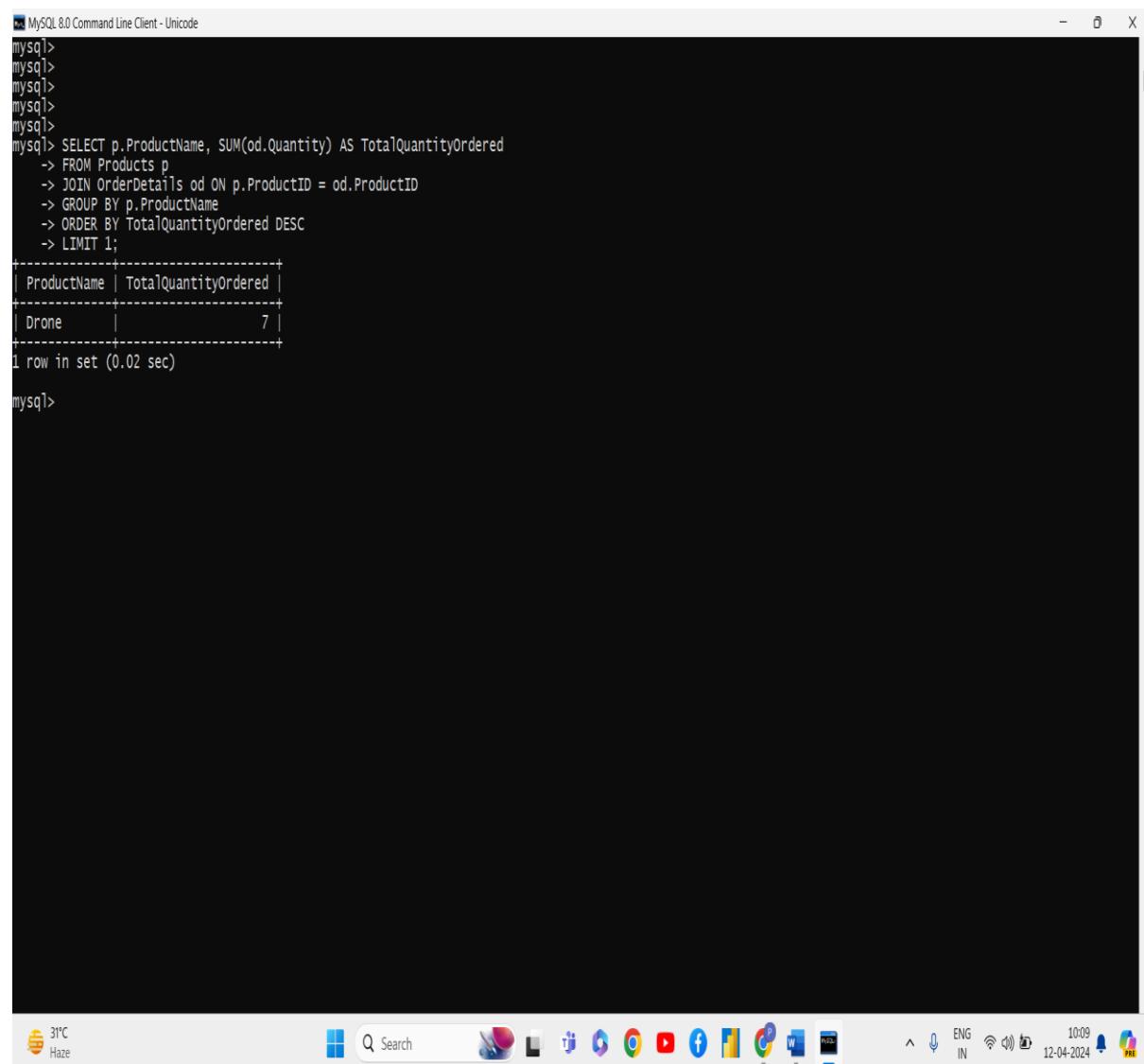
8 rows in set (0.00 sec)

```
mysql>
```

The system tray at the bottom shows the date and time as 12-04-2024, 10:05. Other icons include a weather icon (31°C Haze), search, file explorer, browser, and social media links.

**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, SUM(od.Quantity) AS TotalQuantityOrdered
FROM Products p
JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductName
ORDER BY TotalQuantityOrdered DESC
LIMIT 1;
```



The screenshot shows the MySQL 8.0 Command Line Client interface. The command line area displays the following SQL query:

```
mysql>
mysql>
mysql>
mysql>
mysql> SELECT p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered
-> FROM Products p
-> JOIN OrderDetails od ON p.ProductID = od.ProductID
-> GROUP BY p.ProductName
-> ORDER BY TotalQuantityOrdered DESC
-> LIMIT 1;
```

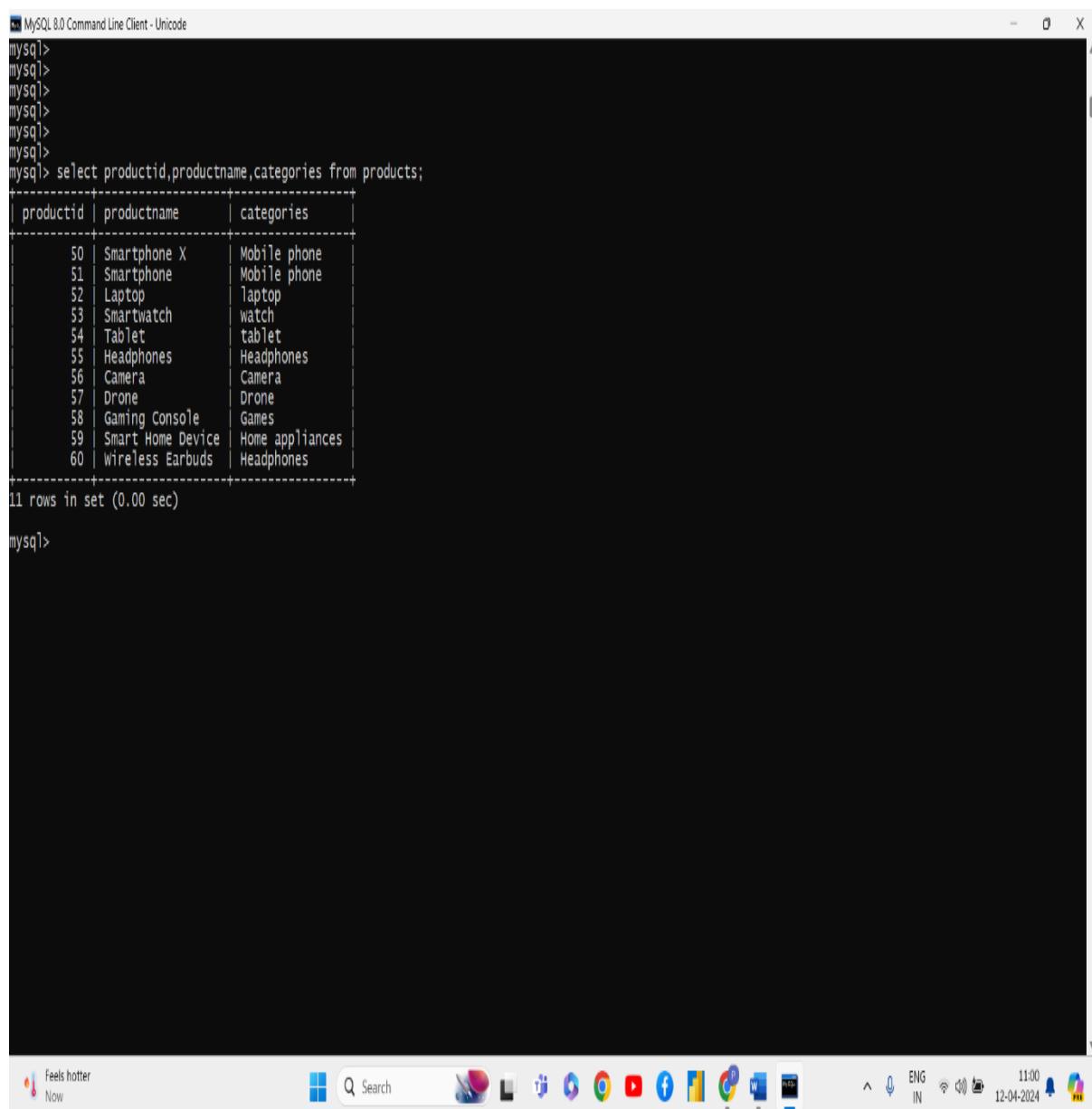
Below the command line, the results are shown in a table:

ProductName	TotalQuantityOrdered
Drone	7

Text at the bottom of the client window indicates "1 row in set (0.02 sec)". The system tray at the bottom of the screen shows the date and time as "12-04-2024 10:09".

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

```
SELECT productid,productname,categories from products;
```



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The user has entered the following SQL query:

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select productid,productname,categories from products;
```

