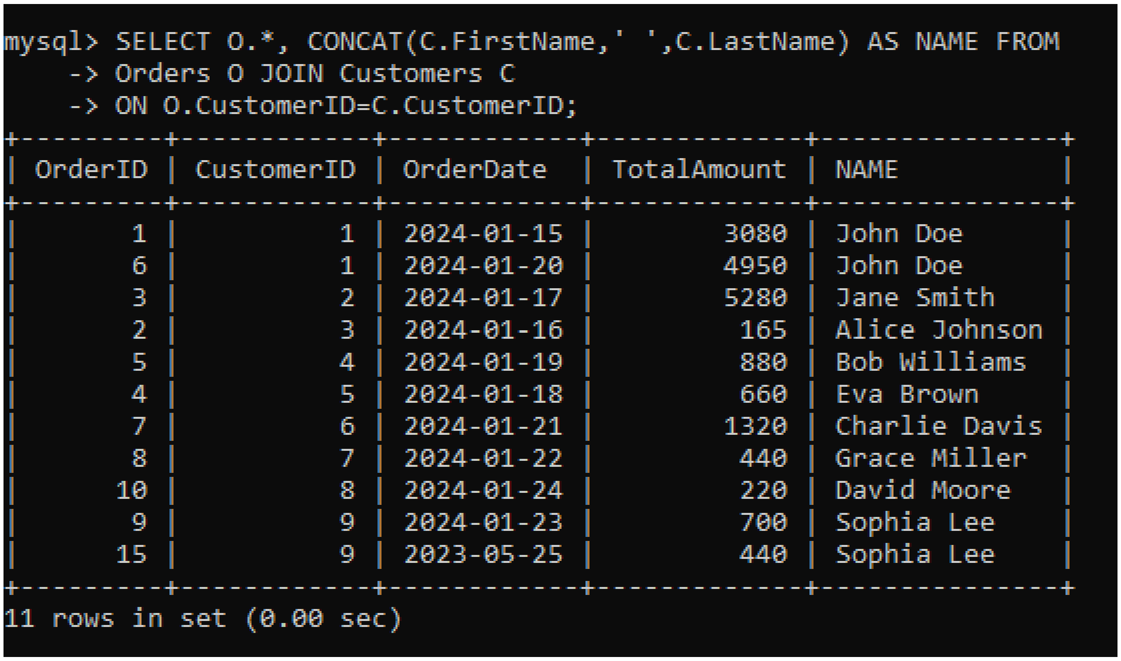
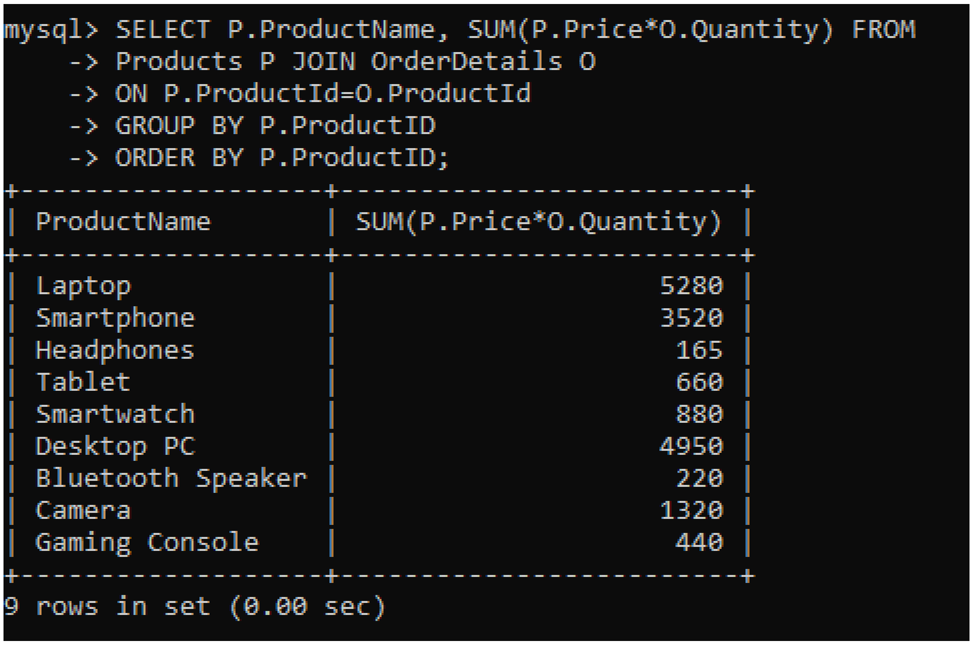
1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.



