**Assignment 4**

1. **Create a stored procedure in the Northwind database that will calculate the average value of Freight for a specified customer.Then, a business rule will be added that will be triggered before every Update and Insert command in the Orders controller,and will use the stored procedure to verify that the Freight does not exceed the average freight. If it does, a message will be displayed and the command will be cancelled.**

alter PROCEDURE que1

AS

SELECT CustomerID, AVG(Freight) as AvgFreight

FROM Orders

GROUP BY CustomerID

GO

--11078

insert into orders values ( 'ALFKI', 1,'1997-08-25 00:00:00.000', '1997-08-01 00:00:00.000','1997-01-01 00:00:00.000', 1,100, 'Wolski Zajazd', 'ul. Filtrowa 68',

'Warszawa', 'Tachira', 24100, 'brazil')

exec que1

UPDATE Orders SET Freight=30 WHERE OrderID = 10248

SELECT \* FROM Orders WHERE OrderID = 10248

create TRIGGER tr\_que1\_update

ON orders

INSTEAD OF UPDATE

AS

BEGIN

Declare @OrderID int

Declare @CustomerID varchar(50)

Declare @Freight money

Declare @AvgFreight money

Declare @t\_ave TABLE(CustomerID nchar(5), AvgFreight money)

INSERT @t\_ave

exec que1

Select \* Into #Temptable FROM Inserted

While(Exists(Select OrderID from #TempTable))

Begin

Select TOP 1 @OrderID = OrderID, @CustomerID = CustomerID, @Freight=Freight

FROM #Temptable

SET @AvgFreight = (SELECT AvgFreight FROM @t\_ave WHERE CustomerID = @CustomerID)

Print @Freight

Print @AvgFreight

IF @Freight > @AvgFreight

BEGIN

RAISERROR ('ABOVE AVERAGE',16,1)

END

ELSE

BEGIN

UPDATE Orders SET Freight = @Freight WHERE OrderID=@OrderID

END

Delete from #TempTable where OrderID = @OrderID

End

END

create TRIGGER tr\_que1\_insert

ON orders

INSTEAD OF INSERT

AS

BEGIN

Declare @OrderID int

Declare @CustomerID varchar(50)

Declare @Freight money

Declare @AvgFreight money

Declare @t\_ave TABLE(CustomerID nchar(5), AvgFreight money)

INSERT @t\_ave

exec que1

Select \* Into #Temptable FROM Inserted

While(Exists(Select OrderID from #TempTable))

Begin

Select TOP 1 @OrderID = OrderID, @CustomerID = CustomerID, @Freight=Freight

FROM #Temptable

SET @AvgFreight = (SELECT AvgFreight FROM @t\_ave WHERE CustomerID = @CustomerID)

IF @Freight > @AvgFreight

BEGIN

RAISERROR ('ABOVE AVERAGE',16,1)

END

ELSE

BEGIN

INSERT INTO Orders (CustomerID,EmployeeID,OrderDate,RequiredDate,ShippedDate,ShipVia,Freight,ShipName,ShipAddress,ShipCity,ShipRegion,ShipPostalCode,ShipCountry)

SELECT CustomerID,EmployeeID,OrderDate,RequiredDate,ShippedDate,ShipVia,Freight,ShipName,ShipAddress,ShipCity,ShipRegion,ShipPostalCode,ShipCountry