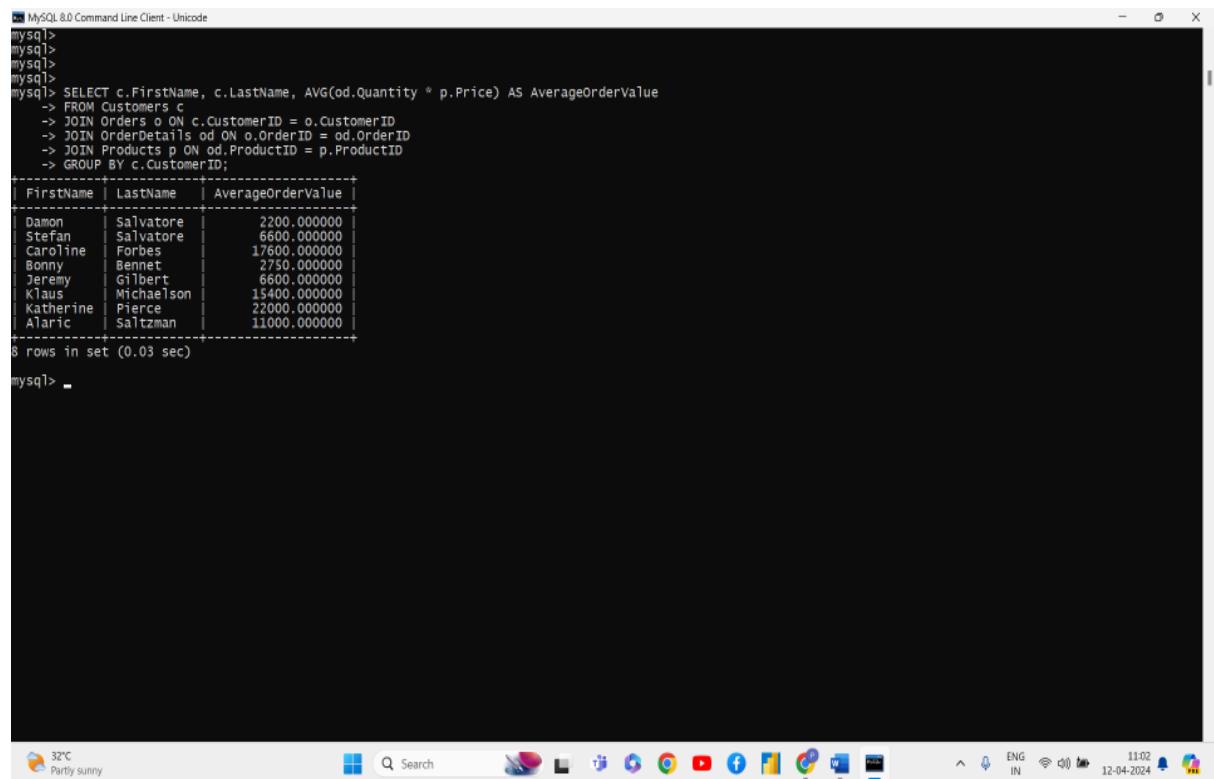
The query returns 11 rows of data, which is displayed in a table:

productid	productname	categories
50	Smartphone X	Mobile phone
51	Smartphone	Mobile phone
52	Laptop	laptop
53	Smartwatch	watch
54	Tablet	tablet
55	Headphones	Headphones
56	Camera	Camera
57	Drone	Drone
58	Gaming Console	Games
59	Smart Home Device	Home appliances
60	Wireless Earbuds	Headphones

Below the table, the message "11 rows in set (0.00 sec)" is displayed. The MySQL prompt "mysql>" appears at the bottom of the window.

**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(od.Quantity * p.Price) AS AverageOrderValue
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
GROUP BY c.CustomerID;
```



The screenshot shows the MySQL 8.0 Command Line Client window. The command entered is:

```
mysql> SELECT c.FirstName, c.LastName, AVG(od.Quantity * p.Price) AS AverageOrderValue
-> 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
-> GROUP BY c.CustomerID;
```

The result set is displayed as a table:

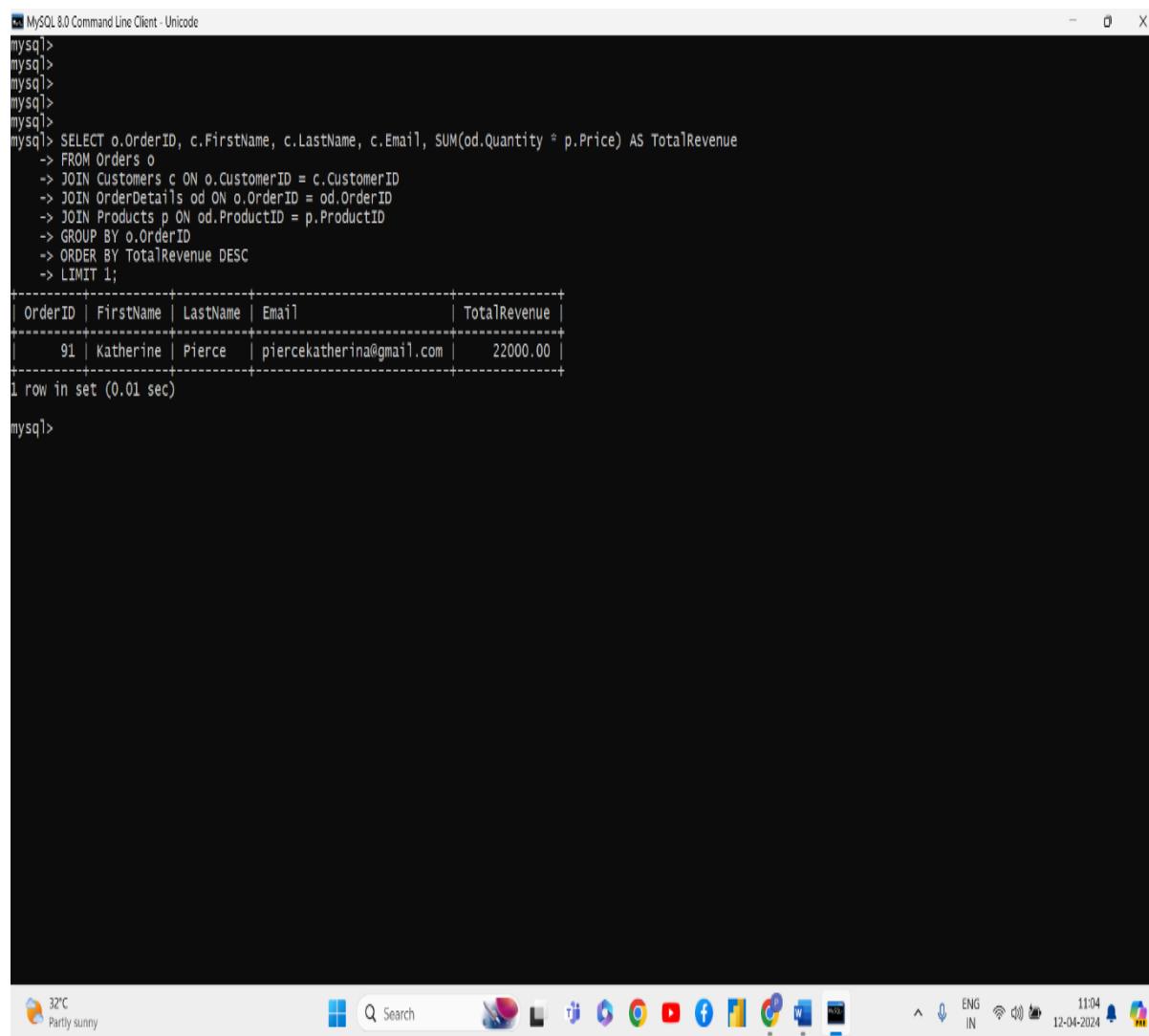
FirstName	LastName	AverageOrderValue
Damon	Salvatore	2200.000000
Stefan	Salvatore	6600.000000
Caroline	Forbes	17600.000000
Bonny	Bennet	2750.000000
Jeremy	Gilbert	6600.000000
Klaus	Michaelson	15400.000000
Katherine	Pierce	22000.000000
Alaric	Saltzman	11000.000000

8 rows in set (0.03 sec)

At the bottom of the window, there is a taskbar with various icons and system status information. The weather icon shows "32°C Partly sunny". The system tray shows the date and time as "12-04-2024 11:02".

**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, (od.Quantity * p.Price) AS TotalRevenue
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
ORDER BY TotalRevenue DESC
LIMIT 1;
```



