1. Write SQL statements to create the tables as described in the database schema

```
4 • ⊖ create table Customers (
           CustomerID int primary key,
           FirstName varchar(50),
 6
           LastName varchar(50),
           Email varchar(100),
 8
           Phone varchar(20),
           RegistrationDate date
10
11
12
13 • ⊖ create table Products (
           ProductID int primary key,
14
15
           ProductName varchar(100),
16
           Category varchar(50),
           Price decimal(10, 2),
17
18
           StockQuantity int
19
20
```

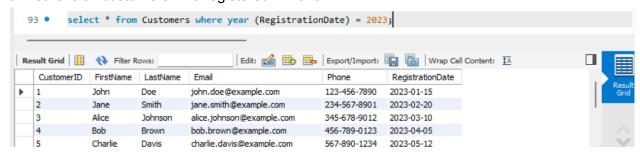
```
21 • ⊖ create table Orders (
           OrderID int primary key,
22
23
           OrderDate date,
           CustomerID int ,
24
25
           TotalAmount decimal(10, 2),
           foreign key (CustomerID) references Customers(CustomerID)
26
      ٠);
27
28 • ⊖ create table OrderDetails (
29
           OrderDetailID int primary key,
30
           OrderID int,
           ProductID int,
31
32
           Quantity int,
33
           Price decimal(10, 2),
           foreign key (OrderID) references Orders(OrderID),
34
           foreign key (ProductID) references Products(ProductID)
35
       );
36
```

2. Insert Data

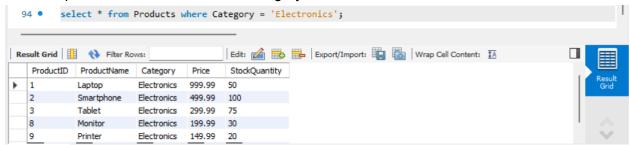
```
39 •
       insert into Customers
40
       values
           (1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '2023-01-15'),
41
42
           (2, 'Jane', 'Smith', 'jane.smith@example.com', '234-567-8901', '2023-02-20'),
43
           (3, 'Alice', 'Johnson', 'alice.johnson@example.com', '345-678-9012', '2023-03-10'),
           (4, 'Bob', 'Brown', 'bob.brown@example.com', '456-789-0123', '2023-04-05'),
44
45
           (5, 'Charlie', 'Davis', 'charlie.davis@example.com', '567-890-1234', '2023-05-12'),
           (6, 'David', 'Wilson', 'david.wilson@example.com', '678-901-2345', '2023-06-15'),
46
47
           (7,'Emma', 'Thomas', 'emma.thomas@example.com', '789-012-3456', '2023-07-01'),
           (8, 'Fiona', 'Garcia', 'fiona.garcia@example.com', '890-123-4567', '2023-07-10'),
48
49
           (9, 'George', 'Martinez', 'george.martinez@example.com', '901-234-5678', '2023-07-20'),
           (10, 'Hannah', 'Rodriguez', 'hannah.rodriguez@example.com', '012-345-6789', '2023-07-25');
50
```

```
53 •
        insert into Products
54
        values
55
            (1, 'Laptop', 'Electronics', 999.99, 50),
            (2, 'Smartphone', 'Electronics', 499.99, 100),
56
57
            (3, 'Tablet', 'Electronics', 299.99, 75),
            (4, 'Headphones', 'Accessories', 49.99, 200),
58
59
            (5, 'Charger', 'Accessories', 19.99, 300),
            (6, 'Keyboard', 'Accessories', 29.99, 150),
60
61
            (7, 'Mouse', 'Accessories', 19.99, 250),
            (8, 'Monitor', 'Electronics', 199.99, 30),
62
            (9, 'Printer', 'Electronics', 149.99, 20),
63
            (10, 'USB Cable', 'Accessories', 9.99, 400);
64
67
        insert into Orders
68
        values
69
            (1,'2023-06-01', 1, 1049.98),
            (2, 2023-06-05', 2, 549.98),
70
71
            (3,'2023-06-10', 3, 999.99),
            (4,'2023-06-15', 4, 69.98),
72
73
            (5,'2023-06-20', 5, 519.98),
74
            (6, '2023-06-25', 6, 229.98),
75
            (7,'2023-07-02', 7, 1199.97),
            (8,'2023-07-12', 8, 49.98),
76
77
            (9,'2023-07-18', 9, 349.98),
            (10, '2023-07-22', 10, 39.98);
78
81 •
       insert into OrderDetails
82
       values
           (1,1, 1, 1, 999.99), (2,1, 4, 1, 49.99),
83
           (3,2, 2, 1, 499.99), (4,2, 5, 1, 49.99),
           (5,3, 1, 1, 999.99), (6,4, 4, 1, 49.99),
85
86
           (7,4, 5, 1, 19.99), (8,5, 2, 1, 499.99),
           (9,5, 5, 1, 19.99), (10,6, 3, 1, 199.99),
87
88
           (11,6, 5, 1, 29.99), (12,7, 1, 1, 999.99),
89
           (13,7, 3, 1, 199.99), (14,8, 7, 1, 19.99),
90
            (15,8, 8, 1, 29.99), (16,9, 4, 2, 149.99),
91
           (17,9, 9, 1, 49.99), (18,10, 10, 4, 9.99);
```

