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

```sql
create table Customers (
    CustomerID int primary key,
    FirstName varchar(50),
    LastName varchar(50),
    Email varchar(100),
    Phone varchar(20),
    RegistrationDate date
);

create table Products (
    ProductID int primary key,
    ProductName varchar(100),
    Category varchar(50),
    Price decimal(10, 2),
    StockQuantity int
);
```

```sql
create table  Orders (
    OrderID int primary key,
    OrderDate date ,
    CustomerID int ,
    TotalAmount decimal(10, 2),
    foreign key (CustomerID) references Customers(CustomerID)
);
create table OrderDetails (
    OrderDetailID int primary key,
    OrderID int,
    ProductID int,
    Quantity int,
    Price decimal(10, 2),
    foreign key (OrderID) references Orders(OrderID),
    foreign key (ProductID) references Products(ProductID)
);
```

2. Insert Data

```sql
insert into Customers
values
    (1,'John', 'Doe', 'john.doe@example.com', '123-456-7890', '2023-01-15'),
    (2,'Jane', 'Smith', 'jane.smith@example.com', '234-567-8901', '2023-02-20'),
    (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'),
    (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'),
    (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'),
    (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');
```

```sql
insert into Products
values
    (1,'Laptop', 'Electronics', 999.99, 50),
    (2,'Smartphone', 'Electronics', 499.99, 100),
    (3,'Tablet', 'Electronics', 299.99, 75),
    (4,'Headphones', 'Accessories', 49.99, 200),
    (5,'Charger', 'Accessories', 19.99, 300),
    (6,'Keyboard', 'Accessories', 29.99, 150),
    (7,'Mouse', 'Accessories', 19.99, 250),
    (8,'Monitor', 'Electronics', 199.99, 30),
    (9,'Printer', 'Electronics', 149.99, 20),
    (10,'USB Cable', 'Accessories', 9.99, 400);
```

```sql
insert into Orders
values
    (1,'2023-06-01', 1, 1049.98),
    (2,'2023-06-05', 2, 549.98),
    (3,'2023-06-10', 3, 999.99),
    (4,'2023-06-15', 4, 69.98),
    (5,'2023-06-20', 5, 519.98),
    (6,'2023-06-25', 6, 229.98),
    (7,'2023-07-02', 7, 1199.97),
    (8,'2023-07-12', 8, 49.98),
    (9,'2023-07-18', 9, 349.98),
    (10,'2023-07-22', 10, 39.98);
```

```sql
insert into OrderDetails
values
    (1,1, 1, 1, 999.99), (2,1, 4, 1, 49.99),
    (3,2, 2, 1, 499.99), (4,2, 5, 1, 49.99),
    (5,3, 1, 1, 999.99), (6,4, 4, 1, 49.99),
    (7,4, 5, 1, 19.99), (8,5, 2, 1, 499.99),
    (9,5, 5, 1, 19.99), (10,6, 3, 1, 199.99),
    (11,6, 5, 1, 29.99), (12,7, 1, 1, 999.99),
    (13,7, 3, 1, 199.99), (14,8, 7, 1, 19.99),
    (15,8, 8, 1, 29.99), (16,9, 4, 2, 149.99),
    (17,9, 9, 1, 49.99), (18,10, 10, 4, 9.99);
```

3. Retrieve all customers who registered in 2023.

```sql
select * from Customers where year (RegistrationDate) = 2023;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	FirstName	LastName	Email	Phone	RegistrationDate
1	John	Doe	john.doe@example.com	123-456-7890	2023-01-15
2	Jane	Smith	jane.smith@example.com	234-567-8901	2023-02-20
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
5	Charlie	Davis	charlie.davis@example.com	567-890-1234	2023-05-12

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

```
94 •  select * from Products where Category = 'Electronics';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ProductID	ProductName	Category	Price	StockQuantity
1	Laptop	Electronics	999.99	50
2	Smartphone	Electronics	499.99	100
3	Tablet	Electronics	299.99	75
8	Monitor	Electronics	199.99	30
9	Printer	Electronics	149.99	20

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

```
95 •  select c.CustomerID, c.FirstName, c.LastName, count(o.OrderID) as TotalOrders
96    from Customers c
97    left join Orders o on c.CustomerID = o.CustomerID
98    group by c.CustomerID, c.FirstName, c.LastName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	FirstName	LastName	TotalOrders
1	John	Doe	1
2	Jane	Smith	1
3	Alice	Johnson	1
4	Bob	Brown	1
5	Charlie	Davis	1

6. Calculate the total sales amount for each product.

```
99 •  select p.ProductID, p.ProductName, sum(od.Price * od.Quantity) as TotalSalesAmount
100   from Products p
101   inner join OrderDetails od on p.ProductID = od.ProductID
102   group by p.ProductID, p.ProductName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductID	ProductName	TotalSalesAmount
1	Laptop	2999.97
2	Smartphone	999.98
3	Tablet	399.98
4	Headphones	399.96
5	Charger	119.96

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

```
103 •  select o.OrderID, o.OrderDate, c.FirstName, c.LastName, o.TotalAmount
104    from Orders o
105    inner join Customers c on o.CustomerID = c.CustomerID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

OrderID	OrderDate	FirstName	LastName	TotalAmount
1	2023-06-01	John	Doe	1049.98
2	2023-06-05	Jane	Smith	549.98
3	2023-06-10	Alice	Johnson	999.99
4	2023-06-15	Bob	Brown	69.98
5	2023-06-20	Charlie	Davis	519.98

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