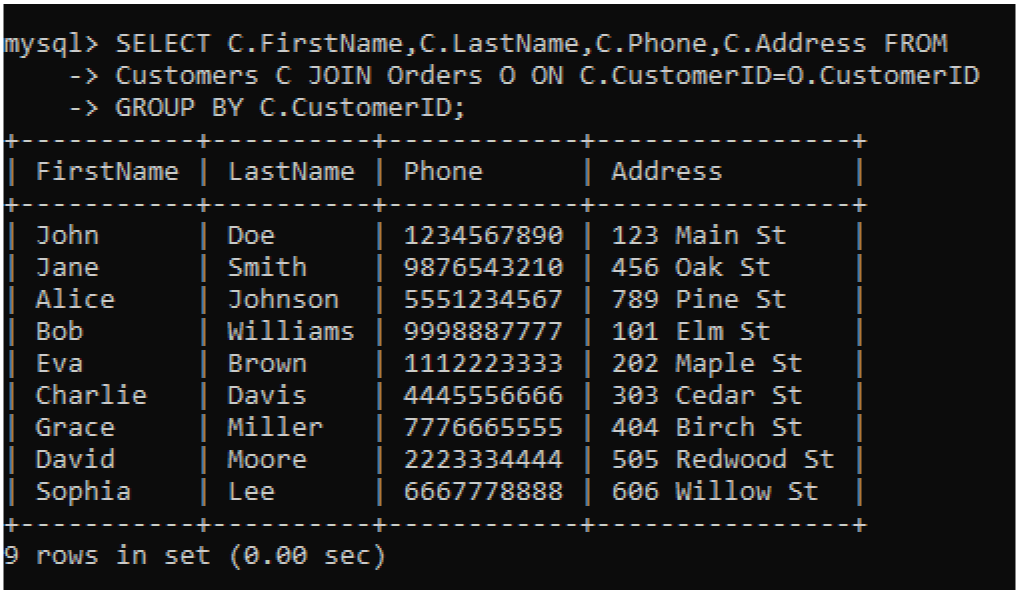
```
mysql> SELECT O.*, CONCAT(C.FirstName,' ',C.LastName) AS NAME FROM
    -> Orders O JOIN Customers C
    -> ON O.CustomerID=C.CustomerID;
+---------+------------+------------+-------------+---------------+
| OrderID | CustomerID | OrderDate  | TotalAmount | NAME          |
+---------+------------+------------+-------------+---------------+
|       1 |          1 | 2024-01-15 |        3080 | John Doe      |
|       6 |          1 | 2024-01-20 |        4950 | John Doe      |
|       3 |          2 | 2024-01-17 |        5280 | Jane Smith    |
|       2 |          3 | 2024-01-16 |         165 | Alice Johnson |
|       5 |          4 | 2024-01-19 |         880 | Bob Williams  |
|       4 |          5 | 2024-01-18 |         660 | Eva Brown     |
|       7 |          6 | 2024-01-21 |        1320 | Charlie Davis |
|       8 |          7 | 2024-01-22 |         440 | Grace Miller  |
|      10 |          8 | 2024-01-24 |         220 | David Moore   |
|       9 |          9 | 2024-01-23 |         700 | Sophia Lee    |
|      15 |          9 | 2023-05-25 |         440 | Sophia Lee    |
+---------+------------+------------+-------------+---------------+
11 rows in set (0.00 sec)
```

2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.



