```
#all columns
select * from superstore;

#how many columns
select count(*) from superstore;

#Find customers from selected regions

SELECT DISTINCT CustomerName,region
FROM superstore
WHERE region IN ('West', 'East');

#Customers NOT from selected regions

SELECT customername, region
FROM superstore
WHERE region NOT IN ('West','East');

#Aggregation Functions (SUM, AVG, COUNT, MIN, MAX)
SELECT
   COUNT(DISTINCT orderid) AS total_orders,
   ROUND(SUM(sales),2) AS total_sales,
   ROUND(AVG(profit),2) AS avg_profit,
   MAX(sales) AS highest_order_value,
   MIN(sales) AS lowest_order_value
FROM superstore;
```

#Top profitable orders

```
SELECT orderid, sales, profit
FROM superstore
ORDER BY profit DESC;
```

#Customers whose name starts with ‘A’

```
SELECT DISTINCT customername
FROM superstore
WHERE customername LIKE 'a%';
```

#product sub-categories containing 'chair'

```
select distinct productsubcategory from
superstore where productsubcategory like '%chair%';
```

#cost estimation sales-profit

```
select orderid,sales,Profit,(sales-Profit)
as cost_estimation from superstore where Profit>0 and sales>100;
```

#orders with sales between 500 and 1000

```
select OrderID,sales from superstore where sales between 500 and 1000;
```

#orders with sales higher than all office suppliers sales

```
select OrderID,sales from superstore where
sales > all (select sales from superstore where productcategory = officesuppliers );
```

```
#customer with above average sales
select CustomerName,sales from superstore where sales > (select avg(sales) from
superstore);

#resuable  monthly sales view
create view monthlysales_view as select month(OrderDate)
as month,round(sum(sales),2) as total_sales from superstore group by month(OrderDate);
select* from monthlysales_view;

#string functions
select CustomerName,upper(customername),
        lower(customername),
        length(customername),
        substring(customername,1,5) from superstore;

#second highest salary using sub query
select max(sales) as second_highest from superstore where
 sales<(select max(sales) from superstore);

#total sales & total profit
select round(sum(sales),2) as total_sales,
      round(sum(profit),2) as total_profit from superstore;

# sales by product category
select ProductCategory,round(sum(sales),2) from superstore
 group by Productcategory order by Productcategory desc;
```

```
#profit by product subcategory
select productsubcategory,round(sum(profit),2) as subcategory from superstore
 group by productsubcategory order by productsubcategory desc;

# top 10 customer by sales
select customername,round(sum(sales),2) as totalsales from superstore
group by customername order by customername desc limit 10;

#sales by city
select city,sum(sales) as total_sales from superstore group by city order by city desc;

#profit analysis by product
select  ProductName,sum(sales) as sum_sales,
sum(Profit) as sum_profit from superstore group by ProductName order by sum_sales desc;


#regional sales and profit performance
select region ,round(sum(sales),2) as totalsales,round(sum(profit),2) as totalprofit from
superstore group by region order by totalsales desc;

#Monthly Sales Trend
SELECT
   orderdate AS month,
   ROUND(SUM(sales),2) AS monthly_sales
FROM superstore
GROUP BY orderdate
ORDER BY month;
```

```
#Discount Impact on Profit
SELECT
   discount,
   ROUND(AVG(profit),2) AS avg_profit
FROM superstore
GROUP BY discount
ORDER BY avg_profit;

#Loss-Making Orders
SELECT
   orderid,
   sales,
   profit
FROM superstore
WHERE profit < 0;

#Average Delivery Time (Shipping Analysis)
SELECT
   ROUND(AVG(DATEDIFF(shipdate, orderdate)),2) AS avg_delivery_days
FROM superstore;

#Customer Segment Performance
```

```
SELECT
   customersegment,
   ROUND(SUM(sales),2) AS total_sales,
   ROUND(SUM(profit),2) AS total_profit
FROM superstore
GROUP BY customersegment;



