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.FirstName, c.LastName, c.Email, c.Phone,
    c.Address,o.OrderDate, o.TotalAmount

    FROM Orders o

    JOIN Customers c ON o.CustomerID = c.CustomerID

    ORDER BY o.OrderDate ;
```

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(o.Quantity * p.Price) AS TotalRevenue

    FROM OrderDetails od

    JOIN Products p ON o.ProductID = p.ProductID

    GROUP BY p.ProductID, p.ProductName

    ORDER BY TotalRevenue DESC;
```

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

```
    SELECT DISTINCT c.* from Customers c JOIN Orders o ON
    c.CustomerID=o.CustomerID;
```

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(o.Quantity) as TotalQuantity

FROM OrderDetails o

JOIN Products p ON o.ProductID = p.ProductID

GROUP BY p.ProductID, p.ProductName

ORDER BY TotalQuantity DESC;

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

CREATE TABLE Categories (

-> CategoryID INT PRIMARY KEY AUTO_INCREMENT,

-> CategoryName VARCHAR(100)

->);

INSERT INTO Categories (CategoryName)
VALUES('Laptops','Smartphones','Tablets','Accessories');

SELECT p.ProductName, c.CategoryName FROM Products p JOIN Categories c ON p.CategoryID = c.CategoryID;

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.CustomerID, c.FirstName, c.LastName, AVG(o.TotalAmount) AS AverageOrderValue FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID GROUP BY c.CustomerID, c.FirstName, c.LastName;

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.CustomerID, c.FirstName, c.LastName, c.Email,
c.Phone, o.TotalAmount FROM Orders o JOIN Customers c ON
o.CustomerID = c.CustomerID WHERE o.TotalAmount = (SELECT
MAX(TotalAmount) FROM Orders);
```

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

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

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.* 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 = 'Laptop';
```

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(TotalAmount) from Orders where OrderDate BETWEEN
'2025-03-01'  AND '2025-03-10';
```