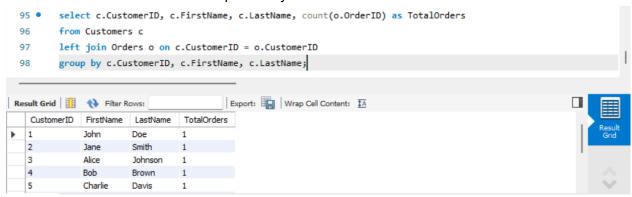
3. Retrieve all customers who registered in 2023.



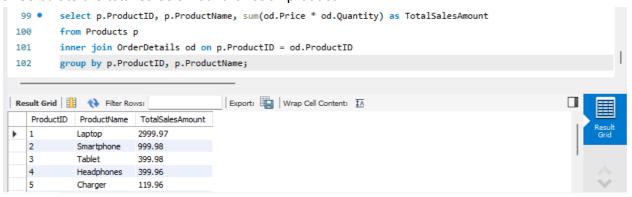
4. List all products in the 'Electronics' category.



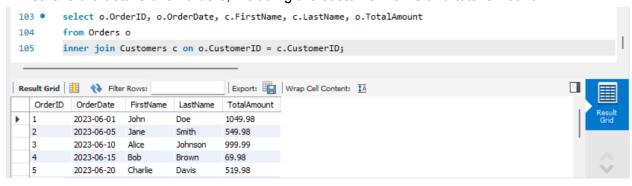
5. Find the total number of orders placed by each customer.



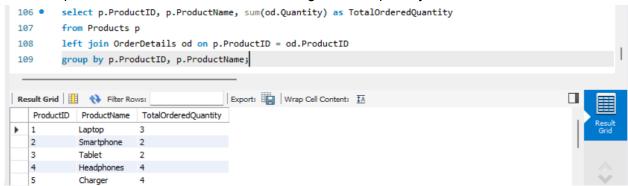
6. Calculate the total sales amount for each product.



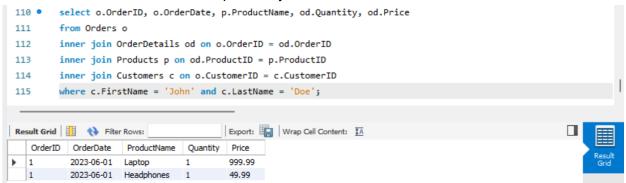
7. Retrieve the details of all orders, including the customer name and total amount.



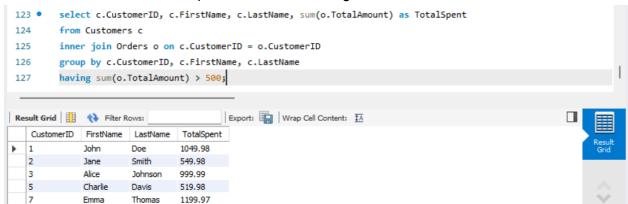
8. List all products that have been ordered along with the quantity ordered for each.