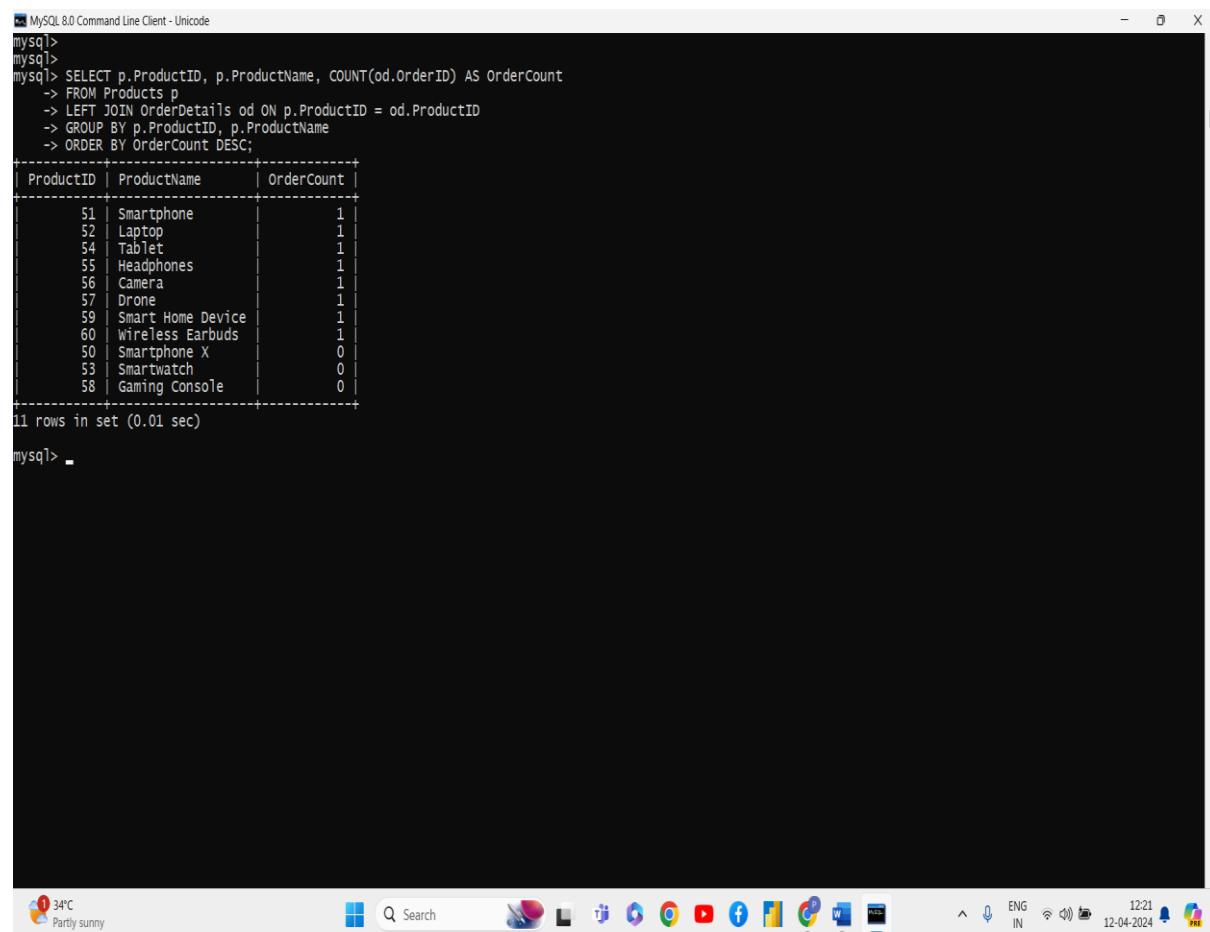
The screenshot shows the MySQL 8.0 Command Line Client window. The query is entered in the command line, and the result is displayed as a table. The table has four columns: OrderID, FirstName, LastName, and Email. The single row returned is for OrderID 91, belonging to Katherine Pierce with email piercekatherine@gmail.com, with a TotalRevenue of 22000.00.

OrderID	FirstName	LastName	Email	TotalRevenue
91	Katherine	Pierce	piercekatherine@gmail.com	22000.00

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.**

```
SELECT p.ProductID, p.ProductName, COUNT(od.OrderID) AS OrderCount
FROM Products p
LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductID, p.ProductName
ORDER BY OrderCount DESC;
```



The screenshot shows the MySQL 8.0 Command Line Client window. The command entered is:

```
mysql> SELECT p.ProductID, p.ProductName, COUNT(od.OrderID) AS OrderCount
-> FROM Products p
-> LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
-> GROUP BY p.ProductID, p.ProductName
-> ORDER BY OrderCount DESC;
```

The result set is displayed as a table:

ProductID	ProductName	OrderCount
51	Smartphone	1
52	Laptop	1
54	Tablet	1
55	Headphones	1
56	Camera	1
57	Drone	1
59	Smart Home Device	1
60	Wireless Earbuds	1
50	Smartphone X	0
53	Smartwatch	0
58	Gaming Console	0

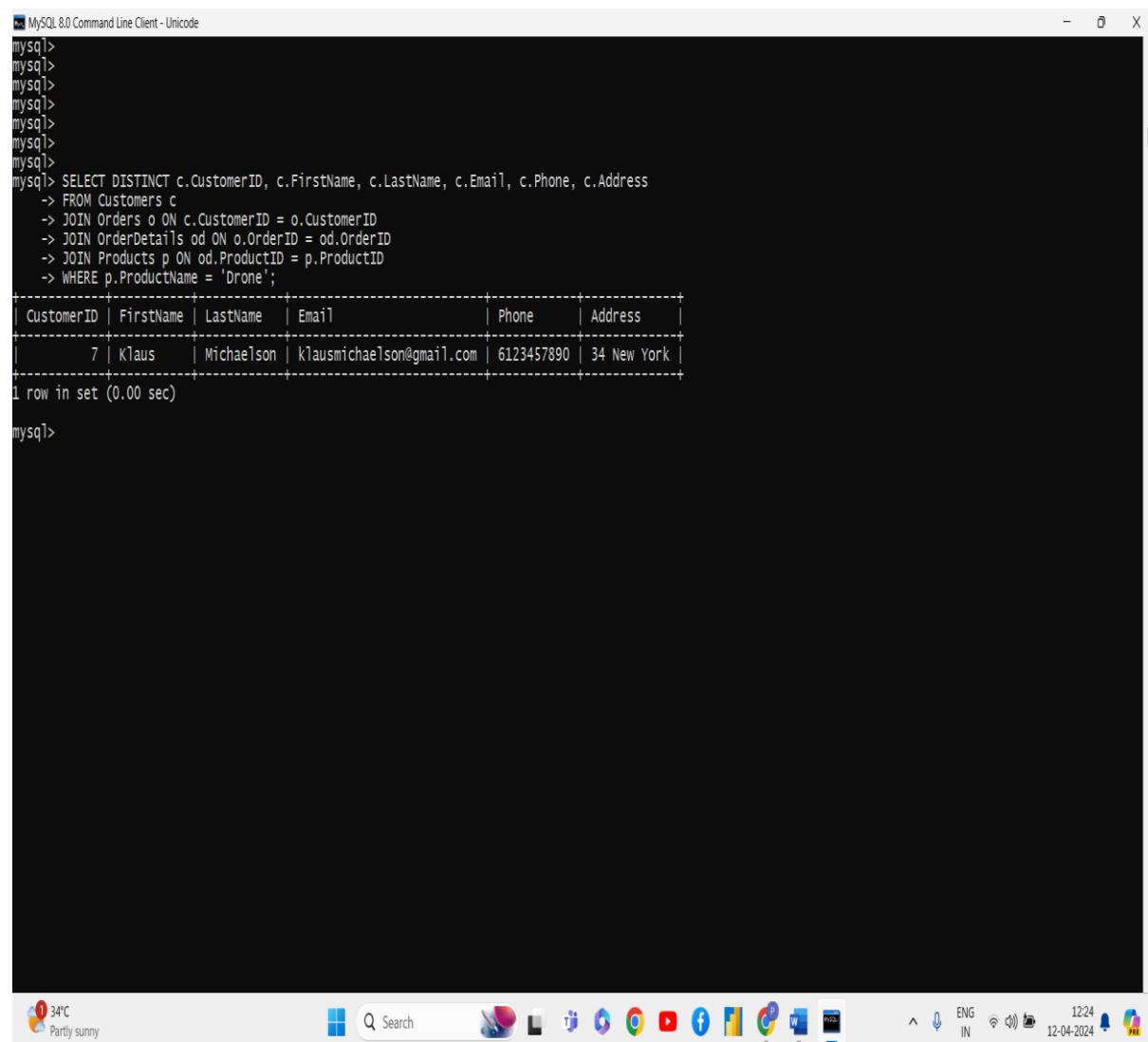
11 rows in set (0.01 sec)

At the bottom of the window, the system tray shows the date and time as 12-04-2024 12:21, along with icons for battery, signal, and network.

**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 DISTINCT c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone, c.Address  
FROM Customers c  
JOIN Orders o ON c.CustomerID = o.CustomerID  
JOIN OrderDetails od ON o.OrderID = od.OrderID  
JOIN Products p ON od.ProductID = p.ProductID  
WHERE p.ProductName = 'Drone';
```



The screenshot shows the MySQL 8.0 Command Line Client window. The command line area contains the following SQL query:

