Intermediate queries

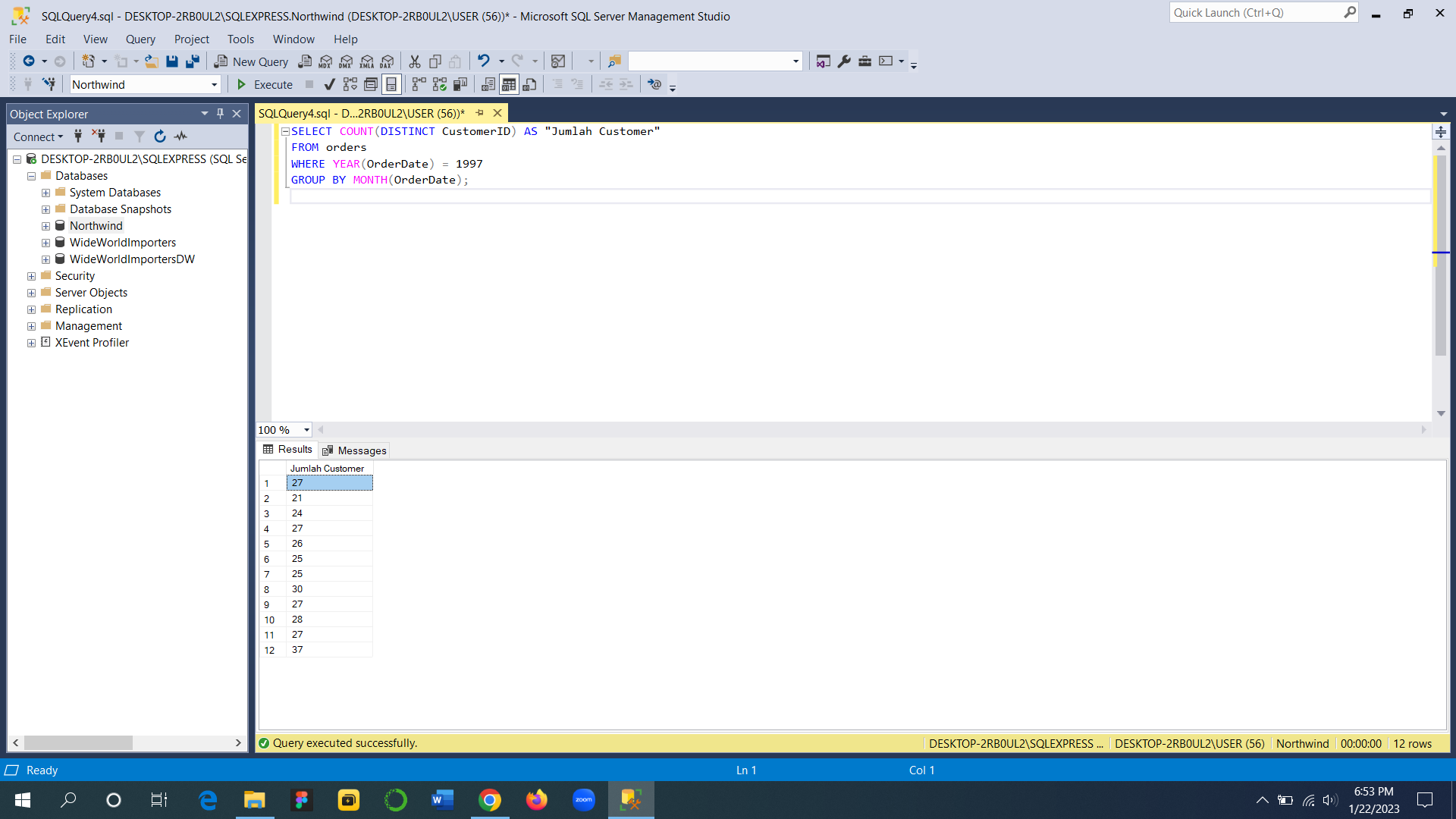
1. Tulis query untuk mendapatkan jumlah customer tiap bulan yang melakukan order pada tahun 1997.

SELECT COUNT(DISTINCT CustomerID) AS "Jumlah Customer"

FROM orders

WHERE YEAR(OrderDate) = 1997

GROUP BY MONTH(OrderDate);