```
106 •  select p.ProductID, p.ProductName, sum(od.Quantity) as TotalOrderedQuantity
107    from Products p
108    left join OrderDetails od on p.ProductID = od.ProductID
109    group by p.ProductID, p.ProductName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductID	ProductName	TotalOrderedQuantity
1	Laptop	3
2	Smartphone	2
3	Tablet	2
4	Headphones	4
5	Charger	4

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

```
110 •  select o.OrderID, o.OrderDate, p.ProductName, od.Quantity, od.Price
111    from Orders o
112    inner join OrderDetails od on o.OrderID = od.OrderID
113    inner join Products p on od.ProductID = p.ProductID
114    inner join Customers c on o.CustomerID = c.CustomerID
115    where c.FirstName = 'John' and c.LastName = 'Doe';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

OrderID	OrderDate	ProductName	Quantity	Price
1	2023-06-01	Laptop	1	999.99
1	2023-06-01	Headphones	1	49.99

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

```
123 •  select c.CustomerID, c.FirstName, c.LastName, sum(o.TotalAmount) as TotalSpent
124    from Customers c
125    inner join Orders o on c.CustomerID = o.CustomerID
126    group by c.CustomerID, c.FirstName, c.LastName
127    having sum(o.TotalAmount) > 500;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	FirstName	LastName	TotalSpent
1	John	Doe	1049.98
2	Jane	Smith	549.98
3	Alice	Johnson	999.99
5	Charlie	Davis	519.98
7	Emma	Thomas	1199.97

11. List the products that have never been ordered.

```
129 •  select p.ProductID, p.ProductName from Products p
130    left join OrderDetails od on p.ProductID = od.ProductID
131    where od.OrderDetailID is null;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductID	ProductName
6	Keyboard

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

```
133 •  select o.OrderID, o.OrderDate, p.ProductName, od.Quantity
134    from Orders o
135    inner join OrderDetails od on o.OrderID = od.OrderID
136    inner join Products p on od.ProductID = p.ProductID
137    where o.CustomerID = 2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

OrderID	OrderDate	ProductName	Quantity
2	2023-06-05	Smartphone	1
2	2023-06-05	Charger	1

13. Calculate the average order value for each customer.

```
139 •  select o.CustomerID, c.FirstName, c.LastName, avg(o.TotalAmount) as AverageOrderValue
140    from Orders o
141    inner join Customers c on o.CustomerID = c.CustomerID
142    group by o.CustomerID, c.FirstName, c.LastName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	FirstName	LastName	AverageOrderValue
1	John	Doe	1049.980000
2	Jane	Smith	549.980000
3	Alice	Johnson	999.990000
4	Bob	Brown	69.980000
5	Charlie	Davis	519.980000

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

```
144 •  select p.Category, count(*) as NumOrders
145    from OrderDetails od
146    inner join Products p on od.ProductID = p.ProductID
147    group by p.Category order by NumOrders desc limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Category	NumOrders
Electronics	9

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

```
149 •  select o.CustomerID,c.FirstName,c.LastName from Orders o
150    inner join(select OrderID,count(*) as NoProducts
151    from OrderDetails
152    group by OrderID
153    having count(*) > 1) as multiOrders on o.OrderID = multiOrders.OrderID
154    inner join Customers c on o.CustomerID = c.CustomerID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	FirstName	LastName
1	John	Doe
2	Jane	Smith
4	Bob	Brown
5	Charlie	Davis
6	David	Wilson

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

```
156 •  select p.Category,sum(od.Price * od.Quantity) as TotalRevenue
157    from OrderDetails od
158    inner join products p on od.ProductID = p.ProductID
159    group by p.Category;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Category	TotalRevenue
Electronics	4479.91
Accessories	579.87

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

```
161 •  select o.CustomerID,c.FirstName,c.LastName,sum(o.TotalAmount) as TotalSpent
162    from Orders o
163    inner join Customers c on o.CustomerID = c.CustomerID
164    group by o.CustomerID, c.FirstName, c.LastName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	FirstName	LastName	TotalSpent
1	John	Doe	1049.98
2	Jane	Smith	549.98
3	Alice	Johnson	999.99
4	Bob	Brown	69.98
5	Charlie	Davis	519.98

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

```
166 •  select Category,avg(Price) as AvgPrice
167    from Products group by Category;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Category	AvgPrice
Electronics	429.990000
Accessories	25.990000

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

```
169 •  select CustomerID,FirstName,LastName
170    from Customers
171    where CustomerID not in (select distinct CustomerID from Orders);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	FirstName	LastName
NULL	NULL	NULL

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

```
173 •  select p.ProductID,p.ProductName,sum(od.Price * od.Quantity) as TotalSalesAmount
174    from Products p
175    inner join OrderDetails od on p.ProductID = od.ProductID
176    group by p.ProductID, p.ProductName
177    order by TotalSalesAmount desc limit 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ProductID	ProductName	TotalSalesAmount
1	Laptop	2999.97
2	Smartphone	999.98
3	Tablet	399.98

Result Grid

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

```
179 •  select o.CustomerID,c.FirstName,c.LastName
180    from Orders o
181    inner join(select OrderID,count(distinct ProductID) as NumProducts
182    from OrderDetails
183    group by OrderID
184    having count(distinct ProductID) > 3)
185    as multiorders on o.OrderID = multiorders.OrderID
186    inner join Customers c on o.CustomerID = c.CustomerID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	FirstName	LastName

Result Grid