```
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql> SELECT DISTINCT c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone, c.Address  
-> FROM Customers c  
-> JOIN Orders o ON c.CustomerID = o.CustomerID  
-> JOIN OrderDetails od ON o.OrderID = od.OrderID  
-> JOIN Products p ON od.ProductID = p.ProductID  
-> WHERE p.ProductName = 'Drone';
```

Below the command line, the results are displayed in a tabular format:

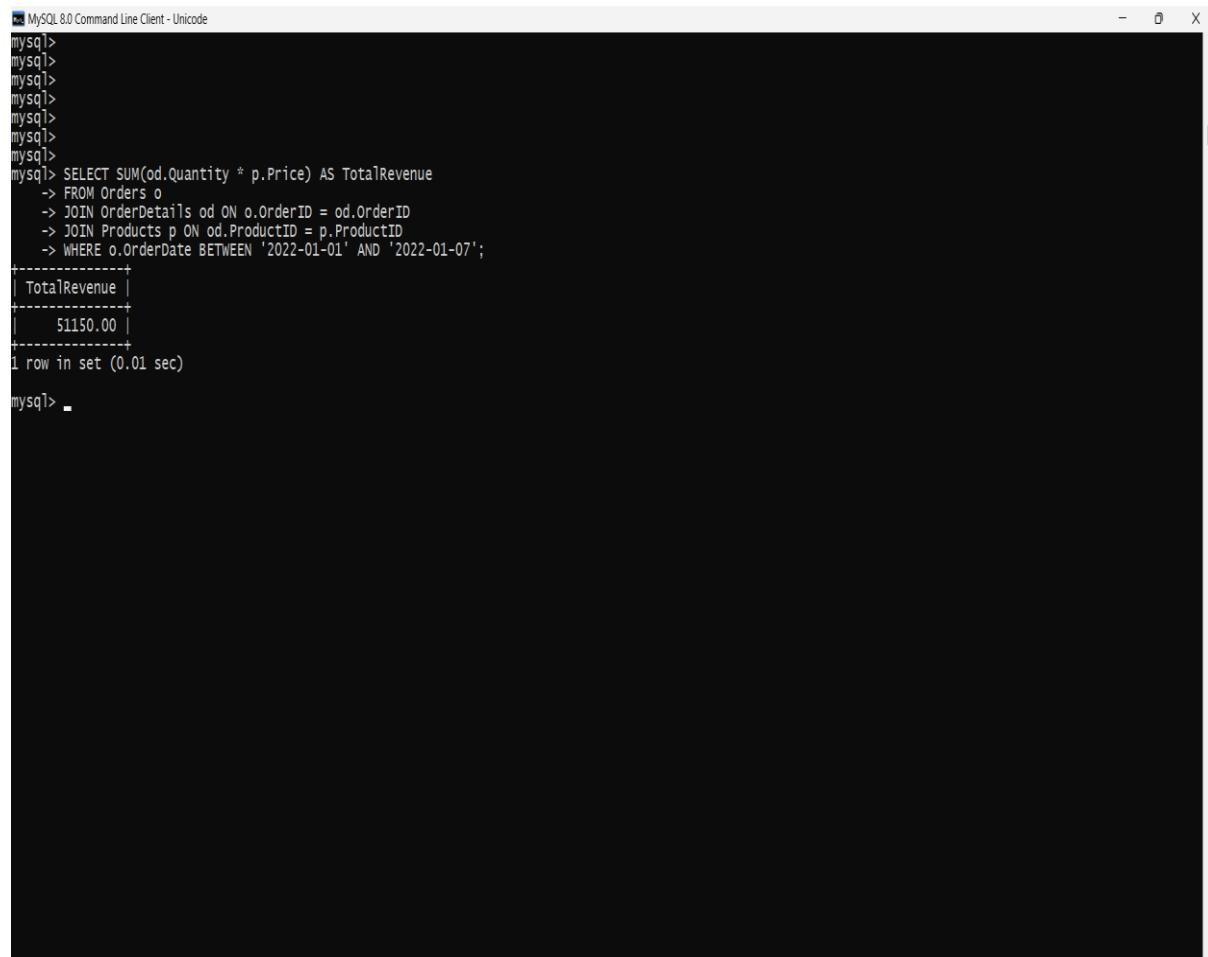
CustomerID	FirstName	LastName	Email	Phone	Address
7	Klaus	Michaelson	klausmichaelson@gmail.com	6123457890	34 New York

Text at the bottom of the results indicates "1 row in set (0.00 sec)".

The system tray at the bottom of the screen shows various icons, including a weather icon (34°C, Partly sunny), a search bar, and several application icons (File Explorer, Task View, Edge, Google Chrome, YouTube, Facebook, Mail, etc.). Language and date/time settings are also visible.

**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.**

```
SELECT SUM(od.Quantity * p.Price) AS TotalRevenue  
FROM Orders o  
JOIN OrderDetails od ON o.OrderID = od.OrderID  
JOIN Products p ON od.ProductID = p.ProductID  
WHERE o.OrderDate BETWEEN '2022-01-01' AND '2022-01-07';
```



The screenshot shows the MySQL 8.0 Command Line Client interface. The command window displays the following SQL query and its execution results:

```
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql> SELECT SUM(od.Quantity * p.Price) AS TotalRevenue  
-> FROM Orders o  
-> JOIN OrderDetails od ON o.OrderID = od.OrderID  
-> JOIN Products p ON od.ProductID = p.ProductID  
-> WHERE o.OrderDate BETWEEN '2022-01-01' AND '2022-01-07';  
+-----+  
| TotalRevenue |  
+-----+  
| 51150.00 |  
+-----+  
1 row in set (0.01 sec)  
  
mysql>
```

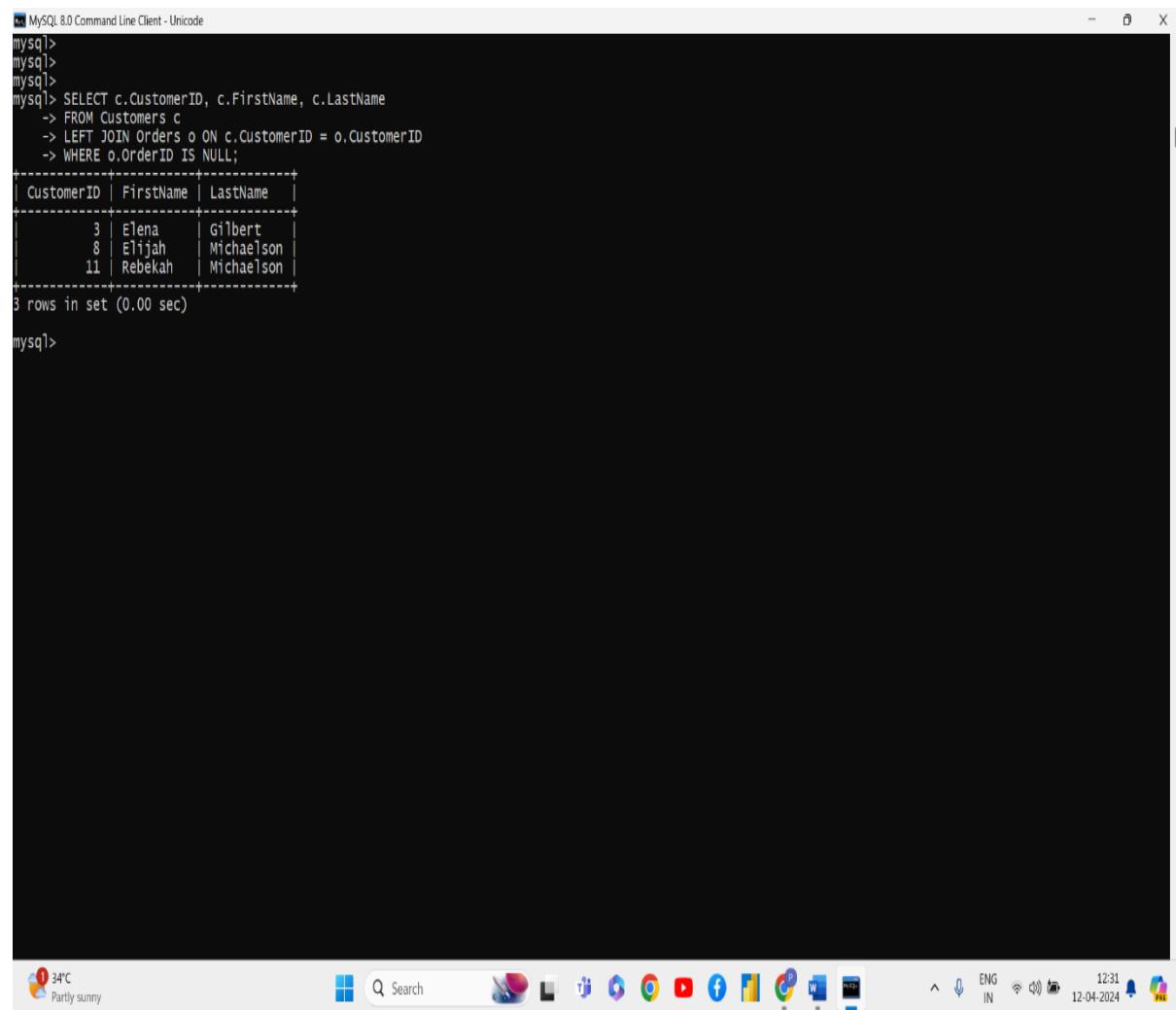
The result of the query is a single row with the column name "TotalRevenue" and the value "51150.00".

At the bottom of the screen, the Windows taskbar is visible, showing the date and time (12-04-2024, 12:28), system tray icons, and the Start button.

## 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 c  
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID  
WHERE o.OrderID IS NULL;
```