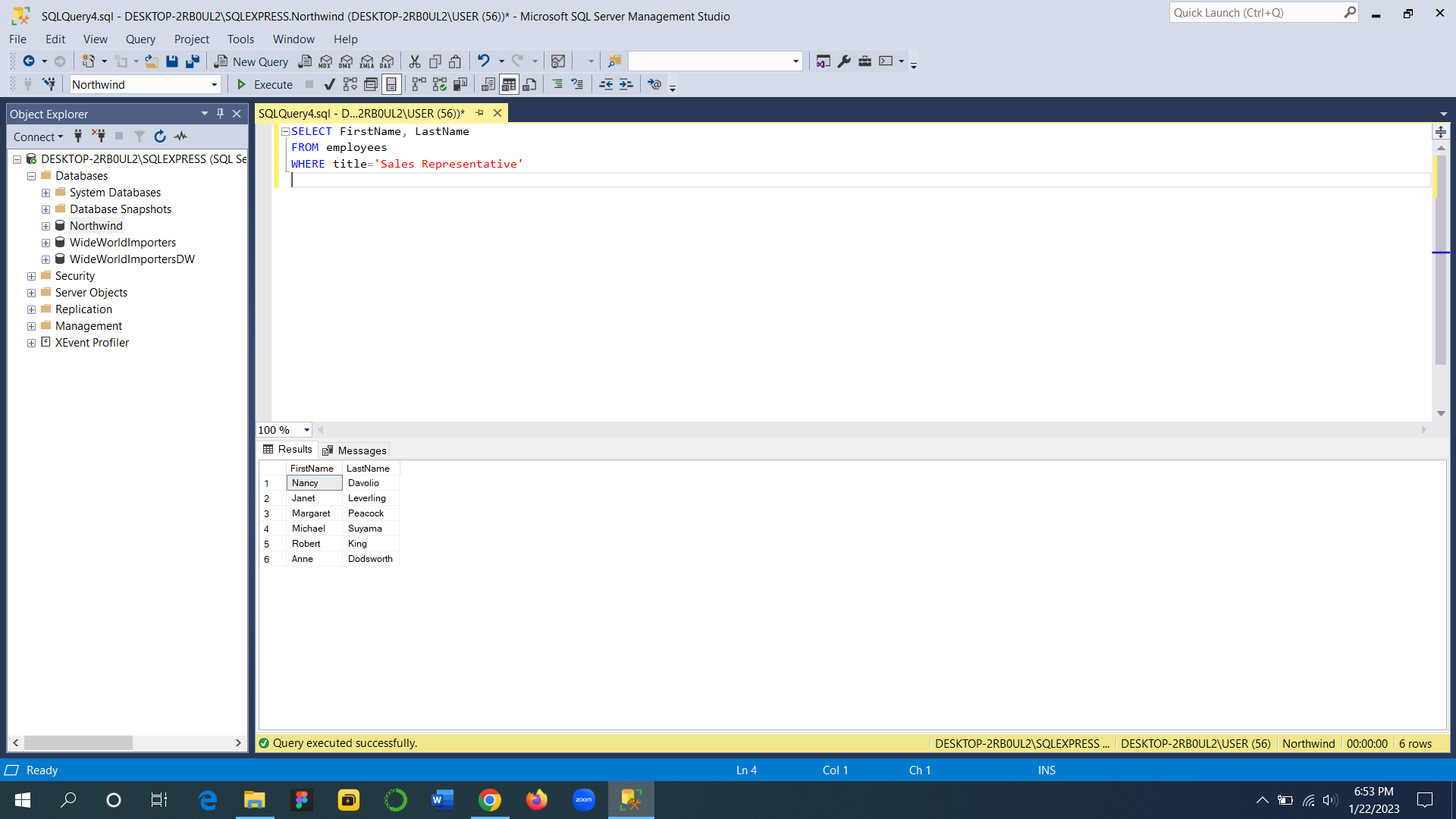


1. Tulis query untuk mendapatkan nama employee yang termasuk Sales Representative.

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

P.ProductName,

SUM(Quantity) as total\_quantity

FROM

[Products] P

left join [Order Details] O on P.[ProductID] = O.[ProductID]

left join [Orders] OD on O.[OrderID] =OD.[OrderID]