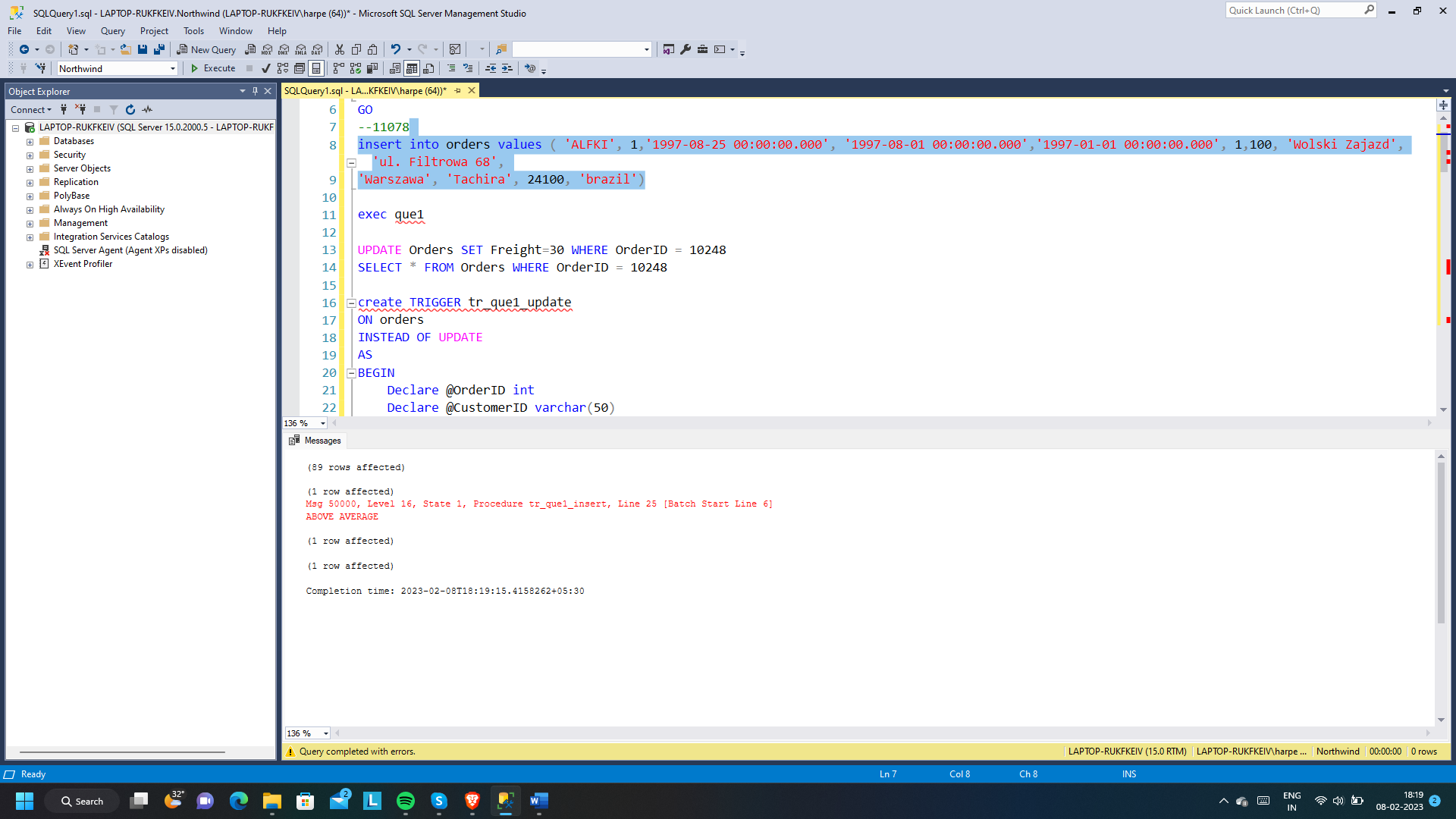
From Inserted

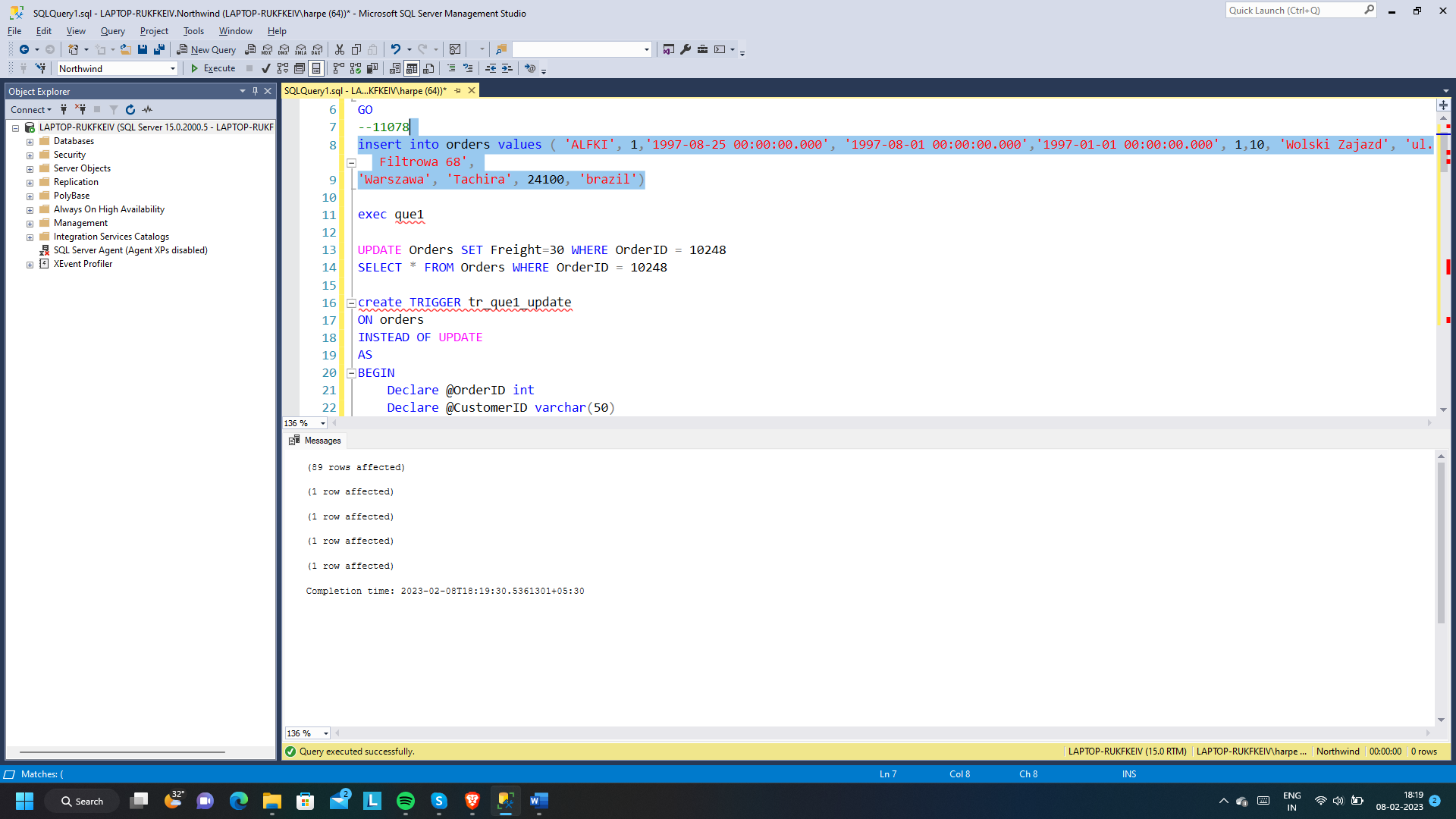
END

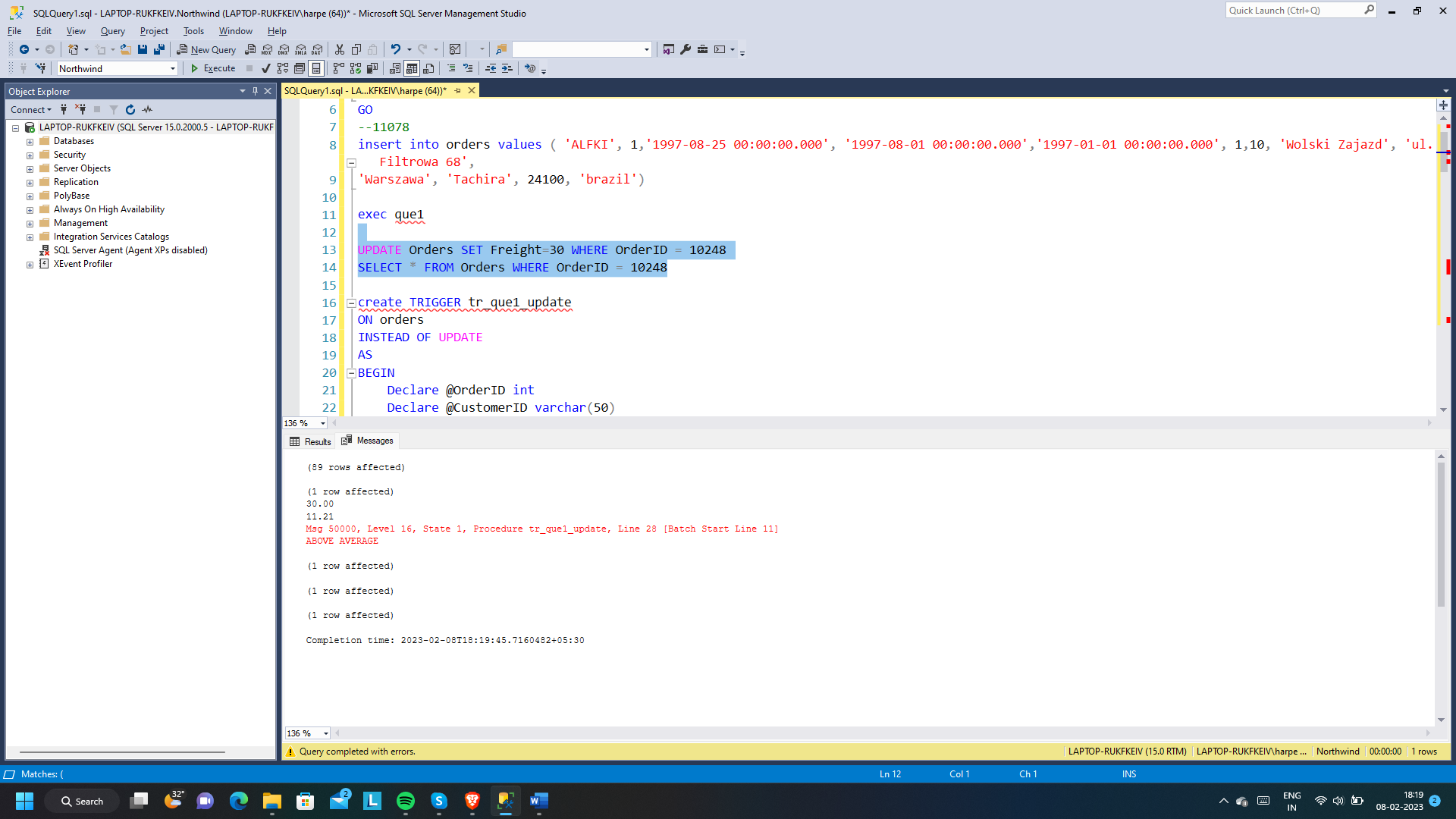
Delete from #TempTable where OrderID = @OrderID

End

END







1. **write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country.**

create procedure prsalesbycountry

@country varchar(30)

as begin

select e.FirstName, e.LAStName, o.ShipCountry, od.Subtotal as "Total sales"

FROM Employees e

left join ([Orders] o join [Order Subtotals] od on o.OrderID= od.OrderID)

