**Nama: Hendri Darmawan**

**Kelompok: 2**

**Mini Project DE**

* **Intermediate Queries**

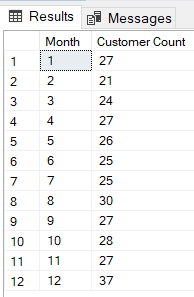
1. Tulis query untuk mendapatkan jumlah customer tiap bulan yang melakukan order pada tahun 1997.

SELECT MONTH(OrderDate) as 'Month', COUNT(DISTINCT CustomerID) as 'Customer Count'

FROM Orders

WHERE YEAR(OrderDate) = 1997

GROUP BY MONTH(OrderDate)

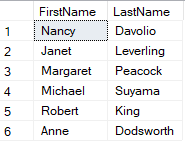
****

1. Tulis query untuk mendapatkan nama employee yang termasuk Sales Representative.

SELECT FirstName, LastName

FROM Employees

WHERE Title = 'Sales Representative'



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

SELECT TOP 5 Products.ProductName, SUM([Order Details].Quantity) as 'Total Quantity'

FROM Products

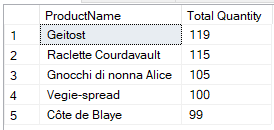
INNER JOIN [Order Details] ON Products.ProductID = [Order Details].ProductID

INNER JOIN Orders ON [Order Details].OrderID = Orders.OrderID

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

GROUP BY Products.ProductName

ORDER BY SUM([Order Details].Quantity) DESC



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

SELECT DISTINCT CompanyName

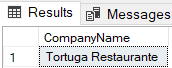
FROM Customers

INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

INNER JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID

INNER JOIN Products ON [Order Details].ProductID = Products.ProductID

WHERE MONTH(Orders.OrderDate) = 6 AND YEAR(Orders.OrderDate) = 1997 AND ProductName = 'Chai'



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

SELECT

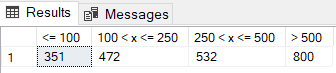
SUM(CASE WHEN (UnitPrice \* Quantity) <= 100 THEN 1 ELSE 0 END) as '<= 100',

SUM(CASE WHEN (UnitPrice \* Quantity) > 100 AND (UnitPrice \* Quantity) <= 250 THEN 1 ELSE 0 END) as '100 < x <= 250',

SUM(CASE WHEN (UnitPrice \* Quantity) > 250 AND (UnitPrice \* Quantity) <= 500 THEN 1 ELSE 0 END) as '250 < x <= 500',

SUM(CASE WHEN (UnitPrice \* Quantity) > 500 THEN 1 ELSE 0 END) as '> 500'

FROM [Order Details]



1. Tulis query untuk mendapatkan Company name yang melakukan sales di atas 500 pada tahun 1997.

SELECT DISTINCT CompanyName

FROM Customers

INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

INNER JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID

WHERE (UnitPrice \* Quantity) > 500 AND YEAR(Orders.OrderDate) = 1997



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

WITH sales\_per\_month AS (

SELECT

MONTH(OrderDate) AS month,

ProductName,

SUM(Quantity \* Products.UnitPrice) AS total\_sales,

ROW\_NUMBER() OVER (PARTITION BY MONTH(OrderDate) ORDER BY SUM(Quantity \* Products.UnitPrice) DESC) as rank

FROM Orders

JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID

JOIN Products ON [Order Details].ProductID = Products.ProductID

WHERE YEAR(OrderDate) = 1997

GROUP BY MONTH(OrderDate), ProductName

)

SELECT

month,

ProductName,

total\_sales

FROM sales\_per\_month

WHERE rank <= 5

ORDER BY month, total\_sales DESC



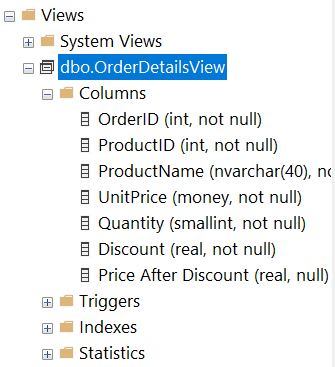
1. Buatlah view untuk melihat Order Details yang berisi OrderID, ProductID, ProductName, UnitPrice, Quantity, Discount, Harga setelah diskon.

CREATE VIEW OrderDetailsView AS

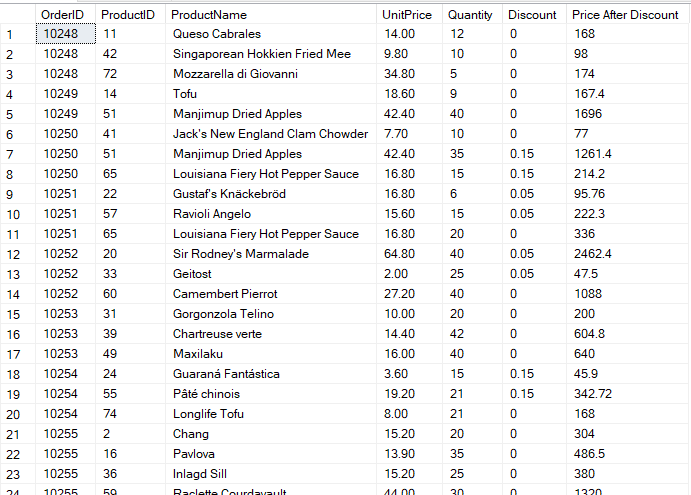
SELECT OrderID, [Order Details].ProductID, ProductName, [Order Details].UnitPrice, Quantity, Discount, ([Order Details].UnitPrice \* Quantity \* (1 - Discount)) as 'Price After Discount'