WHERE

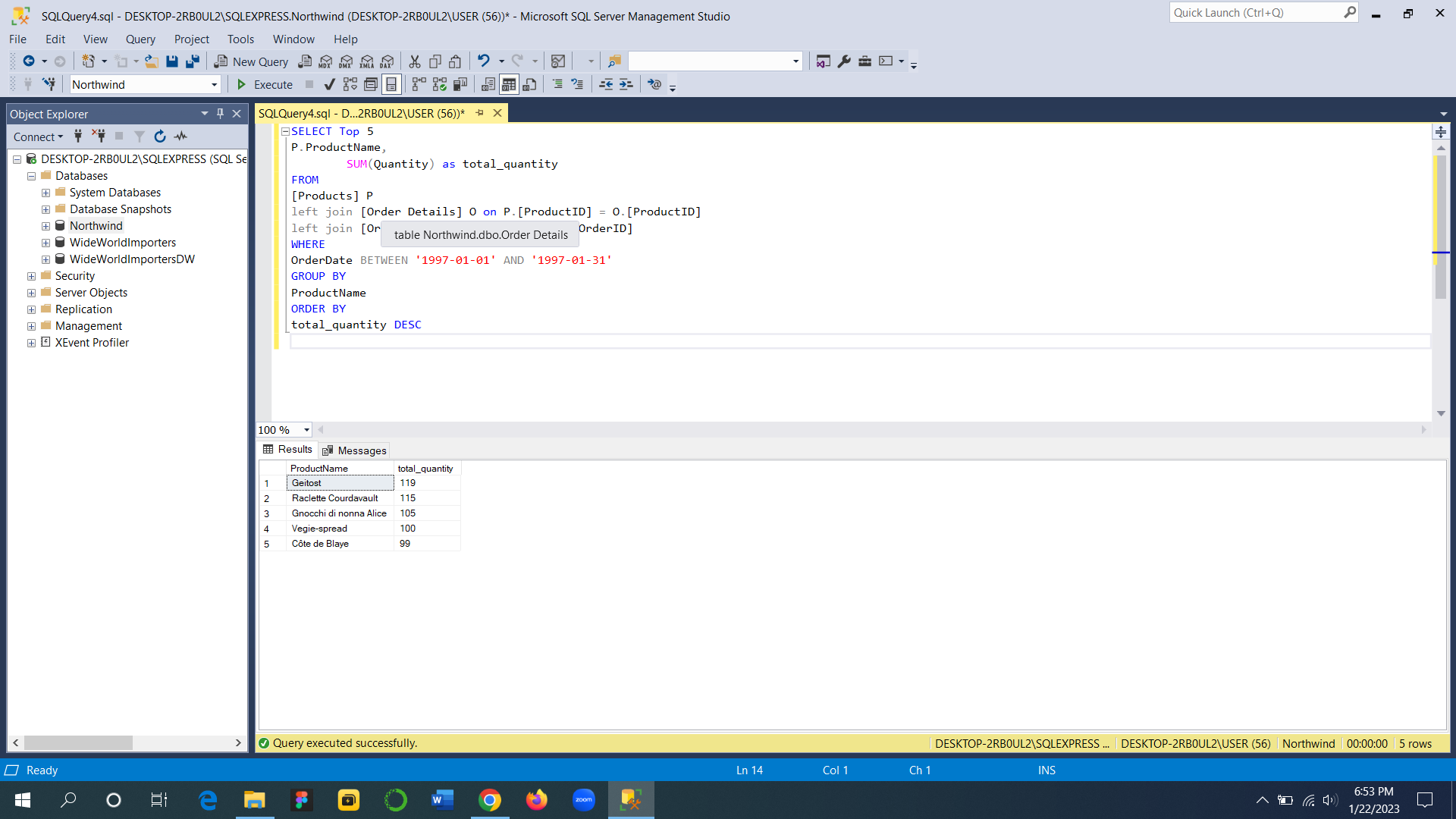
OrderDate BETWEEN '1997-01-01' AND '1997-01-31'

GROUP BY

ProductName

ORDER BY

total\_quantity DESC



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

SELECT

c.CompanyName

FROM

[Customers] c

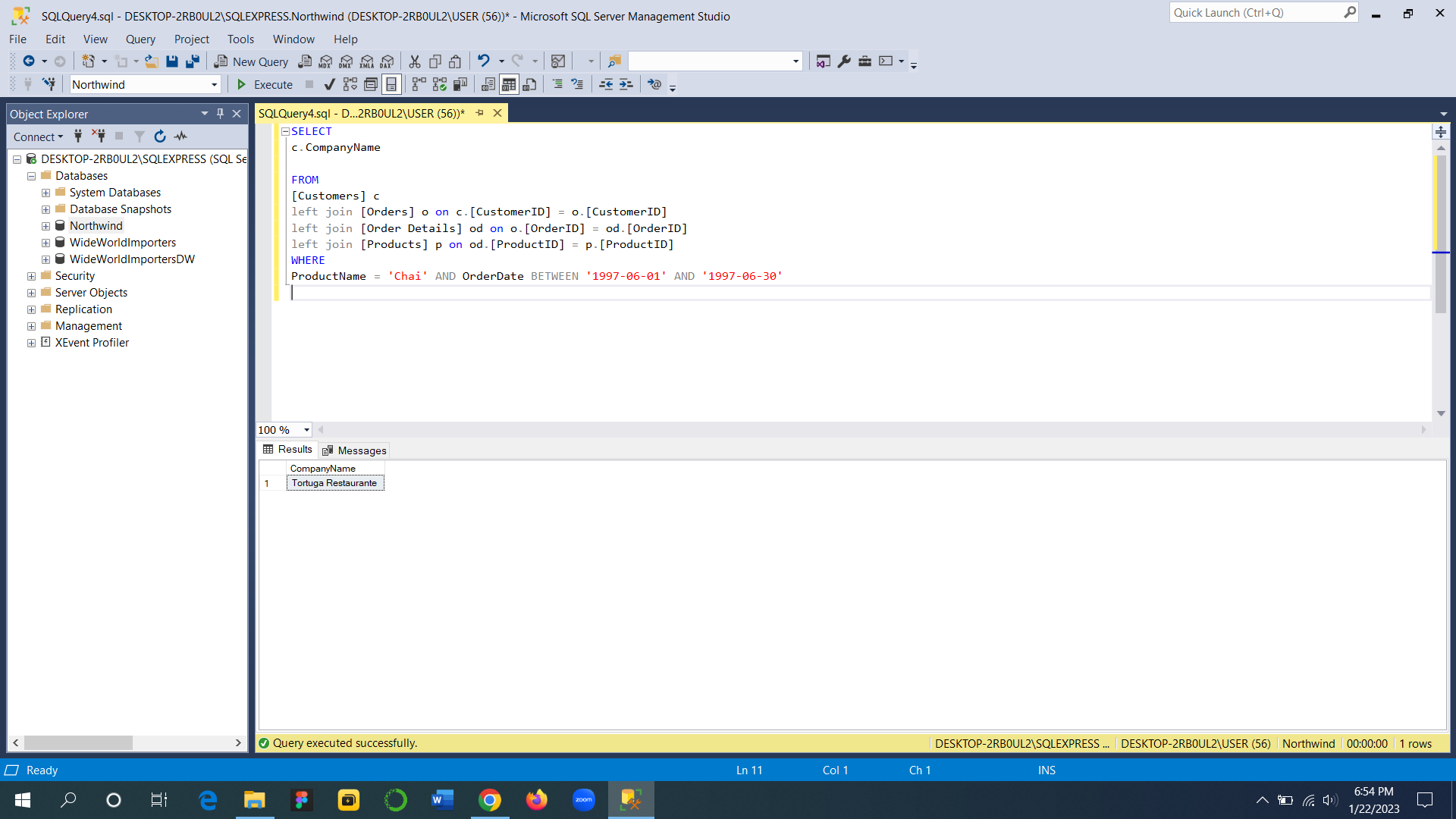
left join [Orders] o on c.[CustomerID] = o.[CustomerID]