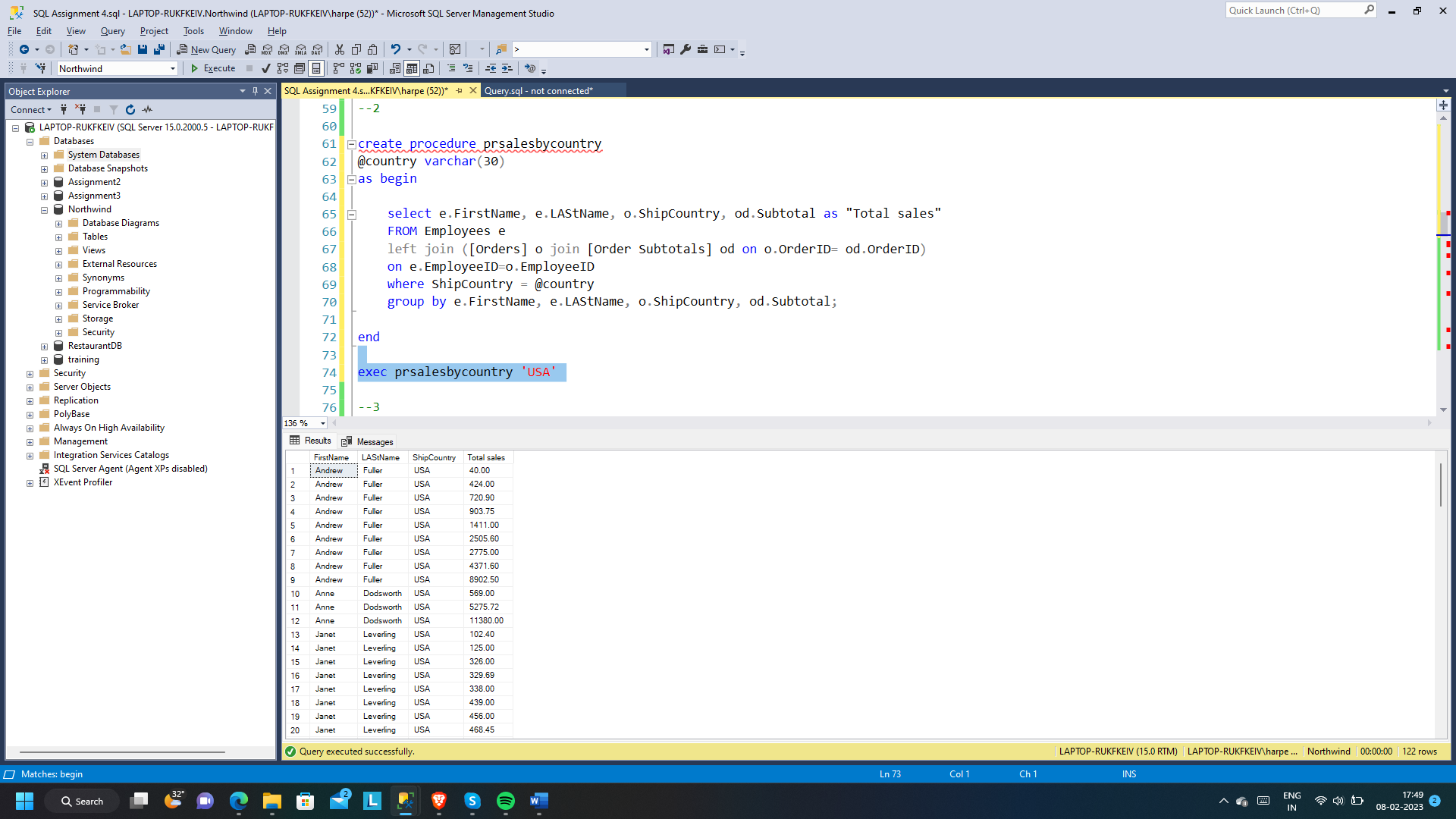
on e.EmployeeID=o.EmployeeID

where ShipCountry = @country

group by e.FirstName, e.LAStName, o.ShipCountry, od.Subtotal;

end

exec prsalesbycountry 'USA'



1. **write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales by Year**

select year(OrderDate) as [Year], count(OrderID) as [Total Orders] from Orders

group by year(OrderDate)