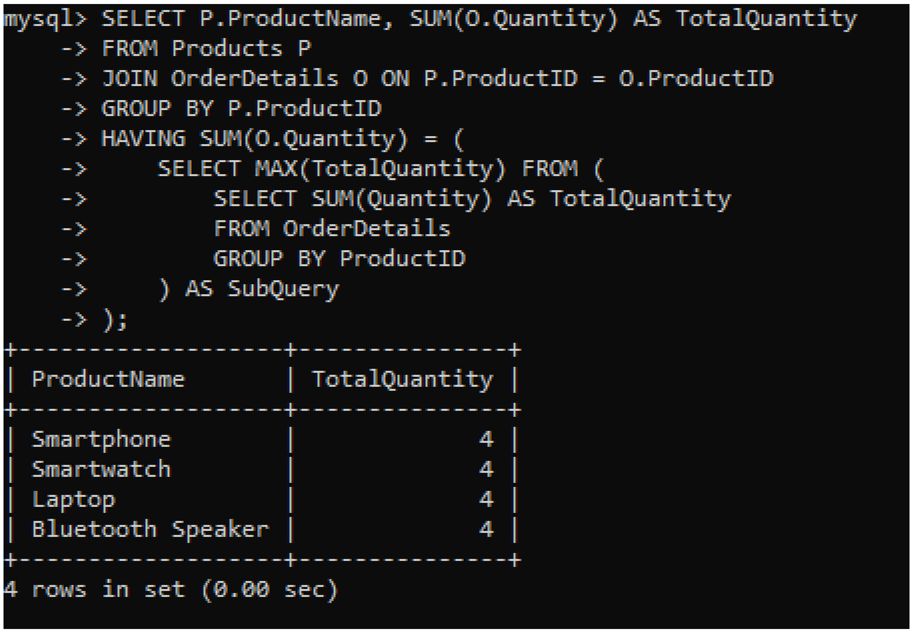
```
mysql> SELECT P.ProductName, SUM(P.Price*O.Quantity) FROM
    -> Products P JOIN OrderDetails O
    -> ON P.ProductId=O.ProductId
    -> GROUP BY P.ProductID
    -> ORDER BY P.ProductID;
+-------------------+-------------------------+
| ProductName       | SUM(P.Price*O.Quantity) |
+-------------------+-------------------------+
| Laptop            |                    5280 |
| Smartphone        |                    3520 |
| Headphones        |                     165 |
| Tablet            |                     660 |
| Smartwatch        |                     880 |
| Desktop PC        |                    4950 |
| Bluetooth Speaker |                     220 |
| Camera            |                    1320 |
| Gaming Console    |                     440 |
+-------------------+-------------------------+
9 rows in set (0.00 sec)
```

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



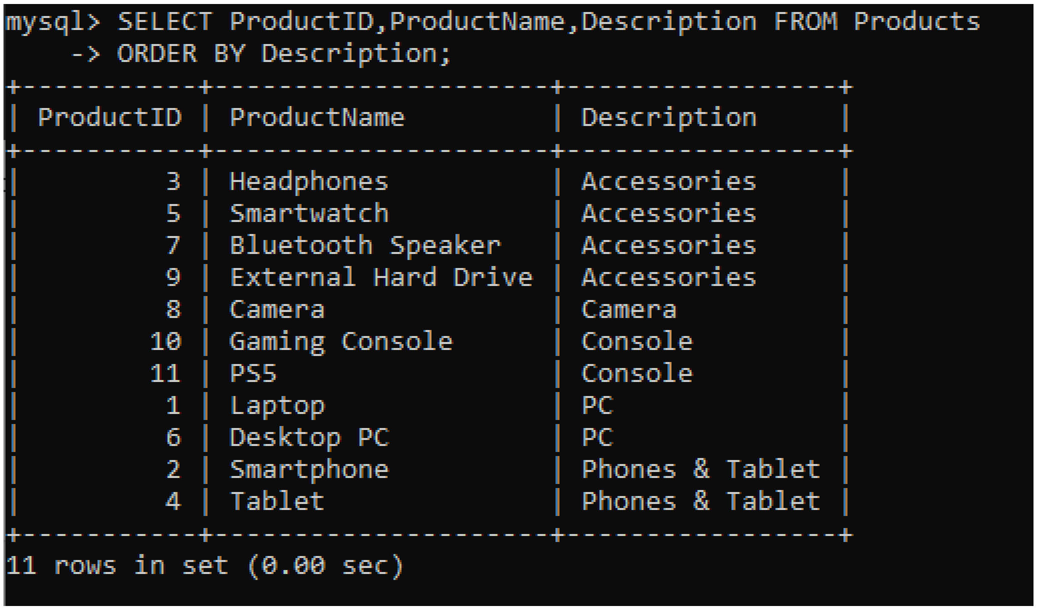
```
mysql> SELECT C.FirstName,C.LastName,C.Phone,C.Address FROM
    -> Customers C JOIN Orders O ON C.CustomerID=O.CustomerID
    -> GROUP BY C.CustomerID;
+-----------+----------+------------+----------------+
| FirstName | LastName | Phone      | Address        |
+-----------+----------+------------+----------------+
| John      | Doe      | 1234567890 | 123 Main St    |
| Jane      | Smith    | 9876543210 | 456 Oak St     |
| Alice     | Johnson  | 5551234567 | 789 Pine St    |
| Bob       | Williams | 9998887777 | 101 Elm St     |
| Eva       | Brown    | 1112223333 | 202 Maple St   |
