**1. Tulis query untuk mendapatkan jumlah customer tiap bulan yang melakukan order pada tahun 1997.**

SELECT DATENAME(month, o.OrderDate) AS [Month], COUNT(DISTINCT c.CustomerID) AS [Number of Customers]

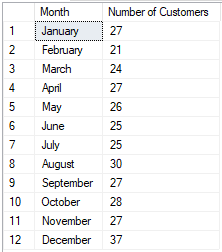
FROM Orders o

INNER JOIN Customers c ON o.CustomerID = c.CustomerID -- additional step to ensure the records match in both tables

WHERE YEAR(o.OrderDate) = 1997

GROUP BY MONTH(o.OrderDate), DATENAME(month, o.OrderDate)

ORDER BY MONTH(o.OrderDate);



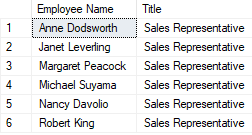
**2. Tulis query untuk mendapatkan nama employee yang termasuk Sales Representative.**

SELECT FirstName + ' ' + LastName AS [Employee Name], Title

FROM Employees

WHERE Title = 'Sales Representative'

ORDER BY FirstName;



**3. Tulis query untuk mendapatkan top 5 nama produk yang quantitynya paling banyak diorder pada bulan Januari 1997.**

SELECT TOP 5 p.ProductName AS [Product Name], SUM(od.Quantity) AS [Total Quantity]

FROM Products p

INNER JOIN [Order Details] od ON p.ProductID = od.ProductID

INNER JOIN Orders o ON od.OrderID = o.OrderID

WHERE MONTH(o.OrderDate) = 1 AND YEAR(o.OrderDate) = 1997

GROUP BY p.ProductName

ORDER BY SUM(od.Quantity) DESC;



**4. Tulis query untuk mendapatkan nama company yang melakukan order Chai pada bulan Juni 1997.**

SELECT c.CompanyName AS [Company Name], p.ProductName AS [Product Name], o.OrderDate AS [Order Date], od.Quantity

FROM Customers c

INNER JOIN Orders o ON c.CustomerID = o.CustomerID

INNER JOIN [Order Details] od ON o.OrderID = od.OrderID

INNER JOIN Products p ON od.ProductID = p.ProductID

WHERE p.ProductName = 'Chai' AND MONTH(o.OrderDate) = 6 AND YEAR(o.OrderDate) = 1997;



**5. Tulis query untuk mendapatkan jumlah OrderID yang pernah melakukan pembelian (unit\_price dikali quantity) <=100, 100<x<=250, 250<x<=500, dan >500.**

WITH PurchaseAmountTable AS (

SELECT OrderID, SUM(UnitPrice \* Quantity) AS PurchaseAmount

FROM [Order Details]

GROUP BY OrderID

)

SELECT COUNT(CASE WHEN PurchaseAmount <= 100 THEN 1 END) AS '<=100',

COUNT(CASE WHEN PurchaseAmount > 100 AND PurchaseAmount <= 250 THEN 2 END) AS '100 < x <= 250',

COUNT(CASE WHEN PurchaseAmount > 250 AND PurchaseAmount <= 500 THEN 3 END) AS '250 < x <= 500',

COUNT(CASE WHEN PurchaseAmount > 500 THEN 4 END) AS '>500'

FROM PurchaseAmountTable;



**6. Tulis query untuk mendapatkan Company name pada tabel customer yang melakukan pembelian di atas 500 pada tahun 1997.**

WITH PurchaseAmountTable AS (

SELECT c.CompanyName, SUM(od.UnitPrice\*od.Quantity) AS TotalPurchaseAmount

FROM Customers c

INNER JOIN Orders o ON c.CustomerID = o.CustomerID

INNER JOIN [Order Details] od ON o.OrderID = od.OrderID

WHERE YEAR(o.OrderDate) = 1997

GROUP BY c.CompanyName

)

SELECT DISTINCT CompanyName, TotalPurchaseAmount FROM PurchaseAmountTable

WHERE TotalPurchaseAmount > 500

ORDER BY CompanyName;



**7. Tulis query untuk mendapatkan nama produk yang merupakan Top 5 sales tertinggi tiap bulan di tahun 1997.**

WITH TotalSalesTable AS (

SELECT

p.ProductName,

MONTH(o.OrderDate) AS Month,

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

ROW\_NUMBER() OVER (PARTITION BY MONTH(o.OrderDate) ORDER BY SUM(od.UnitPrice \* od.Quantity) DESC) AS Row