alter procedure prSalesByYear

@Year int

as

begin

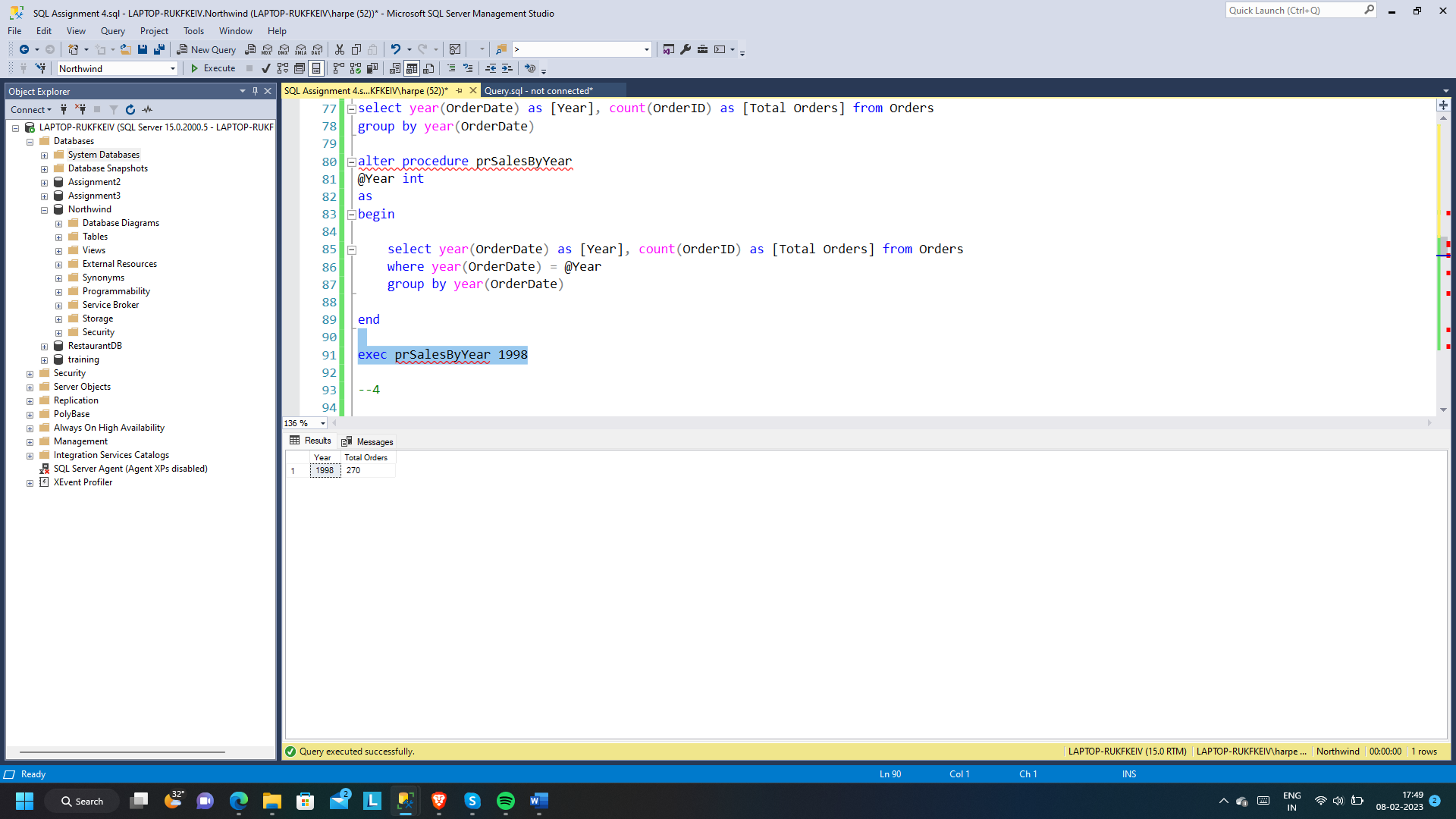
select year(OrderDate) as [Year], count(OrderID) as [Total Orders] from Orders

where year(OrderDate) = @Year

group by year(OrderDate)

end

exec prSalesByYear 1998



1. **write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales By Category**

select c.CategoryID, c.CategoryName as [Category], count(o.OrderID) as [Total Orders] from Categories c

join Products p on c.CategoryID = p.CategoryID

join [Order Details] o on p.ProductID = o.ProductID

group by c.CategoryID, c.CategoryName

alter procedure prSalesByCategory

@CategoryID int

as

begin

select c.CategoryID, c.CategoryName as [Category], count(o.OrderID) as [Total Orders] from Categories c