left join [Order Details] od on o.[OrderID] = od.[OrderID]

left join [Products] p on od.[ProductID] = p.[ProductID]

WHERE

ProductName = 'Chai' AND OrderDate BETWEEN '1997-06-01' AND '1997-06-30'



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

SELECT

COUNT(CASE WHEN P.UnitPrice \* quantity <= 100 THEN OD.OrderID END) as "Dibawah sama dengan 100",

COUNT(CASE WHEN P.UnitPrice \* quantity > 100 and P.UnitPrice \* quantity <= 250 THEN OD.OrderID END) as " Lebih besar dari 100 lebih kecil sama dengan 250",

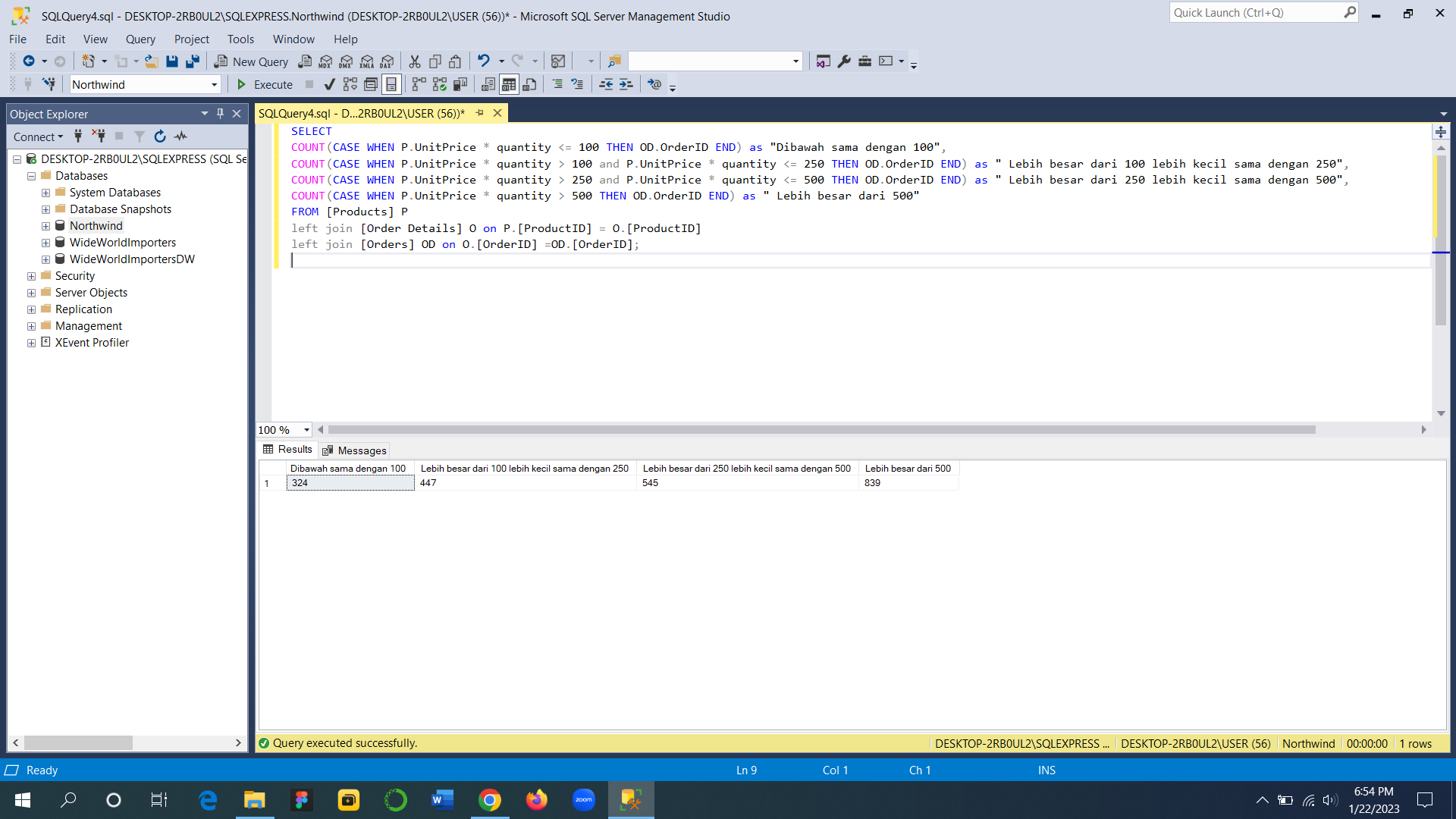
COUNT(CASE WHEN P.UnitPrice \* quantity > 250 and P.UnitPrice \* quantity <= 500 THEN OD.OrderID END) as " Lebih besar dari 250 lebih kecil sama dengan 500",

