Task 4. Subquery and its type:

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

```
mysql> SELECT * FROM Customers WHERE CustomerId NOT IN (SELECT CustomerID FROM Orders);
+------------+-----------+----------+---------------------------+------------+---------------+
| CustomerID | FirstName | LastName | email                     | Phone      | Address       |
+------------+-----------+----------+---------------------------+------------+---------------+
|          2 | Jane      | Smith    | jane.smith@example.com    | 9876543210 | 456 Oak St    |
|          3 | Alice     | Johnson  | alice.johnson@example.com | 5551234567 | 789 Pine St   |
|         10 | Oliver    | Anderson | oliver.anderson@example.com | 3334445555 | 707 Walnut St |
|         11 | Manas     | Rustagi  | manasrustagi123@gmail.com | 7982681438 | Karol Bagh    |
+------------+-----------+----------+---------------------------+------------+---------------+
4 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM Products
    -> WHERE ProductID IN (SELECT ProductID FROM Inventory WHERE QuantityInStock >0);
+-----------+---------------------+-----------------+-------+
| ProductID | ProductName         | Description     | Price |
+-----------+---------------------+-----------------+-------+
|         1 | Laptop              | PC              |  1320 |
|         2 | Smartphone          | Phones & Tablet |   880 |
|         3 | Headphones          | Accessories     |   165 |
|         4 | Tablet              | Phones & Tablet |   330 |
|         5 | Smartwatch          | Accessories     |   220 |
|         6 | Desktop PC          | PC              |  1650 |
|         7 | Bluetooth Speaker   | Accessories     |    55 |
|         8 | Camera              | Camera          |   660 |
|         9 | External Hard Drive | Accessories     |    88 |
|        10 | Gaming Console      | Console         |   440 |
+-----------+---------------------+-----------------+-------+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT SUM(TotalAmount) AS 'Total Revenue Generated' FROM Orders;
+--------------------------+
| Total Revenue Generated |
+--------------------------+
|                    12690 |
+--------------------------+
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.

```
mysql> DELIMITER @@
mysql> CREATE PROCEDURE cat(IN descript text)
    -> BEGIN
    -> SELECT (quantity*No_order) AS 'Average' FROM (SELECT SUM(Quantity) AS quantity,COUNT(OrderID) AS 'No_order' FROM OrderDetails
    -> WHERE ProductID IN (SELECT ProductID FROM Products WHERE Description=descript)) O;
    -> END @@
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> CALL cat('PC');
+---------+
| Average |
+---------+
|       3 |
+---------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 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.

```
mysql> DELIMITER ##
mysql> CREATE PROCEDURE data(IN id int)
    -> BEGIN
    -> SELECT SUM(TotalAmount) AS 'Total Revenue Generated By The Customer' FROM Orders WHERE
    -> CustomerID=id;
    -> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> CALL data(3);
+----------------------------------------+
| Total Revenue Generated By The Customer |
+----------------------------------------+
|                                   NULL |
+----------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL data(2);
+----------------------------------------+
| Total Revenue Generated By The Customer |
+----------------------------------------+
|                                   NULL |
+----------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL data(5);
+----------------------------------------+
| Total Revenue Generated By The Customer |
+----------------------------------------+
|                                    660 |
+----------------------------------------+
1 row in set (0.00 sec)

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

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

```
mysql> SELECT
    -> c.CustomerID,
    -> c.FirstName,
    -> 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;
+------------+-----------+-----------------+
| CustomerID | FirstName | NumberOfOrders |
+------------+-----------+-----------------+
|          1 | John      |               2 |
+------------+-----------+-----------------+
1 row in set (0.00 sec)
```

7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.

```
mysql> SELECT p.Category,SUM(od.Quantity) AS TotalQuantityOrdered FROM OrderDetails od JOIN Products p ON od.ProductID =
 p.ProductID GROUP BY p.Category ORDER BY TotalQuantityOrdered DESC LIMIT 1;
+-------------+----------------------+
| Category    | TotalQuantityOrdered |
+-------------+----------------------+
| Electronics |                    8 |
+-------------+----------------------+
1 row in set (0.01 sec)
```

8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.

```
mysql> SELECT c.CustomerID, c.FirstName, SUM(o.TotalAmount) AS TotalSpending FROM Customers c JOIN Orders o ON c.CustomerID =
 o.CustomerID JOIN OrderDetails od ON o.OrderID = od.OrderID JOIN Products p ON od.ProductID = p.ProductID WHERE p.Category =
 'Electronics' GROUP BY c.CustomerID ORDER BY TotalSpending DESC LIMIT 1;
+------------+-----------+---------------+
| CustomerID | FirstName | TotalSpending |
+------------+-----------+---------------+
|          1 | Aarav     |       1999.98 |
+------------+-----------+---------------+
1 row in set (0.01 sec)
```

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

```
mysql> SELECT c.CustomerID,c.FirstName,COUNT(o.OrderID) AS NumberOfOrders,AVG(o.TotalAmount) AS AverageOrderValue FROM Custom
ers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID GROUP BY c.CustomerID;
+------------+-----------+----------------+-------------------+
| CustomerID | FirstName | NumberOfOrders | AverageOrderValue |
+------------+-----------+----------------+-------------------+
|          1 | Aarav     |              2 |       1199.985000 |
|          2 | Isha      |              1 |       1199.980000 |
|          3 | Vikram    |              1 |       1599.980000 |
|          4 | Anaya     |              1 |        799.990000 |
|          5 | Raj       |              1 |        349.990000 |
|          6 | Aishwarya |              1 |        899.990000 |
|          7 | Arjun     |              0 |              NULL |
|          8 | Meera     |              1 |         79.990000 |
|          9 | Aditya    |              1 |         69.990000 |
|         10 | Kavya     |              1 |        499.990000 |
```

10. Write an SQL query to find the total number of orders placed by each customer and list their

```
mysql> SELECT c.CustomerID,c.FirstName,COUNT(o.OrderID) AS NumberOfOrders FROM Customers c LEFT JOIN Orders o ON c.CustomerID
 = o.CustomerID GROUP BY c.CustomerID;
+------------+-----------+----------------+
| CustomerID | FirstName | NumberOfOrders |
+------------+-----------+----------------+
|          1 | Aarav     |              2 |
|          2 | Isha      |              1 |
|          3 | Vikram    |              1 |
|          4 | Anaya     |              1 |
|          5 | Raj       |              1 |
|          6 | Aishwarya |              1 |
|          7 | Arjun     |              0 |
|          8 | Meera     |              1 |
|          9 | Aditya    |              1 |
|         10 | Kavya     |              1 |
```