| Charlie   | Davis    | 4445556666 | 303 Cedar St   |
| Grace     | Miller   | 7776665555 | 404 Birch St   |
| David     | Moore    | 2223334444 | 505 Redwood St |
| Sophia    | Lee      | 6667778888 | 606 Willow St  |
+-----------+----------+------------+----------------+
9 rows in set (0.00 sec)
```

4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.



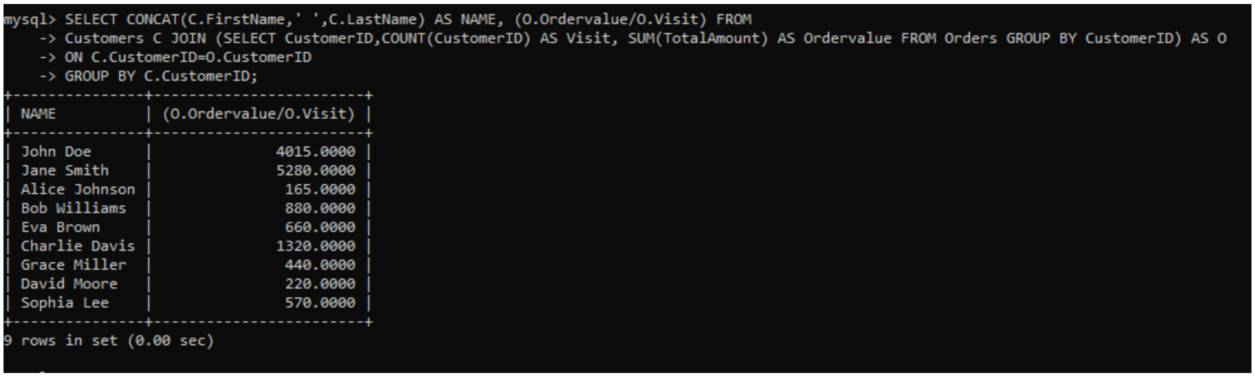
```
mysql> SELECT P.ProductName, SUM(O.Quantity) AS TotalQuantity
    -> FROM Products P
    -> JOIN OrderDetails O ON P.ProductID = O.ProductID
    -> GROUP BY P.ProductID
    -> HAVING SUM(O.Quantity) = (
    ->     SELECT MAX(TotalQuantity) FROM (
    ->         SELECT SUM(Quantity) AS TotalQuantity
    ->         FROM OrderDetails
    ->         GROUP BY ProductID
    ->     ) AS SubQuery
    -> );
+-------------------+---------------+
| ProductName       | TotalQuantity |
+-------------------+---------------+
| Smartphone        |             4 |
| Smartwatch        |             4 |
| Laptop            |             4 |
| Bluetooth Speaker |             4 |
+-------------------+---------------+
4 rows in set (0.00 sec)
```

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



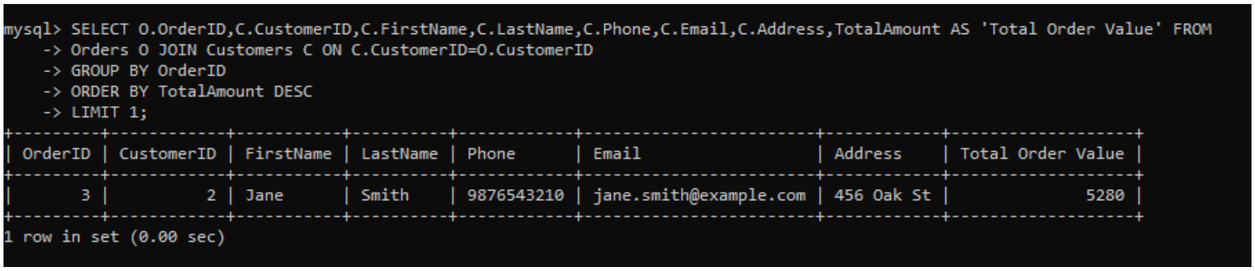
```
mysql> SELECT ProductID,ProductName,Description FROM Products
    -> ORDER BY Description;
+-----------+---------------------+-----------------+
| ProductID | ProductName         | Description     |
+-----------+---------------------+-----------------+
|         3 | Headphones          | Accessories     |
|         5 | Smartwatch          | Accessories     |