COUNT(CASE WHEN P.UnitPrice \* quantity > 500 THEN OD.OrderID END) as " Lebih besar dari 500"

FROM [Products] P

left join [Order Details] O on P.[ProductID] = O.[ProductID]

left join [Orders] OD on O.[OrderID] =OD.[OrderID];



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

SELECT DISTINCT C.CompanyName AS 'Nama Perusahaan'

FROM [Customers] C

left join [Orders] O on C.CustomerID = O.CustomerID

left join [Order Details] OD on O.OrderID = OD.OrderID

WHERE

(SELECT SUM(UnitPrice\*Quantity)

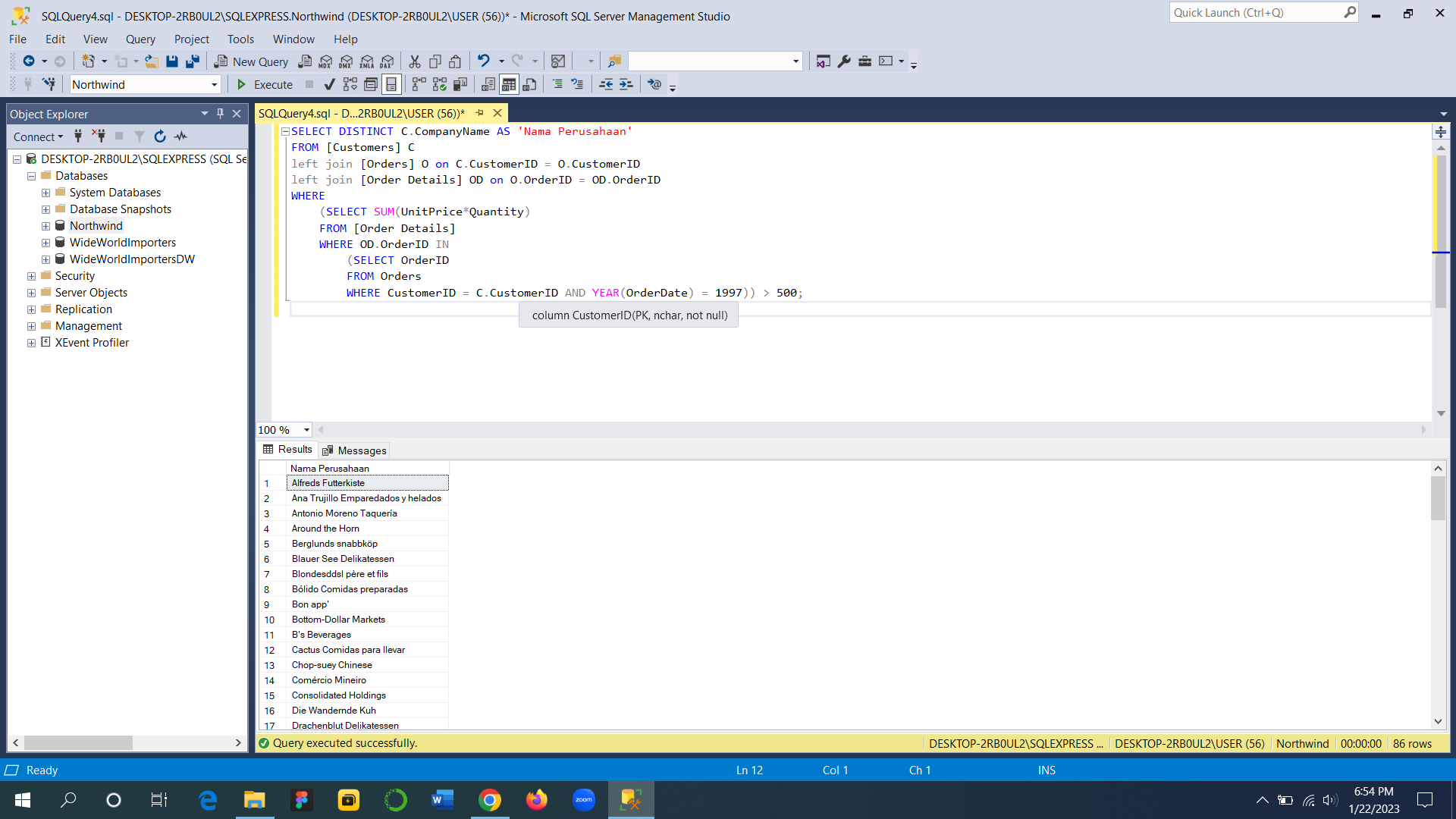
FROM [Order Details]