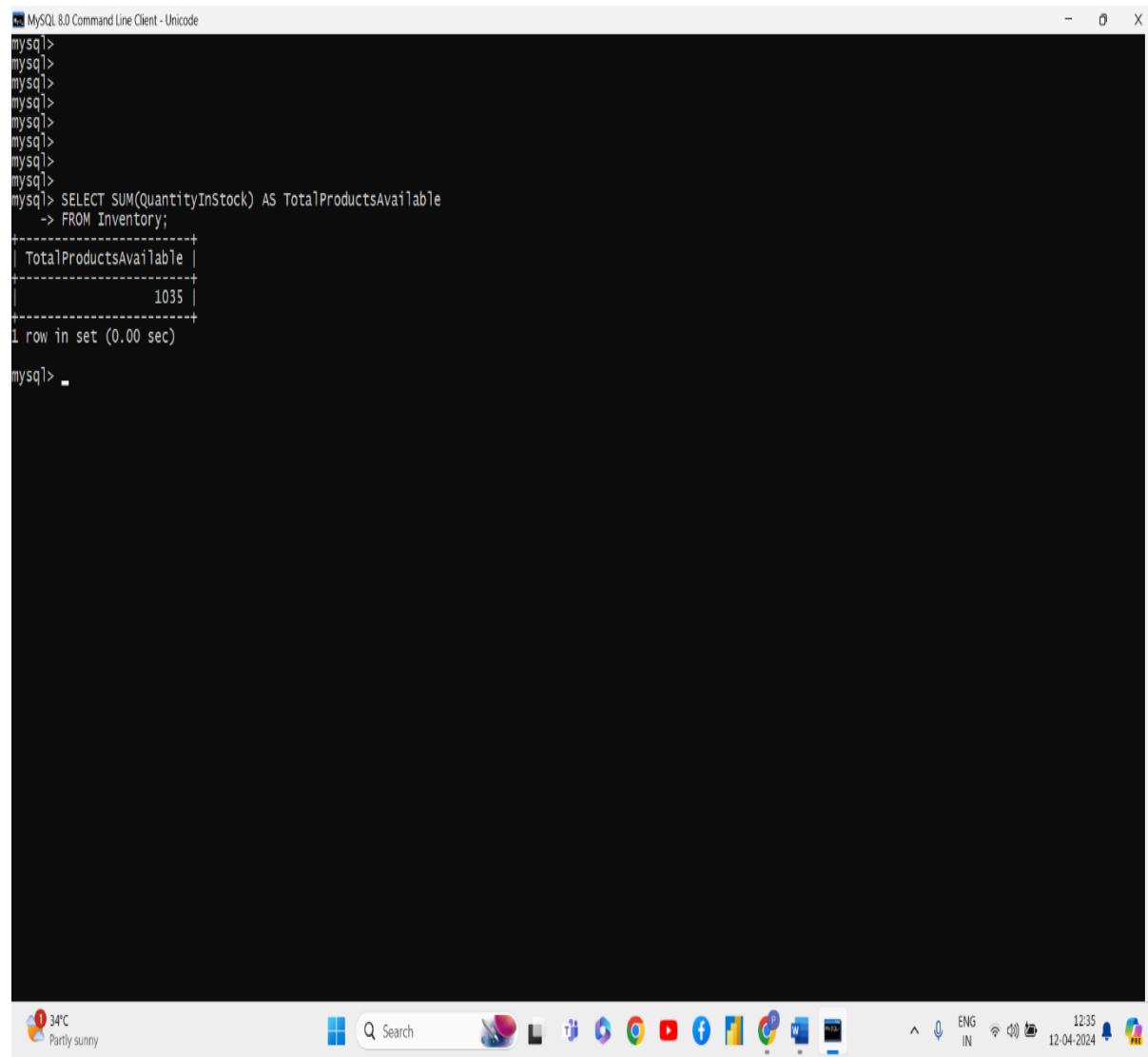
The screenshot shows the MySQL 8.0 Command Line Client window. The command line displays the following SQL query and its execution results:

```
mysql>  
mysql>  
mysql>  
mysql> SELECT c.CustomerID, c.FirstName, c.LastName  
-> FROM Customers c  
-> LEFT JOIN Orders o ON c.CustomerID = o.CustomerID  
-> WHERE o.OrderID IS NULL;  
+-----+-----+-----+  
| CustomerID | FirstName | LastName |  
+-----+-----+-----+  
| 3 | Elena | Gilbert |  
| 8 | Elijah | Michaelson |  
| 11 | Rebekah | Michaelson |  
+-----+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql>
```

The results show three rows of data from the Customers table where no corresponding row exists in the Orders table (OrderID is NULL). The columns are CustomerID, FirstName, and LastName.

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

```
SELECT SUM(QuantityInStock) AS TotalProductsAvailable  
FROM Inventory;
```



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The command entered is:

```
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql> SELECT SUM(QuantityInStock) AS TotalProductsAvailable  
-> FROM Inventory;
```

The output is:

TotalProductsAvailable
1035

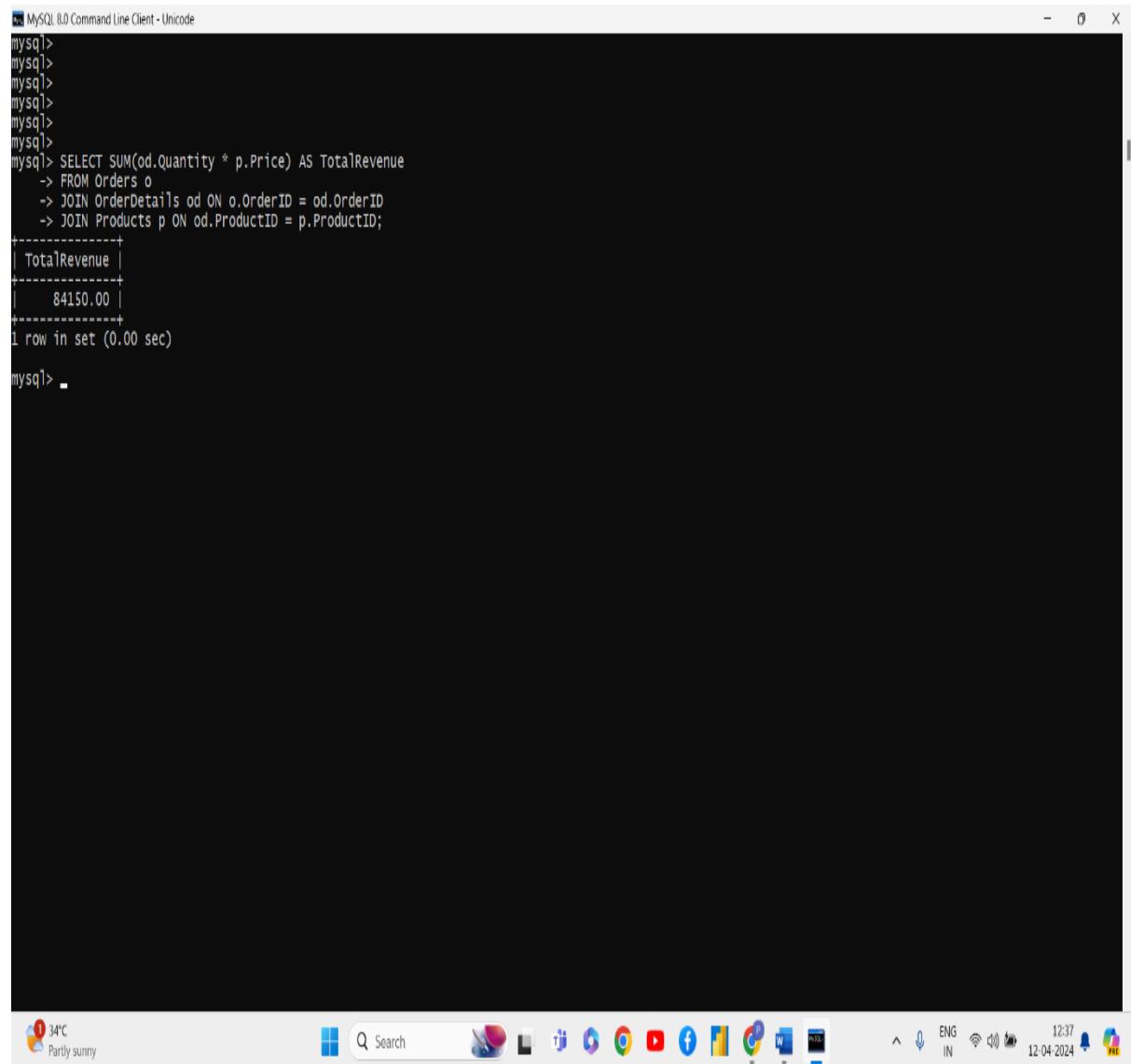
1 row in set (0.00 sec)

```
mysql> -
```

The system tray at the bottom of the screen shows the date and time as 12-04-2024, 12:35. It also displays icons for weather (34°C Partly sunny), search, and various application icons like Microsoft Edge, Google Chrome, and File Explorer.

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

```
SELECT SUM(od.Quantity * p.Price) AS TotalRevenue  
FROM Orders o  
JOIN OrderDetails od ON o.OrderID = od.OrderID  
JOIN Products p ON od.ProductID = p.ProductID;
```



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The user has entered the following SQL query:

```
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql> SELECT SUM(od.Quantity * p.Price) AS TotalRevenue  
    -> FROM Orders o  
    -> JOIN OrderDetails od ON o.OrderID = od.OrderID  
    -> JOIN Products p ON od.ProductID = p.ProductID;  
+-----+  
| TotalRevenue |  
+-----+  
| 84150.00 |  
+-----+  
1 row in set (0.00 sec)  
  
mysql> -
```

The query returns a single row with the value 84150.00 in the "TotalRevenue" column. The terminal window is set against a dark background, and the system tray at the bottom shows various icons and system status information.

