**Query 1:** **We want to see a list of customers who have placed orders with a total value greater than $50,000.**

**Query:**

USE northwind;

SELECT

c.CustomerID,

c.CompanyName,

SUM(od.UnitPrice \* od.Quantity) AS TotalOrderValue

FROM

customers c

JOIN

orders o ON c.CustomerID = o.CustomerID

JOIN

`order details` od ON o.OrderID = od.OrderID

GROUP BY

c.CustomerID, c.CompanyName

HAVING

TotalOrderValue > 50000

ORDER BY

TotalOrderValue DESC;

**Query 2: Create a view that shows the total sales for each employee in the company.**

**Query:**

USE northwind;

CREATE OR REPLACE VIEW EmployeeSalesView AS

SELECT

e.EmployeeID,

CONCAT(e.FirstName, ' ', e.LastName) AS EmployeeName,

SUM(od.UnitPrice \* od.Quantity) AS TotalSales

FROM

employees e

JOIN

orders o ON e.EmployeeID = o.EmployeeID

JOIN

`order details` od ON o.OrderID = od.OrderID

GROUP BY

e.EmployeeID, EmployeeName;

**Query 3: List top 5 customers by total order value, along with the number of orders and the average order value.**

**Query:**

USE northwind;

SELECT

c.CustomerID,

c.CompanyName,

COUNT(o.OrderID) AS NumberOfOrders,

SUM(od.UnitPrice \* od.Quantity) AS TotalOrderValue,

AVG(od.UnitPrice \* od.Quantity) AS AverageOrderValue

FROM

customers c

JOIN

orders o ON c.CustomerID = o.CustomerID

JOIN

`order details` od ON o.OrderID = od.OrderID

GROUP BY

c.CustomerID, c.CompanyName

ORDER BY

TotalOrderValue DESC

LIMIT 5;

**Query 4: Find the top 5 most profitable products for each salesperson in the company, along with the total revenue and profit generated by each salesperson.**

**Query:**

USE northwind;

WITH ProductProfitability AS (

SELECT

p.ProductID,

p.ProductName,

od.UnitPrice,

od.Quantity,

(od.UnitPrice \* od.Quantity) AS Revenue,

((od.UnitPrice \* 0.8) \* od.Quantity) AS Profit,

o.EmployeeID,

ROW\_NUMBER() OVER (PARTITION BY o.EmployeeID ORDER BY ((od.UnitPrice \* 0.8) \* od.Quantity) DESC) AS ProfitRank

FROM

products p

JOIN

`order details` od ON p.ProductID = od.ProductID

JOIN

orders o ON od.OrderID = o.OrderID

),

SalespersonSummary AS (

SELECT

EmployeeID,

SUM(Revenue) AS TotalRevenue,

SUM(Profit) AS TotalProfit

FROM

ProductProfitability

GROUP BY

EmployeeID

)

SELECT

s.EmployeeID,

CONCAT(e.FirstName, ' ', e.LastName) AS SalespersonName,

p.ProductID,

p.ProductName,

pp.Revenue,

pp.Profit,

s.TotalRevenue,

s.TotalProfit

FROM

ProductProfitability pp

JOIN

SalespersonSummary s ON pp.EmployeeID = s.EmployeeID

JOIN

employees e ON s.EmployeeID = e.EmployeeID

JOIN

products p ON pp.ProductID = p.ProductID

WHERE

pp.ProfitRank <= 5

ORDER BY

s.EmployeeID, pp.ProfitRank;

**Query 5: Find the top 3 categories that generate the most revenue for each quarter of the year, along with the total revenue and profit for each category.**

**Query:**

USE northwind;

WITH CategoryRevenue AS (

SELECT

c.CategoryID,

c.CategoryName,

YEAR(o.OrderDate) AS OrderYear,

QUARTER(o.OrderDate) AS OrderQuarter,

SUM(od.UnitPrice \* od.Quantity) AS Revenue,

SUM(od.UnitPrice \* od.Quantity) - SUM(od.UnitPrice \* od.Quantity) AS Profit,

ROW\_NUMBER() OVER (PARTITION BY YEAR(o.OrderDate), QUARTER(o.OrderDate) ORDER BY SUM(od.UnitPrice \* od.Quantity) DESC) AS RevenueRank

FROM

categories c

JOIN

products p ON c.CategoryID = p.CategoryID

JOIN

`order details` od ON p.ProductID = od.ProductID

JOIN

orders o ON od.OrderID = o.OrderID

GROUP BY

c.CategoryID, c.CategoryName, YEAR(o.OrderDate), QUARTER(o.OrderDate)

)

SELECT

CategoryID,

CategoryName,

OrderYear,

OrderQuarter,

Revenue,

Profit

FROM

CategoryRevenue

WHERE

RevenueRank <= 3

ORDER BY

OrderYear

**Query 6: Find all the customers who have never placed an order, along with their contact information.**

**Query:**

USE northwind;

SELECT

c.CustomerID,

c.CompanyName,

c.ContactName,

c.ContactTitle,

c.Address,

c.City,

c.Region,

c.PostalCode,

c.Country,

c.Phone,

c.Fax

FROM

customers c

LEFT JOIN

orders o ON c.CustomerID = o.CustomerID

WHERE

o.CustomerID IS NULL;

**Query 7: Find the top 3 most ordered products for each category, including categories with no orders, along with the total revenue and profit for each product.**

**Query:**

USE northwind;

WITH ProductOrderCounts AS (

SELECT

p.ProductID,

p.ProductName,

p.CategoryID,

COUNT(o.OrderID) AS OrderCount,

SUM(od.UnitPrice \* od.Quantity) AS Revenue,

SUM((od.UnitPrice - p.UnitPrice) \* od.Quantity) AS Profit,

ROW\_NUMBER() OVER (PARTITION BY p.CategoryID ORDER BY COUNT(o.OrderID) DESC) AS RowNum

FROM

products p

LEFT JOIN

`order details` od ON p.ProductID = od.ProductID

LEFT JOIN

orders o ON od.OrderID = o.OrderID

GROUP BY

p.ProductID, p.ProductName, p.CategoryID

)

SELECT

pc.CategoryID,

pc.CategoryName,

p.ProductID,

p.ProductName,

p.OrderCount,

p.Revenue,

p.Profit

FROM

(

SELECT DISTINCT CategoryID, CategoryName

FROM categories

) pc

LEFT JOIN

ProductOrderCounts p ON pc.CategoryID = p.CategoryID

WHERE

p.RowNum <= 3 OR p.RowNum IS NULL

ORDER BY

pc.CategoryID, p.RowNum;

**Query 8,9: Find the total revenue generated by each employee, including those who haven't made any sales.**

**Query:**

USE northwind;

SELECT

e.EmployeeID,

CONCAT(e.FirstName, ' ', e.LastName) AS EmployeeName,

COALESCE(SUM(od.UnitPrice \* od.Quantity), 0) AS TotalRevenue

FROM

employees e

LEFT JOIN

orders o ON e.EmployeeID = o.EmployeeID

LEFT JOIN

`order details` od ON o.OrderID = od.OrderID

GROUP BY

e.EmployeeID, EmployeeName

ORDER BY

TotalRevenue DESC;

**Query 10: Get a list of all possible combinations of products and suppliers, along with the unit price that each supplier charges for each product.**

**Query:**

USE northwind;

SELECT

p.ProductID,

p.ProductName,

s.SupplierID

FROM

products p

CROSS JOIN

suppliers s;