WHERE OD.OrderID IN

(SELECT OrderID

FROM Orders

WHERE CustomerID = C.CustomerID AND YEAR(OrderDate) = 1997)) > 500;



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

SELECT top 5

ProductName,

SUM(P.UnitPrice\*Quantity) as TotalSales

FROM [Order Details] OD

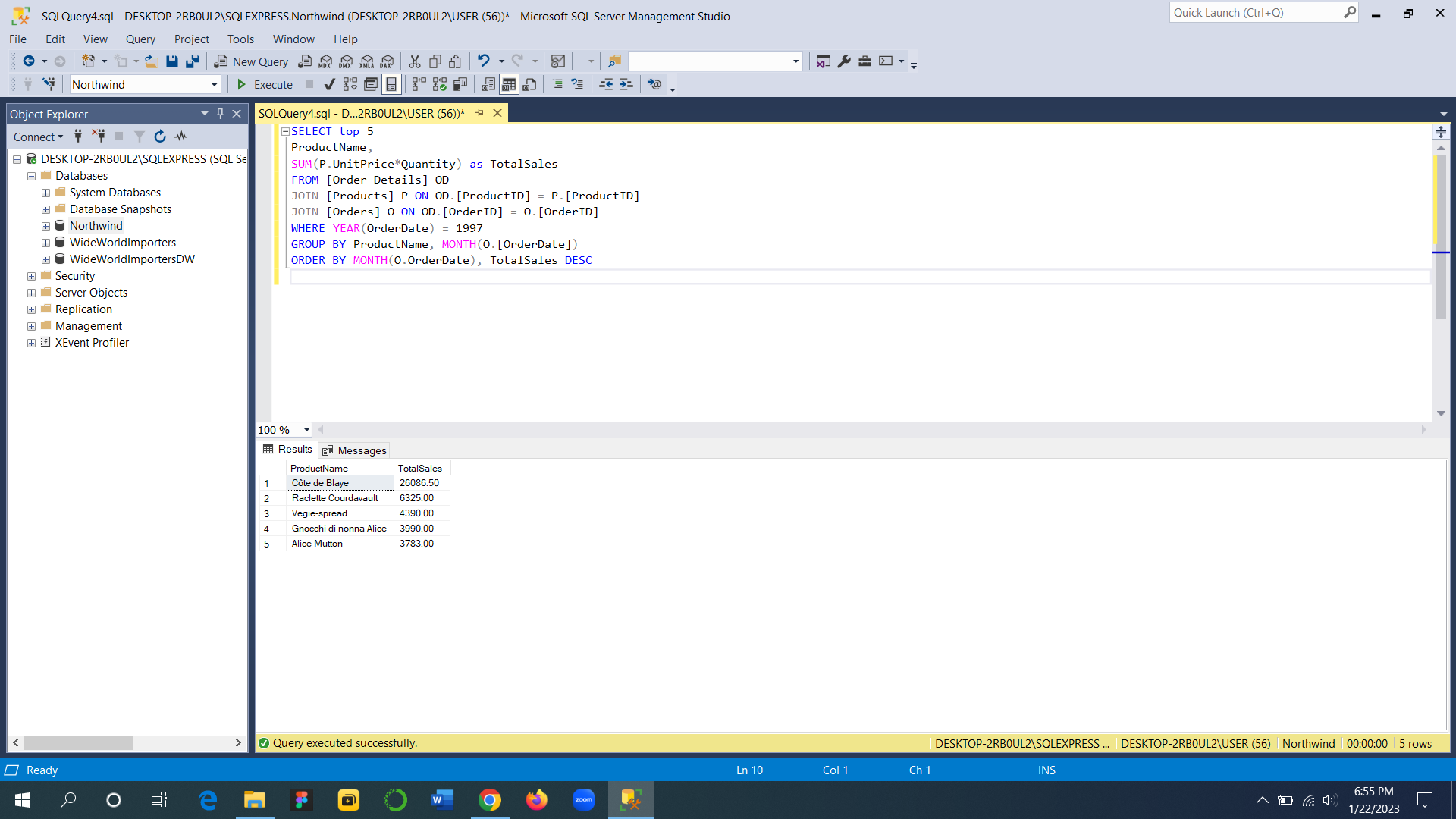
JOIN [Products] P ON OD.[ProductID] = P.[ProductID]