#Year-wise Sales & Profit Trend
# is the bussiness growing year over year?

SELECT
   YEAR(orderdate) AS year,
   ROUND(SUM(sales),2) AS total_sales,
   ROUND(SUM(profit),2) AS total_profit
FROM superstore
GROUP BY YEAR(orderdate)
ORDER BY year;

#monthwise sales and profit
SELECT
   month(orderdate) AS month,
   ROUND(SUM(sales),2) AS total_sales,
   ROUND(SUM(profit),2) AS total_profit
FROM superstore
GROUP BY month(orderdate)
ORDER BY month;
```

```
#day wise sales
SELECT
   day(orderdate) AS day,
   ROUND(SUM(sales),2) AS total_sales,
   ROUND(SUM(profit),2) AS total_profit
FROM superstore
GROUP BY day(orderdate)
ORDER BY day;


#quarter wise sales
SELECT
   quarter(orderdate) AS quarter,
   ROUND(SUM(sales),2) AS total_sales,
   ROUND(SUM(profit),2) AS total_profit
FROM superstore
GROUP BY quarter(orderdate)
ORDER BY quarter;
#orderdate from superstore
select orderdate from superstore;

#Top 5 Most Profitable Products (Sub-Category)
#which products drive maximum profit
SELECT
   productsubcategory,
   ROUND(SUM(profit),2) AS total_profit
FROM superstore
GROUP BY productsubcategory
```

```
ORDER BY total_profit DESC
LIMIT 5;

#bottom 5 loss-making sub_category
#where is the business losing money?
select productsubcategory,round(sum(profit),2) as total_loss
from superstore group by productsubcategory
having total_loss<4
order by total_loss;

#Average Order Value (AOV)
#how much does a customer spend per order
SELECT
   ROUND(SUM(quantityorderednew*unitprice) / COUNT(DISTINCT orderid), 2) AS
avg_order_value
FROM superstore;

select month(orderdate) as month,
round(sum(sales), 2) as monthly_sales from superstore
group by month order by month;


#Customer Repeat Purchase Analysis
#are customer comming back?
SELECT
   customername,
   COUNT(DISTINCT orderid) AS total_orders,
   ROUND(SUM(sales),2) AS total_sales
```

```
FROM superstore
GROUP BY customername
HAVING total_orders >1
ORDER BY total_orders DESC;

#High Discount vs Profit Impact
#is giving more discount reducing profit?
SELECT
   CASE
      WHEN discount = 0 THEN 'No Discount'
      WHEN discount BETWEEN 0.01 AND 0.20 THEN 'Low Discount'
      WHEN discount BETWEEN 0.21 AND 0.50 THEN 'Medium Discount'
      ELSE 'High Discount'
   END AS discount_range,
   ROUND(SUM(sales),2) AS total_sales,
   ROUND(SUM(profit),2) AS total_profit
FROM superstore
GROUP BY discount_range
ORDER BY total_profit DESC;

#Region-wise avg Delivery Time
#which region have slow delivery

SELECT
   region,
   ROUND(avg(DATEDIFF(shipdate, orderdate)),2) AS avg_delivery_days
FROM superstore
GROUP BY region
```

```
ORDER BY avg_delivery_days DESC;


#Category Contribution % to Total Sales
#which category contributes most to revenue?
SELECT
   productcategory,
   ROUND(SUM(sales),2) AS category_sales,
   ROUND(
      (SUM(sales) / (SELECT SUM(sales) FROM superstore)) * 100, 2
   ) AS sales_percentage
FROM superstore
GROUP BY ProductCategory
ORDER BY sales_percentage DESC;

 # Customer Segment Profitability
 #which customer segment is most profitable?

 SELECT
   customersegment,
   ROUND(SUM(sales),2) AS total_sales,
   ROUND(SUM(profit),2) AS total_profit,
   ROUND(AVG(profit),2) AS avg_profit_per_order
FROM superstore
GROUP BY customersegment;




# Orders with High Sales but Low/Negative Profit
```

```
#pricing or discount issue detection
SELECT
   orderid,
   sales,
   profit,
   discount
FROM superstore
WHERE sales > 500 AND profit <= 0
ORDER BY sales DESC;


#Top 3 Cities by Profit in Each Region
#location based strategy
SELECT
   region,
   city,
   ROUND(SUM(profit),2) AS total_profit
FROM superstore
GROUP BY region, city
ORDER BY region, total_profit DESC;


#High-Value Customers (Pareto 80/20 Rule)
#top customer contributing majority of revenue
SELECT
   customername,
   ROUND(SUM(sales),2) AS total_sales
FROM superstore
```

```
GROUP BY CustomerName
ORDER BY total_sales DESC
LIMIT 20;

#Shipping Delay Impact on Profit
#does delayed shipping reduce profit?