|         7 | Bluetooth Speaker   | Accessories     |
|         9 | External Hard Drive | Accessories     |
|         8 | Camera              | Camera          |
|        10 | Gaming Console      | Console         |
|        11 | PS5                 | Console         |
|         1 | Laptop              | PC              |
|         6 | Desktop PC          | PC              |
|         2 | Smartphone          | Phones & Tablet |
|         4 | Tablet              | Phones & Tablet |
+-----------+---------------------+-----------------+
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.



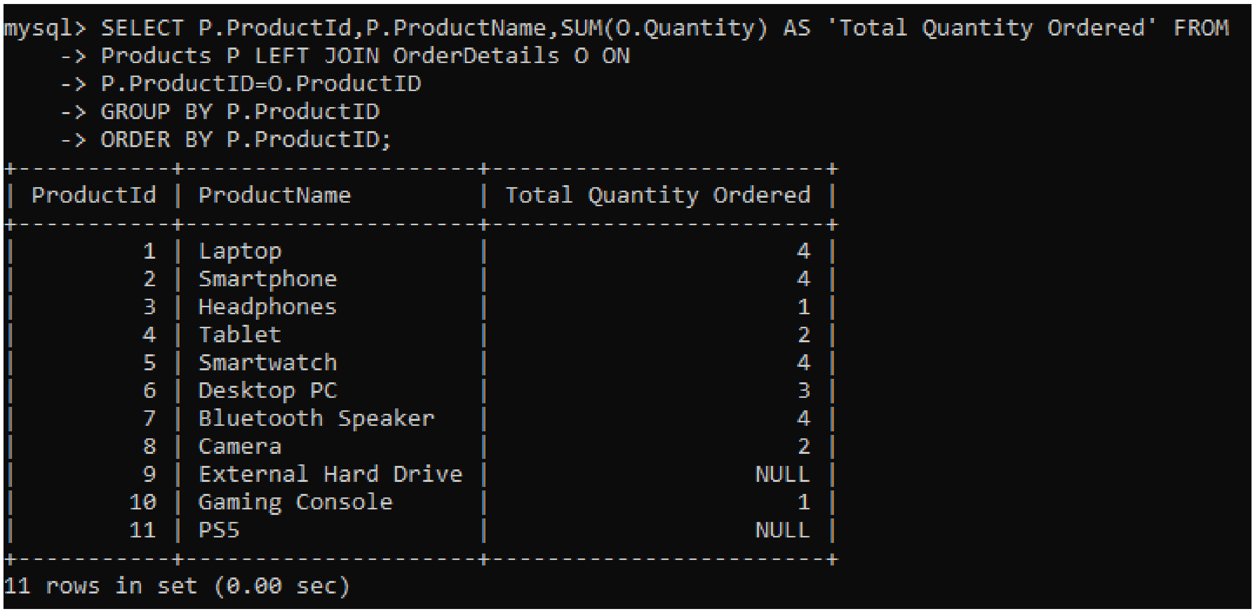
```
mysql> SELECT CONCAT(C.FirstName,' ',C.LastName) AS NAME, (O.Ordervalue/O.Visit) FROM
    -> Customers C JOIN (SELECT CustomerID,COUNT(CustomerID) AS Visit, SUM(TotalAmount) AS Ordervalue FROM Orders GROUP BY CustomerID) AS O
    -> ON C.CustomerID=O.CustomerID
    -> GROUP BY C.CustomerID;
+---------------+-----------------------+
| NAME          | (O.Ordervalue/O.Visit) |
+---------------+-----------------------+
| John Doe      |             4015.0000 |
| Jane Smith    |             5280.0000 |
| Alice Johnson |              165.0000 |
| Bob Williams  |              880.0000 |
| Eva Brown     |              660.0000 |
| Charlie Davis |             1320.0000 |
| Grace Miller  |              440.0000 |
| David Moore   |              220.0000 |
| Sophia Lee    |              570.0000 |
+---------------+-----------------------+
9 rows in set (0.00 sec)
```

7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.