join Products p on c.CategoryID = p.CategoryID

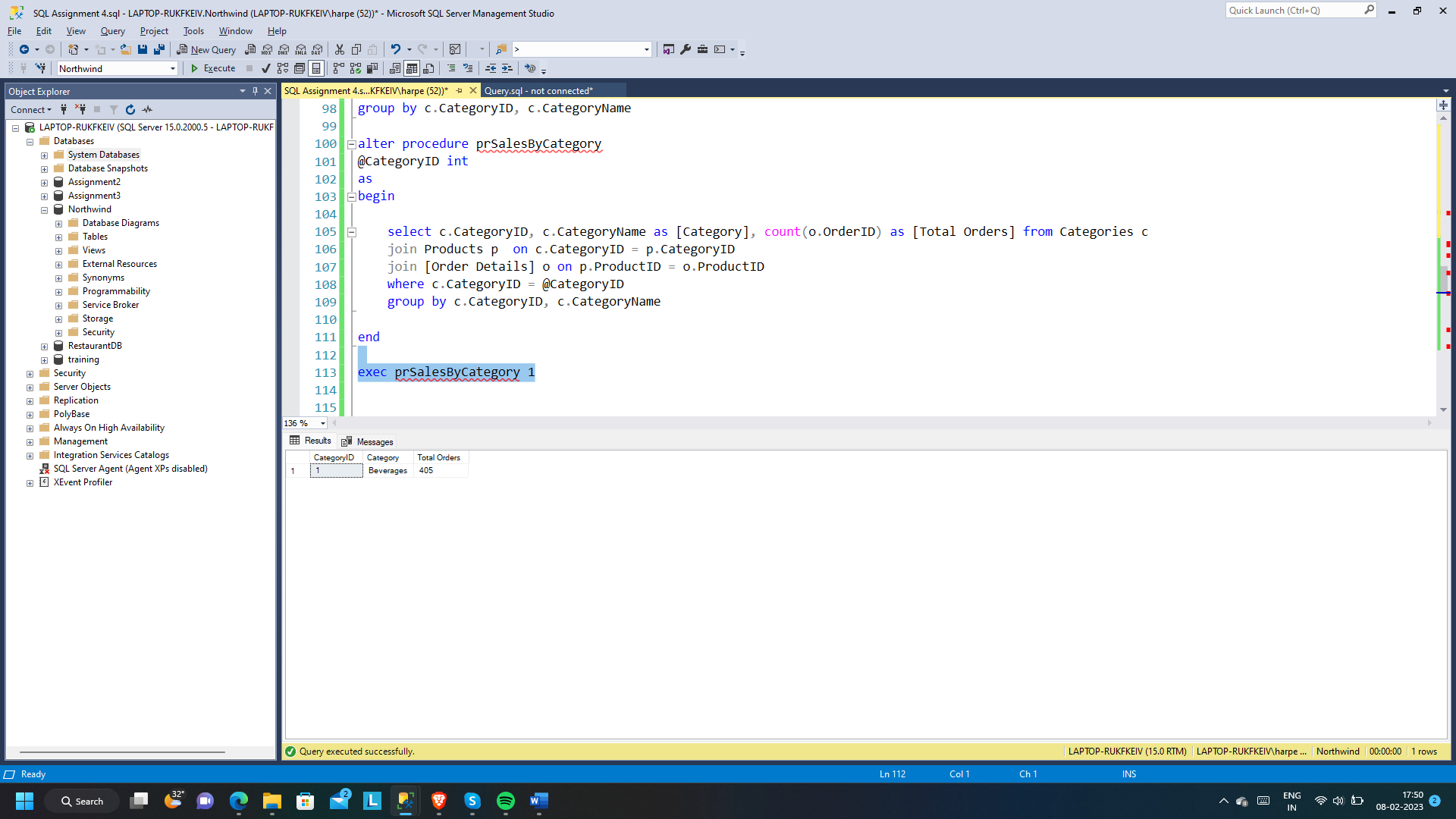
join [Order Details] o on p.ProductID = o.ProductID

where c.CategoryID = @CategoryID

group by c.CategoryID, c.CategoryName

end

exec prSalesByCategory 1



1. **write a SQL query to Create Stored procedure in the Northwind database to retrieve Ten Most Expensive Products**