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 AVG(od.Quantity) AS AverageQuantityOrdered  
FROM OrderDetails od  
JOIN Products p ON od.ProductID = p.ProductID  
WHERE p.Category ='Drone';
```

The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The user has entered the following SQL query:

```
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql> SELECT AVG(od.Quantity) AS AverageQuantityOrdered  
-> FROM OrderDetails od  
-> JOIN Products p ON od.ProductID = p.ProductID  
-> WHERE p.Categories = 'Drone';
```

The query returns a single row of results:

AverageQuantityOrdered
7.0000

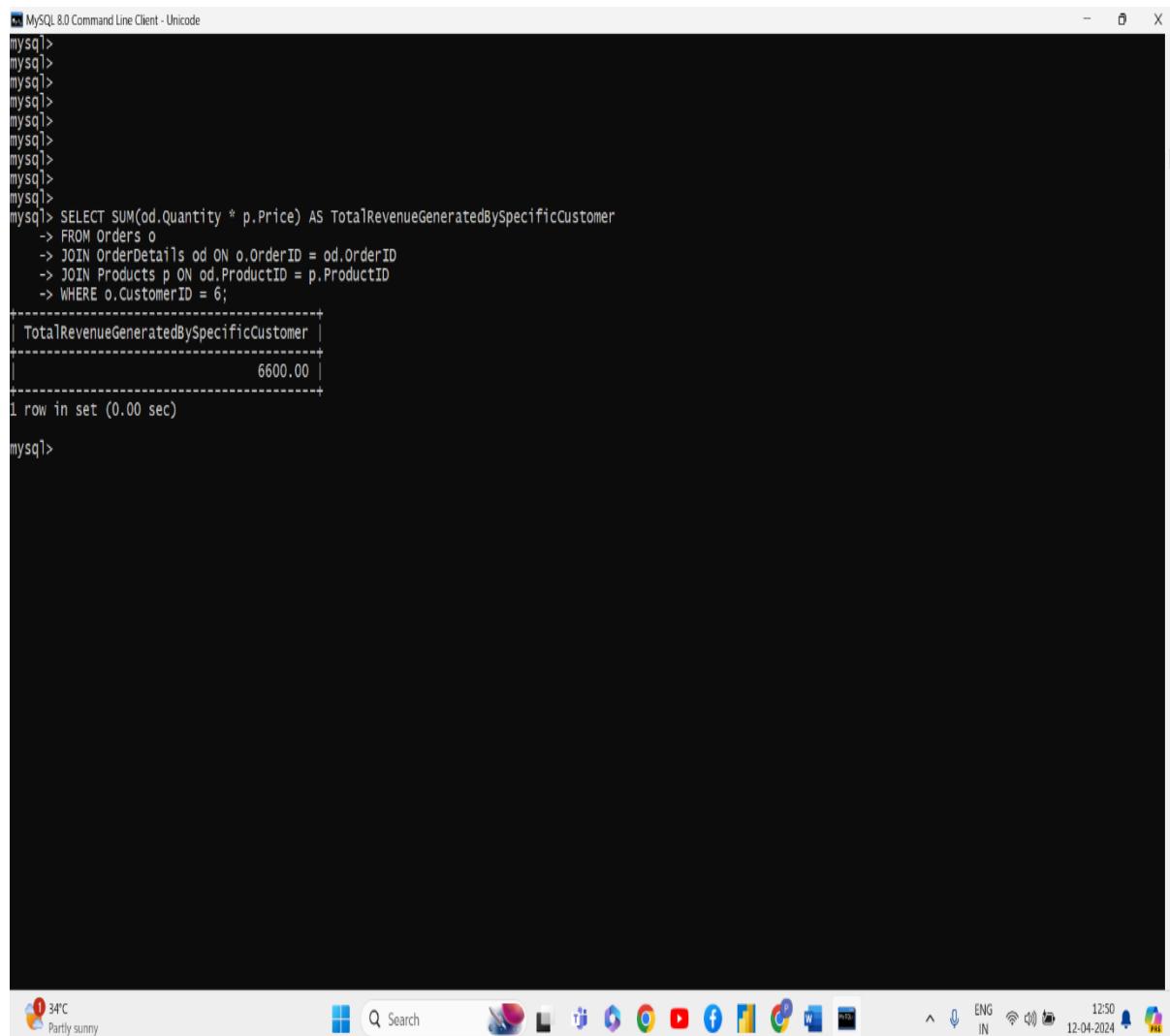
1 row in set (0.00 sec)

mysql> -

The system tray at the bottom of the screen shows the date and time as 12-04-2024 12:42, along with icons for battery, signal, and network status.

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 SUM(od.Quantity * p.Price) AS TotalRevenueGeneratedBySpecificCustomer
FROM Orders o
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
WHERE o.CustomerID = 6;
```



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The command line starts with several blank lines followed by the SQL query:

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> SELECT SUM(od.Quantity * p.Price) AS TotalRevenueGeneratedBySpecificCustomer
-> FROM Orders o
-> JOIN OrderDetails od ON o.OrderID = od.OrderID
-> JOIN Products p ON od.ProductID = p.ProductID
-> WHERE o.CustomerID = 6;
```

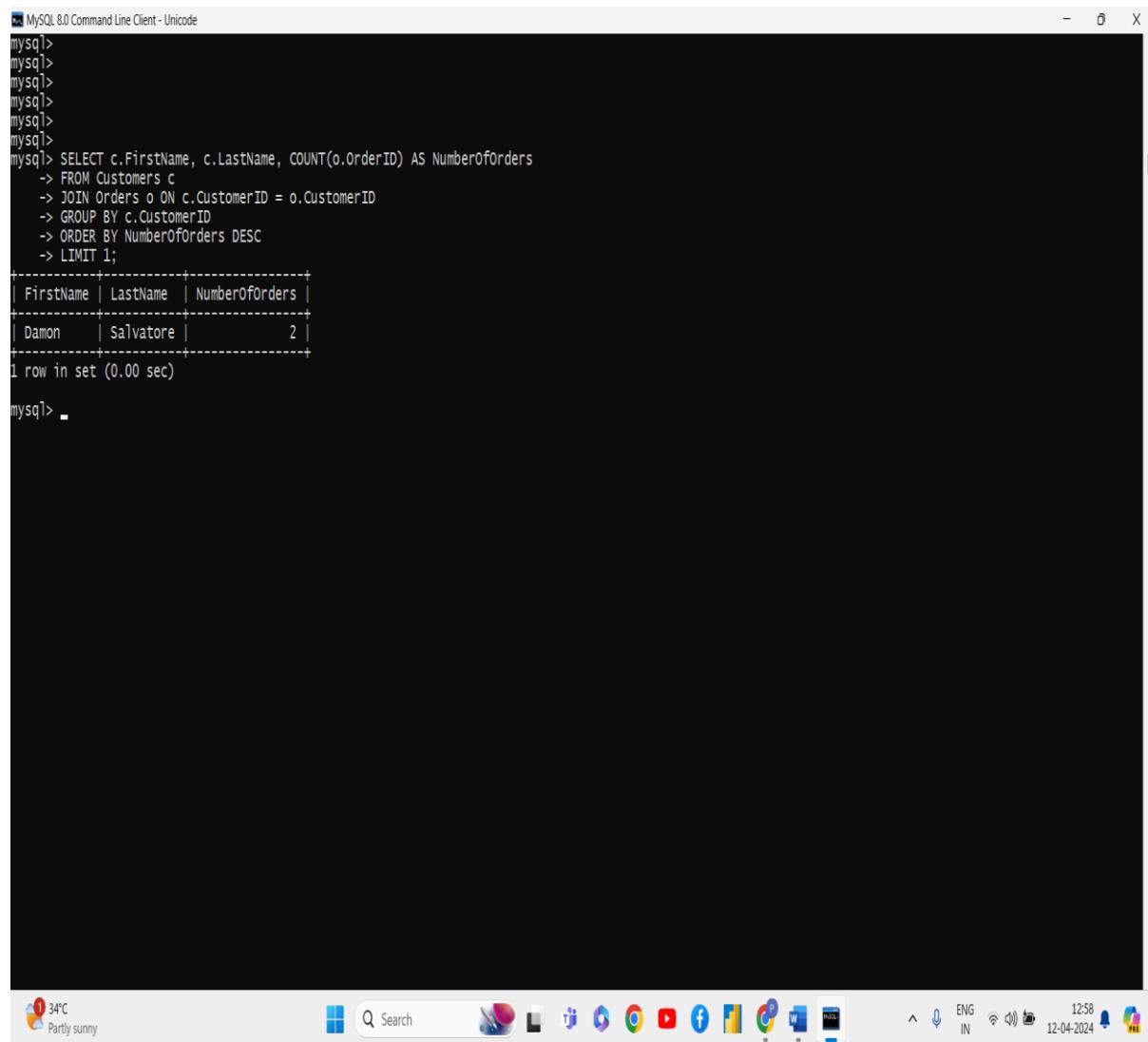
The output shows the result of the query:

TotalRevenueGeneratedBySpecificCustomer
6600.00

Below the table, it says "1 row in set (0.00 sec)". The MySQL prompt "mysql>" appears again at the bottom.

**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 NumberOfOrders
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID
ORDER BY NumberOfOrders DESC
LIMIT 1;
```