FROM [Order Details]

INNER JOIN Products ON [Order Details].ProductID = Products.ProductID



SELECT \* FROM OrderDetailsView;



1. Buatlah procedure Invoice untuk memanggil CustomerID, CustomerName, OrderID, OrderDate, RequiredDate, ShippedDate jika terdapat inputan CustomerID tertentu.

CREATE PROCEDURE Invoice (@customerID nchar(5))

AS

BEGIN

SELECT Customers.CustomerID, CompanyName as 'CustomerName', Orders.OrderID, OrderDate, RequiredDate, ShippedDate

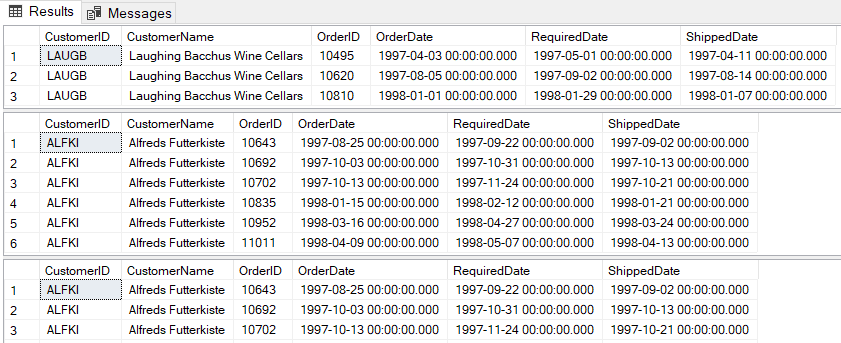
FROM Customers

INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

WHERE Customers.CustomerID = @customerID

END

EXEC Invoice 'LAUGB'



* **Case Study**

**Employee Analysis**

**Contoh: kita dapat menganalisis siapa dan title employee yang banyak berurusan dengan order.**

SELECT Employees.FirstName, Employees.LastName, Employees.Title, Orders.OrderID

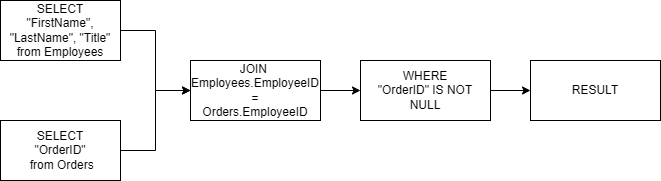
FROM Employees

JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID

WHERE Orders.OrderID IS NOT NULL



**Flowchart**



**Create new table**

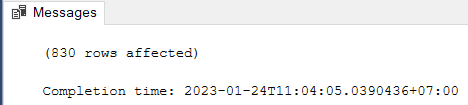
SELECT Employees.FirstName, Employees.LastName, Employees.Title, Orders.OrderID

INTO EmployeeOrders

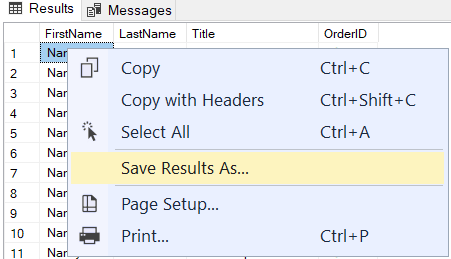
FROM Employees

JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID

WHERE Orders.OrderID IS NOT NULL



**Export ke csv**



**Shipper Analysis**

**Contoh : orderan paling banyak dikirim ke kota mana, dengan shipper apa.**

SELECT Shippers.CompanyName as 'ShipperName', Orders.ShipCity as 'City'

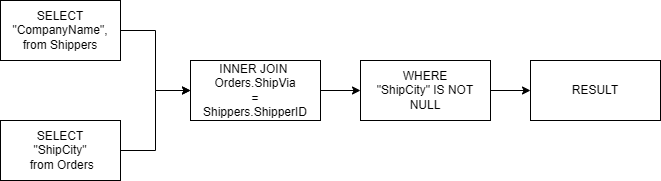
FROM Orders

INNER JOIN Shippers ON Orders.ShipVia = Shippers.ShipperID

WHERE Orders.ShipCity IS NOT NULL



**Flowchart:**



**Create new table:**

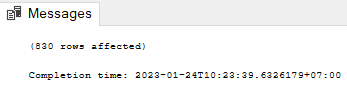
SELECT Shippers.CompanyName as 'ShipperName', Orders.ShipCity as 'City'

INTO ShipperAndCity

FROM Orders

INNER JOIN Shippers ON Orders.ShipVia = Shippers.ShipperID

WHERE Orders.ShipCity IS NOT NULL



**Export ke csv:**