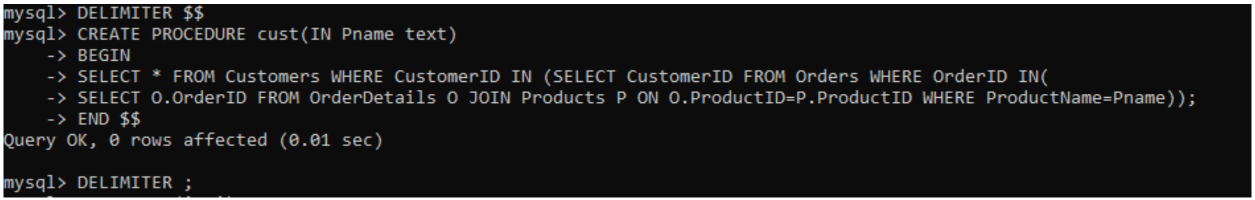
```
mysql> SELECT O.OrderID,C.CustomerID,C.FirstName,C.LastName,C.Phone,C.Email,C.Address,TotalAmount AS 'Total Order Value' FROM
    -> Orders O JOIN Customers C ON C.CustomerID=O.CustomerID
    -> GROUP BY OrderID
    -> ORDER BY TotalAmount DESC
    -> LIMIT 1;
+---------+------------+-----------+----------+------------+------------------------+------------+-------------------+
| OrderID | CustomerID | FirstName | LastName | Phone      | Email                  | Address    | Total Order Value |
+---------+------------+-----------+----------+------------+------------------------+------------+-------------------+
|       3 |          2 | Jane      | Smith    | 9876543210 | jane.smith@example.com | 456 Oak St |              5280 |
+---------+------------+-----------+----------+------------+------------------------+------------+-------------------+
1 row in set (0.00 sec)
```

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



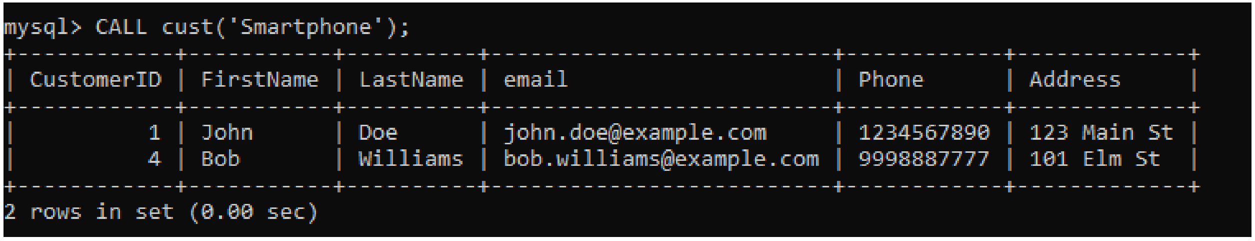
```
mysql> SELECT P.ProductId,P.ProductName,SUM(O.Quantity) AS 'Total Quantity Ordered' FROM
    -> Products P LEFT JOIN OrderDetails O ON
    -> P.ProductID=O.ProductID
    -> GROUP BY P.ProductID
    -> ORDER BY P.ProductID;
+-----------+---------------------+------------------------+
| ProductId | ProductName         | Total Quantity Ordered |
+-----------+---------------------+------------------------+
|         1 | Laptop              |                      4 |
|         2 | Smartphone          |                      4 |
|         3 | Headphones          |                      1 |
|         4 | Tablet              |                      2 |
|         5 | Smartwatch          |                      4 |
|         6 | Desktop PC          |                      3 |
|         7 | Bluetooth Speaker   |                      4 |
|         8 | Camera              |                      2 |