FROM Products p

INNER JOIN [Order Details] od ON p.ProductID = od.ProductID

INNER JOIN Orders o ON o.OrderID = od.OrderID

WHERE YEAR(o.OrderDate) = 1997

GROUP BY p.ProductName, MONTH(o.OrderDate)

)

SELECT ProductName, Month, TotalSales

FROM TotalSalesTable

WHERE row <= 5

ORDER BY Month, TotalSales DESC



**8. Buatlah view untuk melihat Order Details yang berisi OrderID, ProductID, ProductName, UnitPrice, Quantity, Discount, Harga setelah diskon.**

DROP VIEW IF EXISTS OrderDetailsView;

GO

CREATE VIEW OrderDetailsView AS

SELECT

od.OrderID,

od.ProductID,

p.ProductName,

od.UnitPrice,

od.Quantity,

od.Discount,

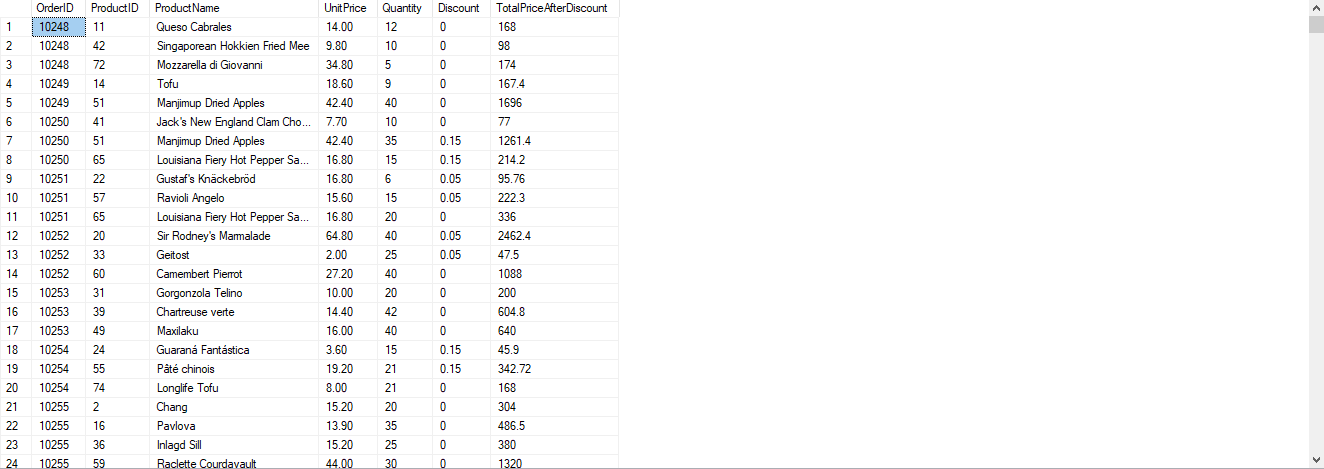
(od.UnitPrice \* od.Quantity \* (1 - od.Discount)) AS TotalPriceAfterDiscount

FROM [Order Details] od

INNER JOIN Products p ON od.ProductID = p.ProductID;

GO

SELECT \* FROM OrderDetailsView



**9. Buatlah procedure Invoice untuk memanggil CustomerID, CustomerName/company name, OrderID, OrderDate, RequiredDate, ShippedDate jika terdapat inputan CustomerID tertentu.**

DROP PROCEDURE IF EXISTS GenerateInvoice

GO

CREATE PROCEDURE GenerateInvoice

@CustomerID NVARCHAR(5)

AS

BEGIN

SET NOCOUNT ON;

SELECT

c.CustomerID,

c.CompanyName AS 'CustomerName',

o.OrderID,

o.OrderDate,

o.RequiredDate,

o.ShippedDate

FROM Customers c

INNER JOIN Orders o ON c.CustomerID = o.CustomerID

INNER JOIN [Order Details] od ON o.OrderID = od.OrderID

WHERE c.CustomerID = @CustomerID

END

GO

EXEC GenerateInvoice @CustomerID = ALFKI;