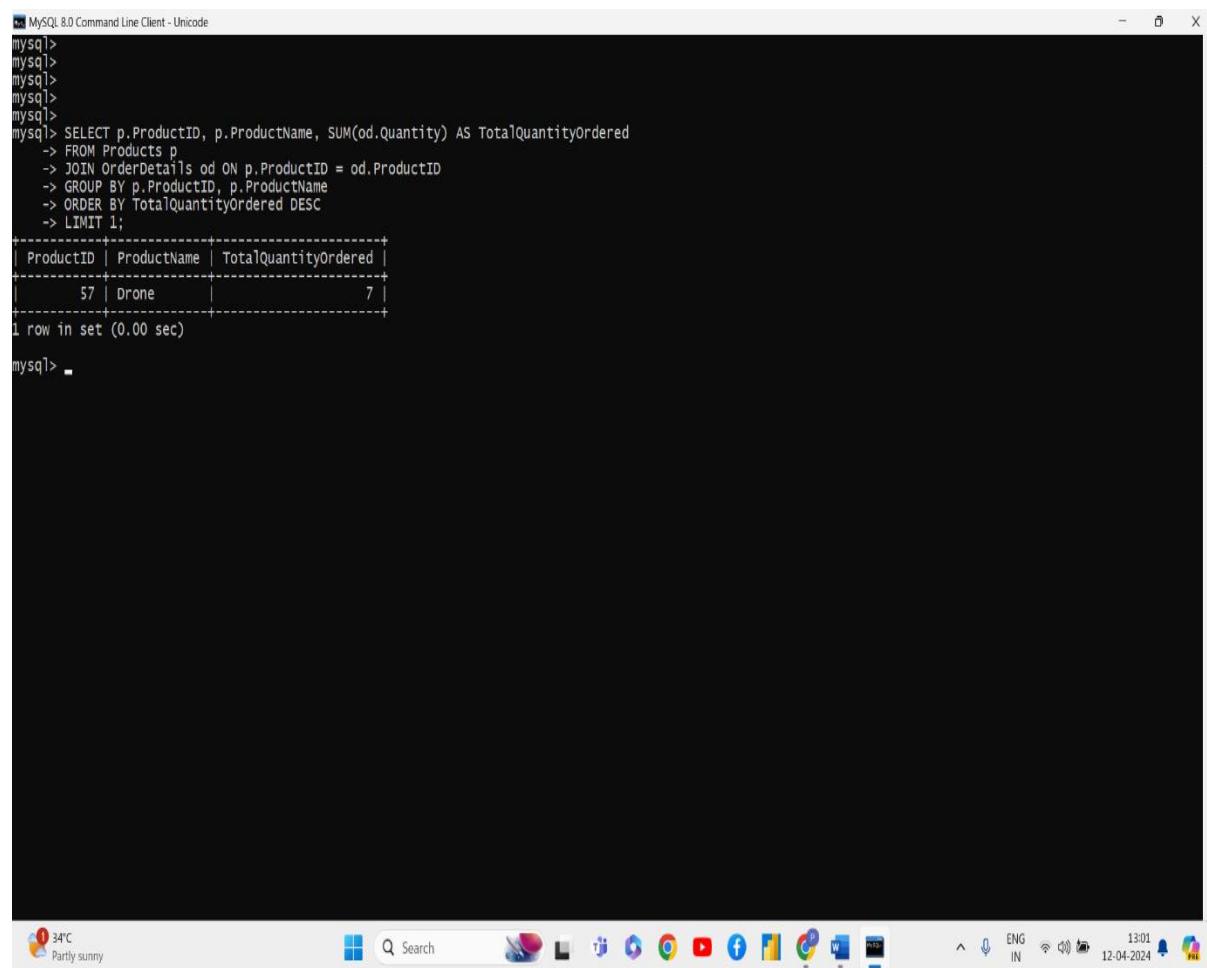
The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The command line starts with several blank lines followed by the SQL query from the previous block. The output shows a single row of data:

FirstName	LastName	NumberOfOrders
Damon	Salvatore	2

Below the table, it says "1 row in set (0.00 sec)". The MySQL prompt "mysql>" appears at the bottom left. The taskbar at the bottom of the screen shows various icons for weather, search, and other applications, along with system status indicators like battery level and network connection.

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 p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered  
FROM Products p  
JOIN OrderDetails od ON p.ProductID = od.ProductID  
GROUP BY p.ProductID, p.ProductName  
ORDER BY TotalQuantityOrdered DESC  
LIMIT 1;
```

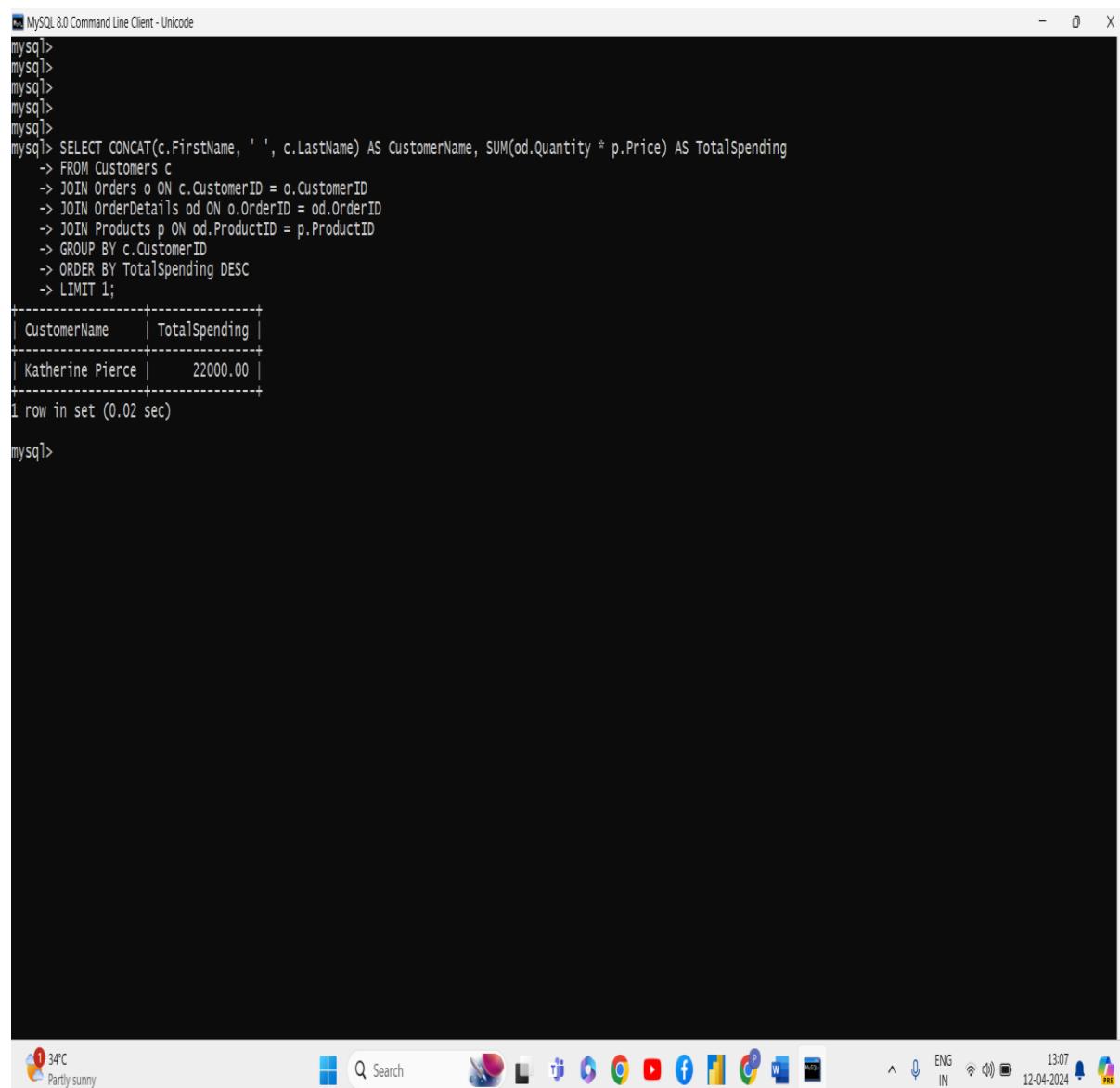


The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The user has run the provided SQL query to find the most popular product category. The output shows a single row with ProductID 57, ProductName "Drone", and TotalQuantityOrdered 7. The MySQL prompt "mysql>" is visible at the bottom.