SELECT
   DATEDIFF(shipdate, orderdate) AS delivery_days,
   ROUND(AVG(profit),2) AS avg_profit
FROM superstore
GROUP BY delivery_days
ORDER BY delivery_days;

#Product Demand Analysis (Quantity Sold)
#which product sell the most
SELECT
   ProductSubCategory,
   SUM(quantityorderednew) AS total_quantity_sold
FROM superstore
GROUP BY ProductSubCategory
ORDER BY total_quantity_sold DESC;


#Rank Customers by Total Sales
#who are the top revenue-generating customers?
SELECT
   customername,
   ROUND(SUM(sales),2) AS total_sales,
```

```
    RANK() OVER (ORDER BY SUM(sales) DESC) AS sales_rank
FROM superstore
GROUP BY customername;



#Running Total of Sales Over Time
#cumulative revenue growth
SELECT
    orderdate,
    ROUND(SUM(sales),2) AS daily_sales,
    ROUND(SUM(SUM(sales)) OVER (ORDER BY orderdate),2) AS running_total_sales
FROM superstore
GROUP BY orderdate
ORDER BY orderdate;

#Rank Regions by Profit
#which region is most profitable?
SELECT
    region,
    ROUND(SUM(profit),2) AS total_profit,
    DENSE_RANK() OVER (ORDER BY SUM(profit) DESC) AS profit_rank
FROM superstore
GROUP BY region;

#Customer Lifetime Value (CLV)
#total value generated by each customer
SELECT
```

```
    customername,
    ROUND(SUM(sales) OVER (PARTITION BY customername),2) AS lifetime_sales,
    ROUND(SUM(profit) OVER (PARTITION BY customername),2) AS lifetime_profit
FROM superstore;

#Identify First & Latest Purchase of Each Customer
#customer journey analysis
SELECT
    customername,
   orderdate,
    ROW_NUMBER() OVER (PARTITION BY customername ORDER BY orderdate) AS
purchase_order
FROM superstore;

#Average Sales per Order vs Overall Average
#compare order performance
SELECT
    orderid,
    ROUND(SUM(sales),2) AS order_sales,
    ROUND(AVG(SUM(sales)) OVER (),2) AS overall_avg_sales
FROM superstore
GROUP BY orderid;

#Profit Difference Between Consecutive Orders (LEAD)
#profit volatility analysis

SELECT
    orderdate,
```

```
    profit,
    LEAD(profit) OVER (ORDER BY orderdate) AS next_order_profit,
    ROUND(LEAD(profit) OVER (ORDER BY orderdate) - profit,2) AS profit_change
FROM superstore;



# Identify Orders with Negative Profit and Their Characteristics
SELECT
    OrderID,
    CustomerName,
    CustomerSegment,
    ProductCategory,
    ProductSubCategory,
    ProductName,
    Quantityorderednew,
    Sales,
    Profit,
    Discount,
    ShippingCost,
    CASE
        WHEN Profit < 0 THEN 'Loss'
        ELSE 'Profit'
    END AS ProfitStatus,
    CASE
        WHEN Discount > 0.05 THEN 'High Discount'
        WHEN Discount > 0.02 THEN 'Medium Discount'
        ELSE 'Low/No Discount'
    END AS DiscountLevel
```

```
FROM superstore
WHERE Profit < 0
ORDER BY Profit ASC
LIMIT 15;

#Shipping Mode Analysis with Cost Efficiency
SELECT
   ShipMode,
   COUNT(*) AS TotalShipments,
   ROUND(AVG(DATEDIFF(ShipDate, OrderDate)), 2) AS AvgDeliveryDays,
   ROUND(SUM(ShippingCost), 2) AS TotalShippingCost,
   ROUND(AVG(ShippingCost), 2) AS AvgShippingCostPerOrder,
   ROUND(SUM(Sales), 2) AS TotalSales,
   ROUND(SUM(Profit), 2) AS TotalProfit,
   ROUND(SUM(ShippingCost) / SUM(Sales) * 100, 2) AS ShippingCostToSalesRatio