|         9 | External Hard Drive |                   NULL |
|        10 | Gaming Console      |                      1 |
|        11 | PS5                 |                   NULL |
+-----------+---------------------+------------------------+
11 rows in set (0.00 sec)
```

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



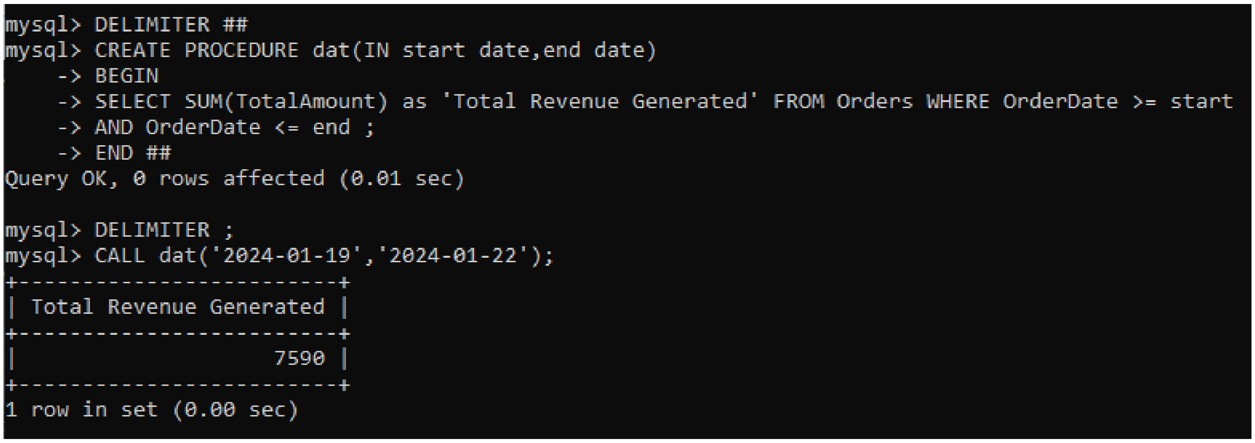
```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE cust(IN Pname text)
    -> BEGIN
    -> SELECT * FROM Customers WHERE CustomerID IN (SELECT CustomerID FROM Orders WHERE OrderID IN(
    -> SELECT O.OrderID FROM OrderDetails O JOIN Products P ON O.ProductID=P.ProductID WHERE ProductName=Pname));
    -> END $$
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
```



```
mysql> CALL cust('Smartphone');
+------------+-----------+----------+--------------------------+------------+-------------+
| CustomerID | FirstName | LastName | email                    | Phone      | Address     |
+------------+-----------+----------+--------------------------+------------+-------------+
|          1 | John      | Doe      | john.doe@example.com     | 1234567890 | 123 Main St |
|          4 | Bob       | Williams | bob.williams@example.com | 9998887777 | 101 Elm St  |
+------------+-----------+----------+--------------------------+------------+-------------+
2 rows in set (0.00 sec)
```

10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters



```
mysql> DELIMITER ##
mysql> CREATE PROCEDURE dat(IN start date,end date)
    -> BEGIN
    -> SELECT SUM(TotalAmount) as 'Total Revenue Generated' FROM Orders WHERE OrderDate >= start
    -> AND OrderDate <= end ;
    -> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> CALL dat('2024-01-19','2024-01-22');
+-------------------------+
| Total Revenue Generated |
+-------------------------+
|                    7590 |
+-------------------------+
1 row in set (0.00 sec)
```