Use northwind database

**Exercise 1: Find the total number of orders placed by each customer.**

SELECT

c.CustomerID,

c.ContactName,

o.TotalOrders

FROM

Customers c

JOIN (

SELECT

CustomerID,

COUNT(OrderID) as TotalOrders

FROM

Orders

GROUP BY

CustomerID

) o ON c.CustomerID = o.CustomerID;

**Exercise 2: Find the highest total order value for each customer.**

SELECT

c.CustomerID,

c.ContactName,

MAX(o.TotalOrderValue) as HighestOrderValue

FROM

Customers c

JOIN (

SELECT

CustomerID,

SUM(UnitPrice \* Quantity) as TotalOrderValue

FROM

Orders o

JOIN

Order\_Details od ON o.OrderID = od.OrderID

GROUP BY

CustomerID

) o ON c.CustomerID = o.CustomerID

GROUP BY

c.CustomerID;

**Exercise 3: Find the top 3 products that have been ordered the most.**

SELECT

p.ProductName,

p.UnitPrice,

od.TotalQuantityOrdered

FROM

Products p

JOIN (

SELECT

ProductID,

SUM(Quantity) as TotalQuantityOrdered

FROM

OrderDetails

GROUP BY

ProductID

ORDER BY

TotalQuantityOrdered DESC

LIMIT 3

) od ON p.ProductID = od.ProductID;

**Exercise 4: List customers who have placed orders in the last 30 days.**

SELECT

c.CustomerID,

c.ContactName

FROM

Customers c

WHERE

EXISTS (

SELECT

1

FROM

Orders o

WHERE

o.CustomerID = c.CustomerID

AND DATEDIFF(NOW(), o.OrderDate) <= 30

);

**Exercise 5: List customers who have placed orders with a total value over $10,000.**

SELECT

c.CustomerID,

c.ContactName,

o.TotalOrderValue

FROM

Customers c

JOIN (

SELECT

CustomerID,

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

FROM

Orders o

JOIN

OrderDetails od ON o.OrderID = od.OrderID

GROUP BY

CustomerID

HAVING

TotalOrderValue > 10000

) as o ON c.CustomerID = o.CustomerID;

**Exercise 6: List products with their suppliers, and display the total quantity of each product ordered.**

SELECT

p.ProductName,

s.CompanyName AS Supplier,

o.TotalQuantityOrdered

FROM

Products p

JOIN

Suppliers s ON p.SupplierID = s.SupplierID

JOIN (

SELECT

ProductID,

SUM(Quantity) AS TotalQuantityOrdered

FROM

OrderDetails

GROUP BY

ProductID

) o ON p.ProductID = o.ProductID;

**Exercise 7: List the products with the highest and lowest unit prices in each category.**

SELECT

p.ProductName,

p.UnitPrice,

p.CategoryID,

pc.CategoryName,

p1.MaxUnitPrice,

p2.MinUnitPrice

FROM

Products p

JOIN

(SELECT CategoryID, MAX(UnitPrice) AS MaxUnitPrice FROM Products GROUP BY CategoryID) p1

ON p.CategoryID = p1.CategoryID

JOIN

(SELECT CategoryID, MIN(UnitPrice) AS MinUnitPrice FROM Products GROUP BY CategoryID) p2

ON p.CategoryID = p2.CategoryID

JOIN

Categories pc ON p.CategoryID = pc.CategoryID;

**Exercise 8: List customers who have placed orders for more than three different products.**

SELECT

c.CustomerID,

c.ContactName

FROM

Customers c

WHERE

(SELECT COUNT(DISTINCT od.ProductID)

FROM OrderDetails od

WHERE od.OrderID IN (SELECT OrderID FROM Orders o WHERE o.CustomerID = c.CustomerID)) > 3;

**Exercise 9: Find the customers who have placed orders in every category.**

SELECT

c.CustomerID,

c.ContactName

FROM

Customers c

WHERE

(SELECT DISTINCT p.CategoryID

FROM Products p) = (SELECT DISTINCT p.CategoryID

FROM OrderDetails od

JOIN Orders o ON od.OrderID = o.OrderID

JOIN Products p ON od.ProductID = p.ProductID

WHERE o.CustomerID = c.CustomerID);

**Exercise 10: Find the employees who have processed the most orders.**

SELECT

e.EmployeeID,

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

TotalOrdersProcessed

FROM

Employees e

JOIN

(SELECT EmployeeID, COUNT(\*) AS TotalOrdersProcessed

FROM Orders

GROUP BY EmployeeID

ORDER BY TotalOrdersProcessed DESC

LIMIT 1) t ON e.EmployeeID = t.EmployeeID;

**Exercise 11: List the products with the highest and lowest unit prices in each category.**

SELECT

p.ProductName,

p.UnitPrice,

c.CategoryName,

t.MaxUnitPrice,

t.MinUnitPrice

FROM

Products p

JOIN

Categories c ON p.CategoryID = c.CategoryID

JOIN

(SELECT CategoryID, MAX(UnitPrice) AS MaxUnitPrice, MIN(UnitPrice) AS MinUnitPrice

FROM Products

GROUP BY CategoryID) t ON p.CategoryID = t.CategoryID;

**Exercise 12: Find the top 5 customers with the most orders processed.**

SELECT

c.CustomerID,

c.ContactName,

TotalOrdersProcessed

FROM

Customers c

JOIN

(SELECT CustomerID, COUNT(\*) AS TotalOrdersProcessed

FROM Orders

GROUP BY CustomerID

ORDER BY TotalOrdersProcessed DESC

LIMIT 5) t ON c.CustomerID = t.CustomerID;

**Exercise 13: Find the top 5 suppliers with the highest total revenue from product sales.**

SELECT

s.SupplierID,

s.CompanyName,

TotalRevenue

FROM

Suppliers s

JOIN

(SELECT p.SupplierID, SUM(od.UnitPrice \* od.Quantity) AS TotalRevenue

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY p.SupplierID

ORDER BY TotalRevenue DESC

LIMIT 5) t ON s.SupplierID = t.SupplierID;

**Exercise 14: Find the customers who have placed orders on the same day they registered as a customer.**

SELECT

c.CustomerID,

c.ContactName

FROM

Customers c

WHERE

c.CustomerID IN (

SELECT o.CustomerID

FROM Orders o

WHERE DATE(o.OrderDate) = DATE(c.RegistrationDate)

);

**Exercise 15: Find the total revenue for each year, along with the year with the highest revenue.**

SELECT

YEAR(OrderDate) AS OrderYear,

SUM(TotalRevenue) AS AnnualRevenue

FROM

(SELECT o.OrderID, o.OrderDate, SUM(od.UnitPrice \* od.Quantity) AS TotalRevenue

FROM Orders o

JOIN OrderDetails od ON o.OrderID = od.OrderID

GROUP BY o.OrderID, o.OrderDate) AS RevenuePerOrder

GROUP BY

OrderYear

ORDER BY

AnnualRevenue DESC

LIMIT 1;