FROM superstore
GROUP BY ShipMode
ORDER BY TotalShipments DESC;

# Product Performance Analysis
SELECT
   ProductCategory,
   ProductSubCategory,
   ProductName,
   COUNT(*) AS TimesOrdered,
   SUM(Quantityorderednew) AS TotalUnitsSold,
   ROUND(SUM(Sales), 2) AS TotalRevenue,
   ROUND(SUM(Profit), 2) AS TotalProfit,
```

```
    ROUND(AVG(Discount) * 100, 2) AS AvgDiscountPercentage,
    ROUND(SUM(Profit) / SUM(Quantityorderednew), 2) AS ProfitPerUnit,
    CASE
        WHEN SUM(Profit) > 0 THEN 'Profitable'
        ELSE 'Loss-Making'
    END AS ProfitabilityStatus
FROM superstore
GROUP BY ProductCategory, ProductSubCategory, ProductName
ORDER BY TotalProfit DESC
LIMIT 15;

 #Discount Effectiveness Analysis
 SELECT
    CASE
        WHEN Discount = 0 THEN 'No Discount'
        WHEN Discount <= 0.05 THEN 'Low Discount (1-5%)'
        WHEN Discount <= 0.10 THEN 'Medium Discount (6-10%)'
        ELSE 'High Discount (>10%)'
    END AS DiscountTier,
    COUNT(*) AS TotalOrders,
    ROUND(AVG(Discount) * 100, 2) AS AvgDiscountPercentage,
    ROUND(SUM(Sales), 2) AS TotalSales,
    ROUND(SUM(Profit), 2) AS TotalProfit,
    ROUND(AVG(Sales), 2) AS AvgOrderValue,
    ROUND(AVG(Quantityorderednew), 2) AS AvgQuantityPerOrder,
    ROUND(SUM(Profit) / SUM(Sales) * 100, 2) AS ProfitMarginPercentage,
    ROUND(SUM(CASE WHEN Profit < 0 THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) AS
LossOrderPercentage
```

```
FROM superstore
GROUP BY DiscountTier
ORDER BY CASE DiscountTier
    WHEN 'No Discount' THEN 1
    WHEN 'Low Discount (1-5%)' THEN 2
    WHEN 'Medium Discount (6-10%)' THEN 3
    ELSE 4
END;

#Products That Have Never Made a Loss
SELECT
    ProductCategory,
    ProductSubCategory,
    ProductName,
    COUNT(*) AS TimesOrdered,
    ROUND(SUM(Profit), 2) AS TotalProfit
FROM superstore
WHERE ProductName NOT IN (
    SELECT DISTINCT ProductName
    FROM superstore
    WHERE Profit < 0
)
GROUP BY ProductCategory, ProductSubCategory, ProductName
ORDER BY TotalProfit DESC;

#Find Most Expensive Order in Each Region
SELECT
    s1.Region,
```

```
    s1.OrderID,
    s1.CustomerName,
    s1.sales AS OrderTotal,
    s1.OrderDate
FROM superstore s1
WHERE s1.Sales = (
    SELECT MAX(s2.Sales)
    FROM superstore s2
    WHERE s2.Region = s1.Region
)
ORDER BY s1.Sales DESC;



#Customers Who Bought Only Once (One-Time Buyers)
SELECT
    CustomerID,
    CustomerName,
    CustomerSegment,
    OrderDate AS PurchaseDate,
    Sales AS PurchaseAmount
FROM superstore s1
WHERE CustomerID IN (
    SELECT CustomerID
    FROM superstore
    GROUP BY CustomerID
    HAVING COUNT(DISTINCT OrderID) = 1
)
ORDER BY PurchaseAmount DESC;
```

```
#Products With Above-Average Discount but Below-Average Sales
SELECT
    ProductName,
    ROUND(AVG(Discount) * 100, 2) AS AvgDiscountPercentage,
    ROUND(AVG(Sales), 2) AS AvgSalesPerOrder,
    COUNT(*) AS TimesOrdered
FROM superstore
WHERE Discount > (
    SELECT AVG(Discount)
    FROM superstore
)
AND Sales < (
    SELECT AVG(Sales)
    FROM superstore
)
GROUP BY ProductName
HAVING COUNT(*) >= 3
ORDER BY AvgDiscountPercentage DESC;