JOIN [Orders] O ON OD.[OrderID] = O.[OrderID]

WHERE YEAR(OrderDate) = 1997

GROUP BY ProductName, MONTH(O.[OrderDate])

ORDER BY MONTH(O.OrderDate), TotalSales DESC



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

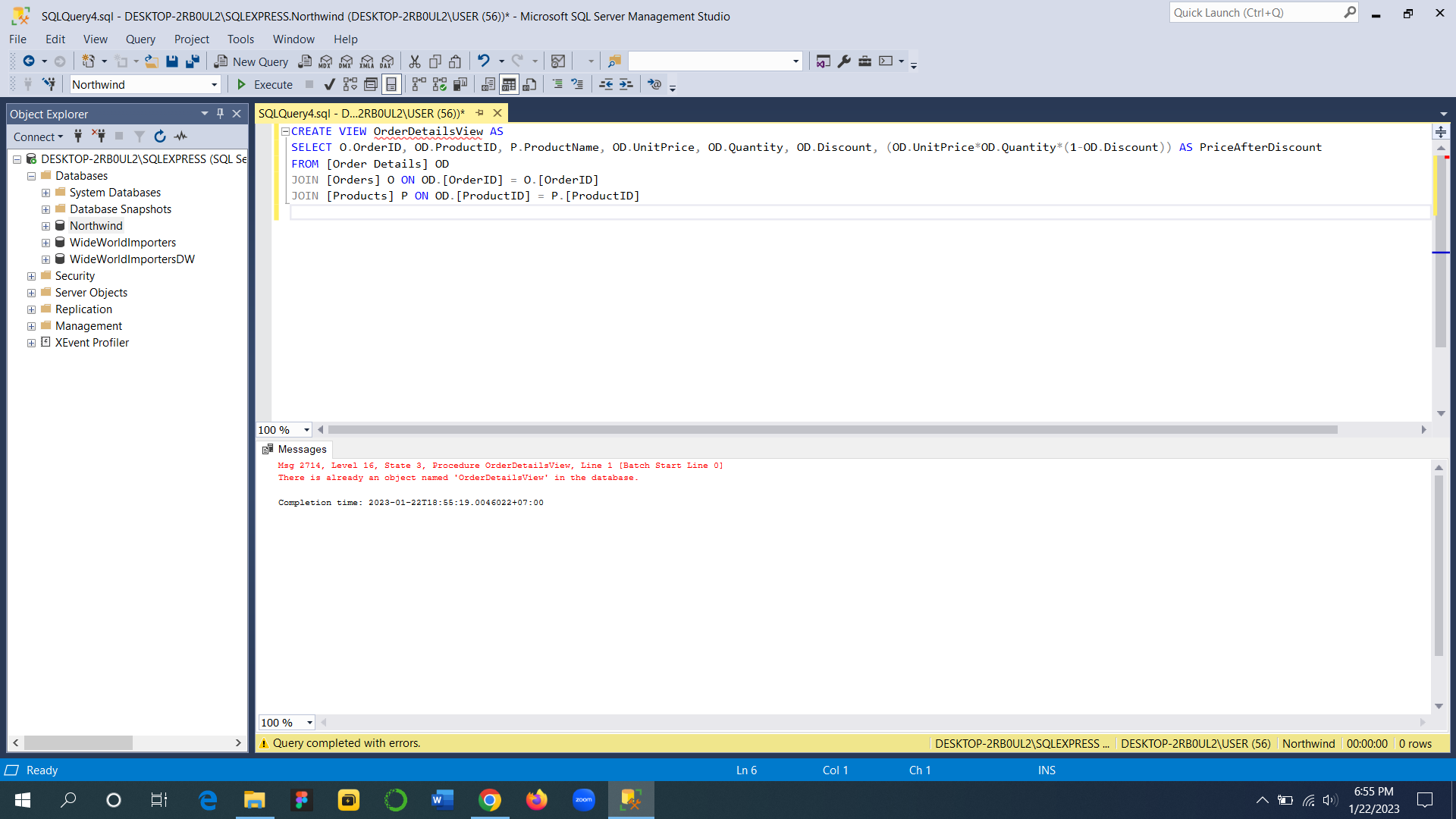
CREATE VIEW OrderDetailsView AS

SELECT O.OrderID, OD.ProductID, P.ProductName, OD.UnitPrice, OD.Quantity, OD.Discount, (OD.UnitPrice\*OD.Quantity\*(1-OD.Discount)) AS PriceAfterDiscount