```
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql> SELECT p.ProductID, p.ProductName, SUM(od.quantity) AS TotalQuantityOrdered  
-> FROM Products p  
-> JOIN OrderDetails od ON p.ProductID = od.ProductID  
-> GROUP BY p.ProductID, p.ProductName  
-> ORDER BY TotalQuantityordered DESC  
-> LIMIT 1;  
+-----+-----+-----+  
| ProductID | ProductName | TotalQuantityOrdered |  
+-----+-----+-----+  
|      57 | Drone       |             7 |  
+-----+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> _
```

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 CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, SUM(od.Quantity * p.Price)
AS TotalSpending
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY c.CustomerID
ORDER BY TotalSpending DESC
LIMIT 1;
```



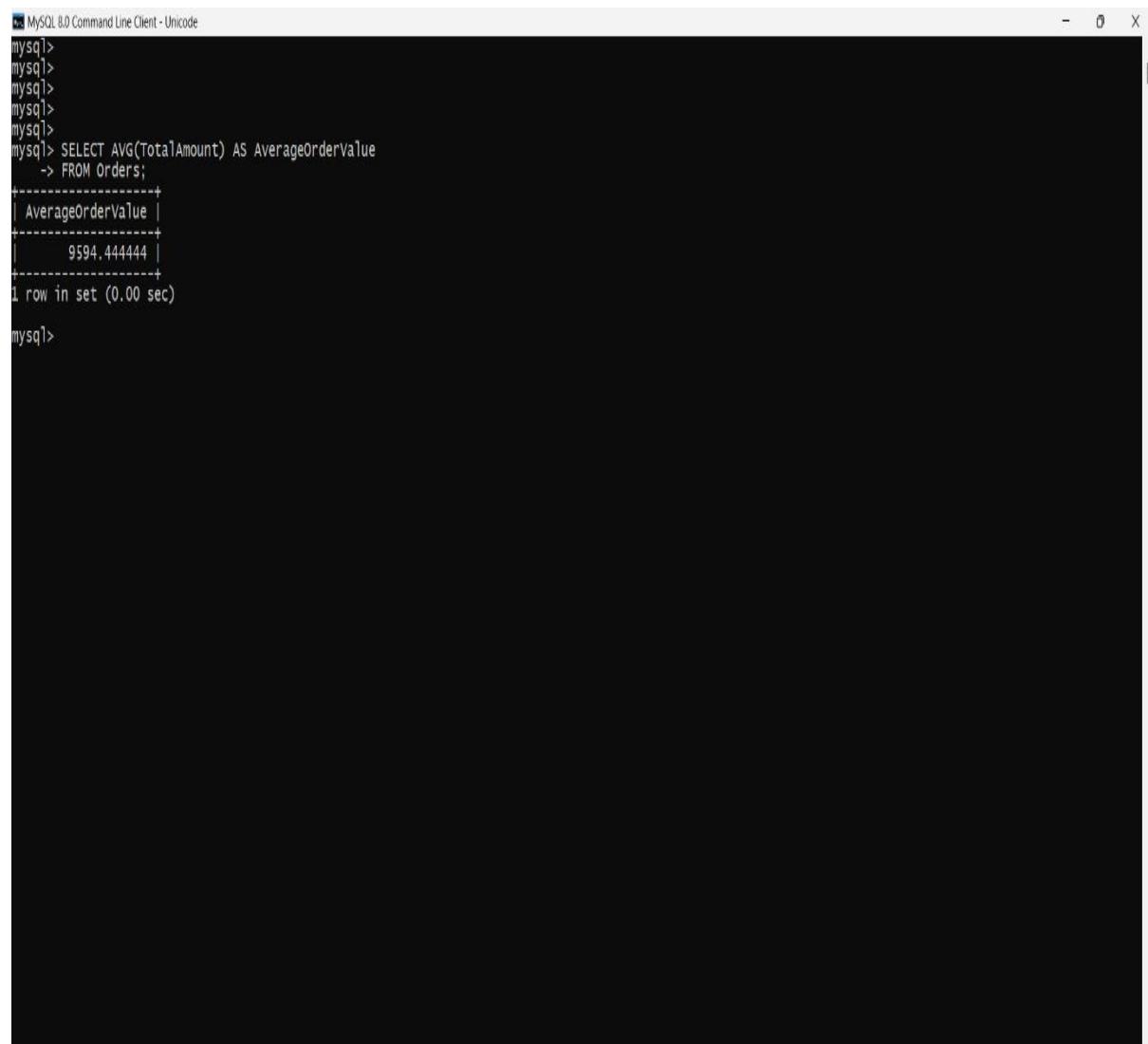
The screenshot shows the MySQL 8.0 Command Line Client interface. The command line shows the execution of the provided SQL query, which retrieves the customer name and total spending for the customer who has spent the most money. The result is displayed in a table format:

CustomerName	TotalSpending
Katherine Pierce	22000.00

Below the table, it says "1 row in set (0.02 sec)". The system tray at the bottom of the window shows various icons, including a weather icon indicating 34°C and partly sunny, and a date/time indicator showing 12-04-2024 at 13:07.

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

```
SELECT AVG(TotalAmount) AS AverageOrderValue  
FROM Orders;
```



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The user has entered the following SQL query:

```
mysql>  
mysql>  
mysql>  
mysql>  
mysql>  
mysql> SELECT AVG(TotalAmount) AS AverageOrderValue  
-> FROM Orders;
```

The output shows a single row with the average order value:

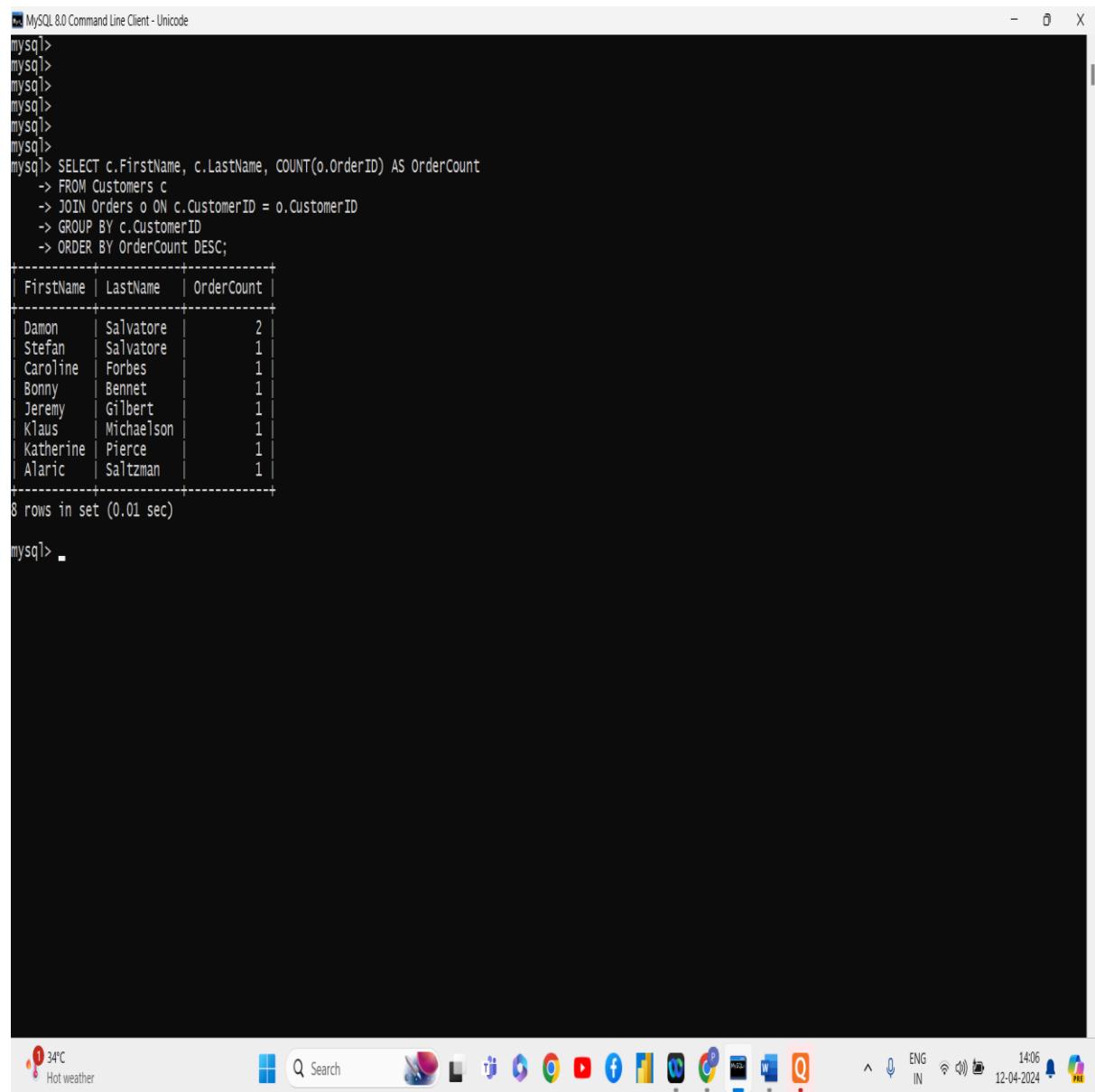
AverageOrderValue
9594.44444

Below the table, it says "1 row in set (0.00 sec)". The MySQL prompt "mysql>" appears at the bottom of the window.

At the bottom of the screen, there is a taskbar with various icons, including a news notification for "India issues Rs 6...", a search bar, and system status indicators like battery level and date/time (12-04-2024).

**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, c.LastName, COUNT(o.OrderID) AS OrderCount  
FROM Customers c  
JOIN Orders o ON c.CustomerID = o.CustomerID  
GROUP BY c.CustomerID  
ORDER BY OrderCount DESC;
```



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The user has run the provided SQL query to find the total number of orders placed by each customer. The results are displayed in a table:

FirstName	LastName	OrderCount
Damon	Salvatore	2
Stefan	Salvatore	1
Caroline	Forbes	1
Bonny	Bennet	1
Jeremy	Gilbert	1
Klaus	Michaelson	1
Katherine	Pierce	1
Alaric	Saltzman	1

Below the table, the message "8 rows in set (0.01 sec)" is shown. The MySQL prompt "mysql> " is visible at the bottom of the terminal window.