# Orders That Were More Profitable Than the Average Order in Their Region
SELECT
    OrderID,
    CustomerName,
    Region,
    Sales,
    Profit,
    OrderDate
FROM superstore s1
```

```
WHERE Profit > (
    SELECT AVG(Profit)
    FROM superstore s2
    WHERE s2.Region = s1.Region
    AND Profit > 0
)
ORDER BY Profit DESC
LIMIT 15;

#Find the Most Recent Order for Each Customer
SELECT
    s1.CustomerID,
    s1.CustomerName,
    s1.OrderID,
    s1.OrderDate,
    s1.Sales,
    s1.ProductName
FROM superstore s1
WHERE s1.OrderDate = (
    SELECT MAX(s2.OrderDate)
    FROM superstore s2
    WHERE s2.CustomerID = s1.CustomerID
)
ORDER BY s1.OrderDate DESC;



#joins#Find Customers Who Bought the Same Product Multiple Times
```

```
SELECT
    d1.CustomerID,
    d1.CustomerName,
    d1.ProductName,
    COUNT(DISTINCT d1.OrderID) AS TimesOrdered,
    MIN(d1.OrderDate) AS FirstOrder,
    MAX(d1.OrderDate) AS LastOrder
FROM superstore d1
JOIN superstore d2 ON d1.CustomerID = d2.CustomerID
    AND d1.ProductName = d2.ProductName
    AND d1.OrderID != d2.OrderID
GROUP BY d1.CustomerID, d1.CustomerName, d1.ProductName
HAVING TimesOrdered >= 2
ORDER BY TimesOrdered DESC;

# Find Products Ordered Together
SELECT
    p1.OrderID,
    p1.ProductName AS Product1,
    p1.ProductCategory AS Category1,
    p2.ProductName AS Product2,
    p2.ProductCategory AS Category2
FROM superstore p1
INNER JOIN superstore p2 ON p1.OrderID = p2.OrderID
WHERE p1.ProductName < p2.ProductName
    AND p1.CustomerSegment = p2.CustomerSegment
LIMIT 15;
```

```
# Customer Purchase Patterns Across Regions
SELECT
    c1.CustomerName,
    c1.Region AS Region1,
    c2.Region AS Region2,
    COUNT(DISTINCT c1.OrderID) AS Orders_Region1,
    COUNT(DISTINCT c2.OrderID) AS Orders_Region2,
    AVG(c1.Sales) AS AvgSale_Region1,
    AVG(c2.Sales) AS AvgSale_Region2
FROM superstore c1
INNER JOIN Document c2 ON c1.CustomerID = c2.CustomerID
    AND c1.Region != c2.Region
GROUP BY c1.CustomerName, c1.Region, c2.Region
HAVING COUNT(DISTINCT c1.OrderID) >= 2
    AND COUNT(DISTINCT c2.OrderID) >= 2
ORDER BY c1.CustomerName;

#Corporate vs Consumer Comparison
SELECT
    corp.ProductName,
    corp.ProductCategory,
    COUNT(corp.OrderID) AS CorporateOrders,
    COUNT(cons.OrderID) AS ConsumerOrders,
    AVG(corp.Sales) AS AvgSale_Corporate,
    AVG(cons.Sales) AS AvgSale_Consumer,
    AVG(corp.Profit) AS AvgProfit_Corporate,
    AVG(cons.Profit) AS AvgProfit_Consumer
```

```
FROM superstore corp
INNER JOIN superstore cons ON corp.ProductName = cons.ProductName
WHERE corp.CustomerSegment = 'Corporate'
   AND cons.CustomerSegment = 'Consumer'
GROUP BY corp.ProductName, corp.ProductCategory
HAVING COUNT(corp.OrderID) >= 2 AND COUNT(cons.OrderID) >= 2
ORDER BY CorporateOrders + ConsumerOrders DESC
LIMIT 10;



#All Possible Customer-Product Combinations
SELECT
   c.CustomerName,
   c.CustomerSegment,
   p.ProductCategory,
   COUNT(o.OrderID) AS TimesOrdered
FROM (
   SELECT DISTINCT CustomerName, CustomerSegment
   FROM superstore
   LIMIT 5
) c
CROSS JOIN (
   SELECT DISTINCT ProductCategory
   FROM superstore