FROM [Order Details] OD

JOIN [Orders] O ON OD.[OrderID] = O.[OrderID]

JOIN [Products] P ON OD.[ProductID] = P.[ProductID]

\* Karena query sudah execute sebelumnya jadi view sudah terbuat

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

CREATE PROCEDURE Invoice (@CustomerID VARCHAR(255))

AS

BEGIN

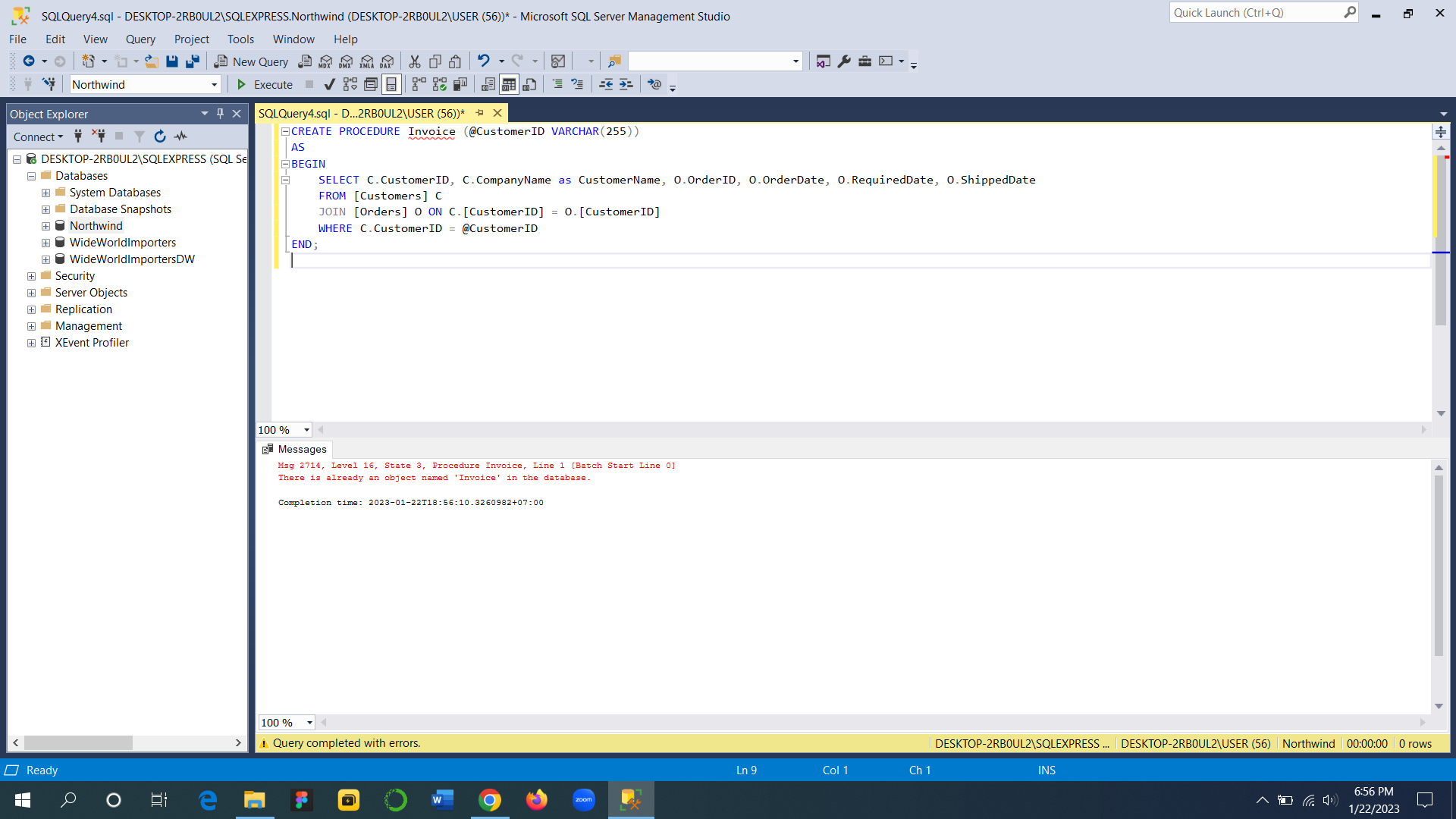
SELECT C.CustomerID, C.CompanyName as CustomerName, O.OrderID, O.OrderDate, O.RequiredDate, O.ShippedDate

FROM [Customers] C

JOIN [Orders] O ON C.[CustomerID] = O.[CustomerID]

WHERE C.CustomerID = @CustomerID

END;



\* Karena procedure sudah di execute sebelumnya jadi procedure sudah terbuat