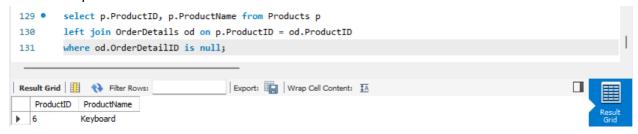
9. Find the order details for orders placed by 'John Doe'.



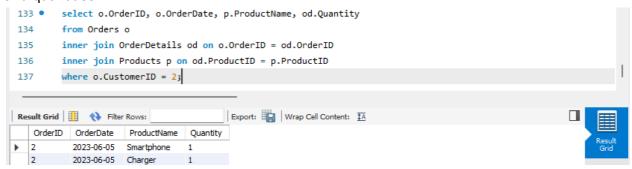
10. Find customers who have placed an order totaling more than \$500.



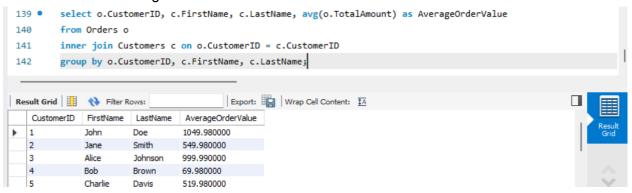
11. List the products that have never been ordered.



12. Retrieve the order history for a specific customer, including order date, product names, and quantities.

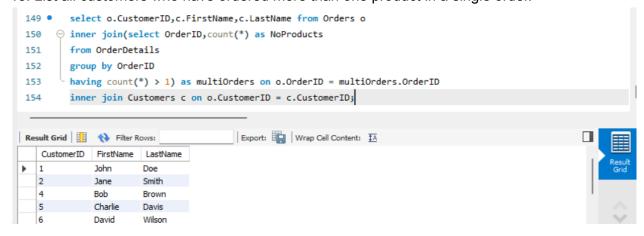


13. Calculate the average order value for each customer.

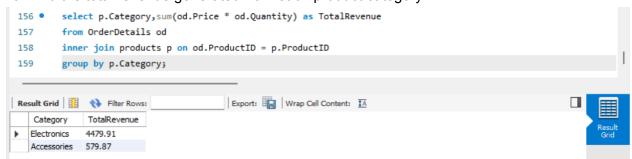


14. Find the most popular product category based on the number of orders

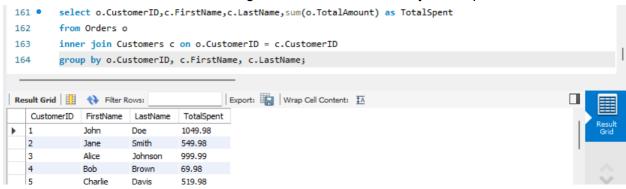
15. List all customers who have ordered more than one product in a single order.



16. Find the total revenue generated from each product category.



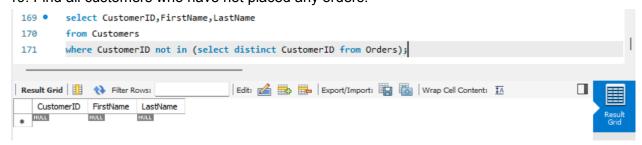
17. Retrieve the list of customers along with the total amount they have spent.



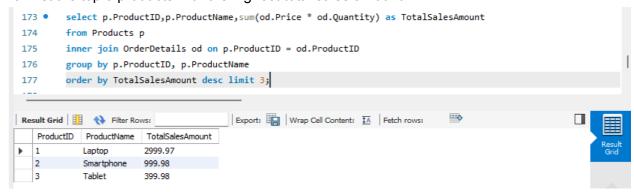
18. Find the average price of products in each category.



19. Find all customers who have not placed any orders.



20. List the top 3 products with the highest total sales amount.



21. Find customers who have placed orders for more than 3 different products.