   LIMIT 5
) p
LEFT JOIN superstore o ON c.CustomerName = o.CustomerName
   AND p.ProductCategory = o.ProductCategory
```

```
GROUP BY c.CustomerName, c.CustomerSegment, p.ProductCategory
ORDER BY c.CustomerName, p.ProductCategory;

#Region-ShipMode Performance Matrix
SELECT
   r.Region,
   s.ShipMode,
   COUNT(o.OrderID) AS TotalOrders,
   AVG(o.Sales) AS AvgSaleAmount,
   AVG(o.Profit) AS AvgProfit
FROM (
   SELECT DISTINCT Region FROM superstore
) r
CROSS JOIN (
   SELECT DISTINCT ShipMode FROM superstore
) s
LEFT JOIN superstore o ON r.Region = o.Region AND s.ShipMode = o.ShipMode
GROUP BY r.Region, s.ShipMode
HAVING COUNT(o.OrderID) >= 2
ORDER BY r.Region, TotalOrders DESC;

#Seasonal Analysis by Quarter
SELECT
   CASE
      WHEN MONTH(OrderDate) IN (12, 1, 2) THEN 'Winter'
      WHEN MONTH(OrderDate) IN (3, 4, 5) THEN 'Spring'
      WHEN MONTH(OrderDate) IN (6, 7, 8) THEN 'Summer'
      WHEN MONTH(OrderDate) IN (9, 10, 11) THEN 'Fall'
```

```
    END AS Season,
    COUNT(*) AS Total_Orders,
    SUM(Sales) AS Total_Sales,
    AVG(Sales) AS Avg_Sale_Amount,
    SUM(Profit) AS Total_Profit,
    AVG(Profit) AS Avg_Profit
FROM SuperStore
GROUP BY
    CASE
        WHEN MONTH(OrderDate) IN (12, 1, 2) THEN 'Winter'
        WHEN MONTH(OrderDate) IN (3, 4, 5) THEN 'Spring'
        WHEN MONTH(OrderDate) IN (6, 7, 8) THEN 'Summer'
        WHEN MONTH(OrderDate) IN (9, 10, 11) THEN 'Fall'
    END
ORDER BY Total_Sales DESC;

#Seasonal Analysis with Month Details
SELECT
    CASE
        WHEN MONTH(OrderDate) IN (12, 1, 2) THEN 'Winter'
        WHEN MONTH(OrderDate) IN (3, 4, 5) THEN 'Spring'
        WHEN MONTH(OrderDate) IN (6, 7, 8) THEN 'Summer'
        WHEN MONTH(OrderDate) IN (9, 10, 11) THEN 'Fall'
    END AS Season,
    MONTHNAME(OrderDate) AS Month_Name,
    COUNT(*) AS Total_Orders,
    SUM(Sales) AS Total_Sales,
    SUM(Profit) AS Total_Profit,
```

```
    ROUND(SUM(Profit)/SUM(Sales)*100, 2) AS Profit_Margin_Percent
FROM SuperStore
GROUP BY
    CASE
        WHEN MONTH(OrderDate) IN (12, 1, 2) THEN 'Winter'
        WHEN MONTH(OrderDate) IN (3, 4, 5) THEN 'Spring'
        WHEN MONTH(OrderDate) IN (6, 7, 8) THEN 'Summer'
        WHEN MONTH(OrderDate) IN (9, 10, 11) THEN 'Fall'
    END,
    MONTHNAME(OrderDate),
    MONTH(OrderDate)
ORDER BY Season, MONTH(OrderDate);

#Seasonal Analysis by Product Category
SELECT
    ProductCategory,
    CASE
        WHEN MONTH(OrderDate) IN (12, 1, 2) THEN 'Winter'
        WHEN MONTH(OrderDate) IN (3, 4, 5) THEN 'Spring'
        WHEN MONTH(OrderDate) IN (6, 7, 8) THEN 'Summer'
        WHEN MONTH(OrderDate) IN (9, 10, 11) THEN 'Fall'
    END AS Season,
    COUNT(*) AS Total_Orders,
    SUM(Sales) AS Total_Sales,
    SUM(Profit) AS Total_Profit,
    SUM(Quantityorderednew) AS Total_Units_Sold
FROM SuperStore
GROUP BY ProductCategory,
```

```
    CASE
        WHEN MONTH(OrderDate) IN (12, 1, 2) THEN 'Winter'
        WHEN MONTH(OrderDate) IN (3, 4, 5) THEN 'Spring'
        WHEN MONTH(OrderDate) IN (6, 7, 8) THEN 'Summer'
        WHEN MONTH(OrderDate) IN (9, 10, 11) THEN 'Fall'
    END
ORDER BY ProductCategory, Total_Sales DESC;
```