create procedure prTopMostExpencive

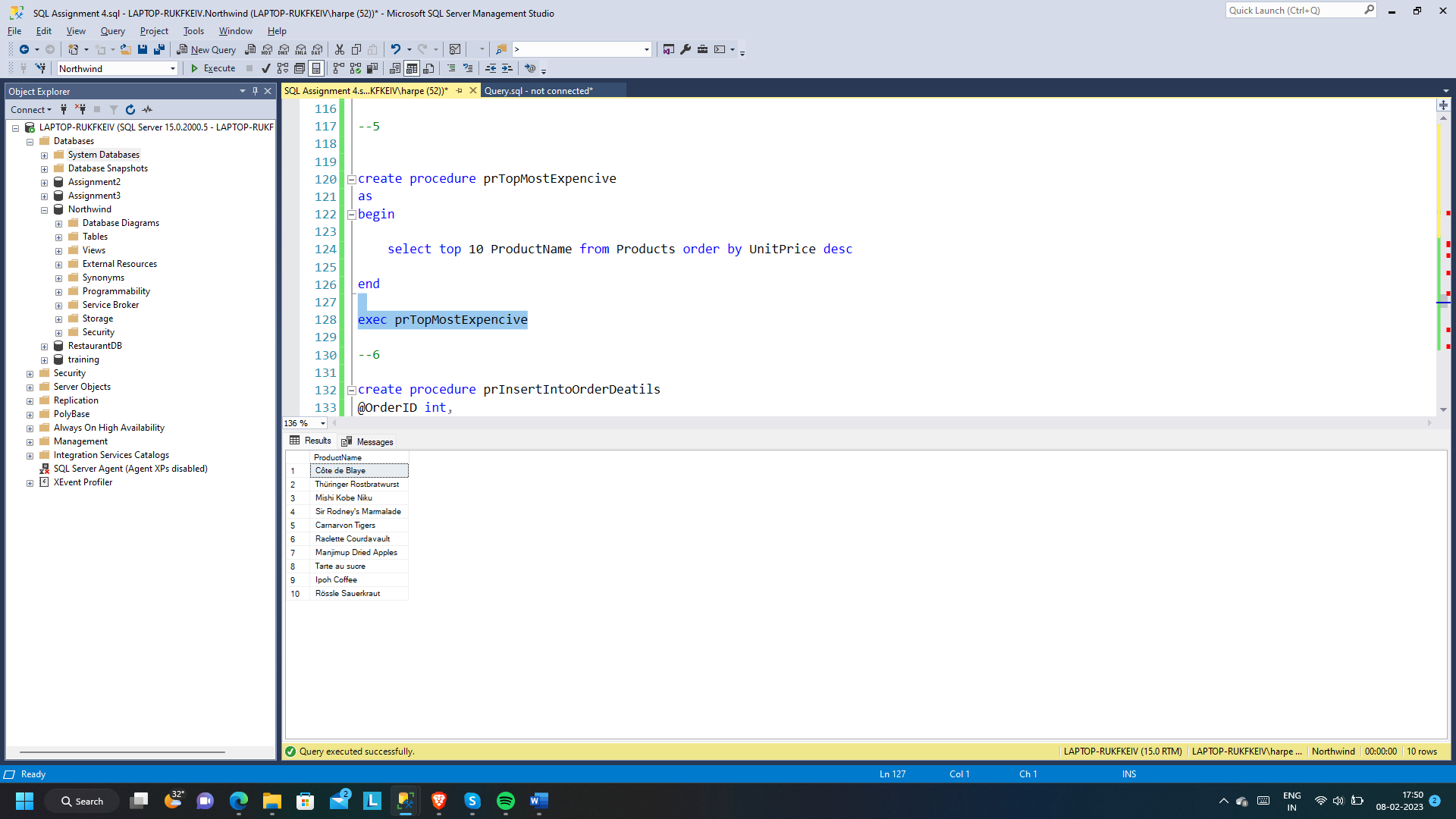
as

begin

select top 10 ProductName from Products order by UnitPrice desc

end

exec prTopMostExpencive



1. **write a SQL query to Create Stored procedure in the Northwind database to insert Customer Order Details**

create procedure prInsertIntoOrderDeatils

@OrderID int,

@ProductID int,

@UnitPrice decimal,

@Quantity int,

@Discount decimal

as

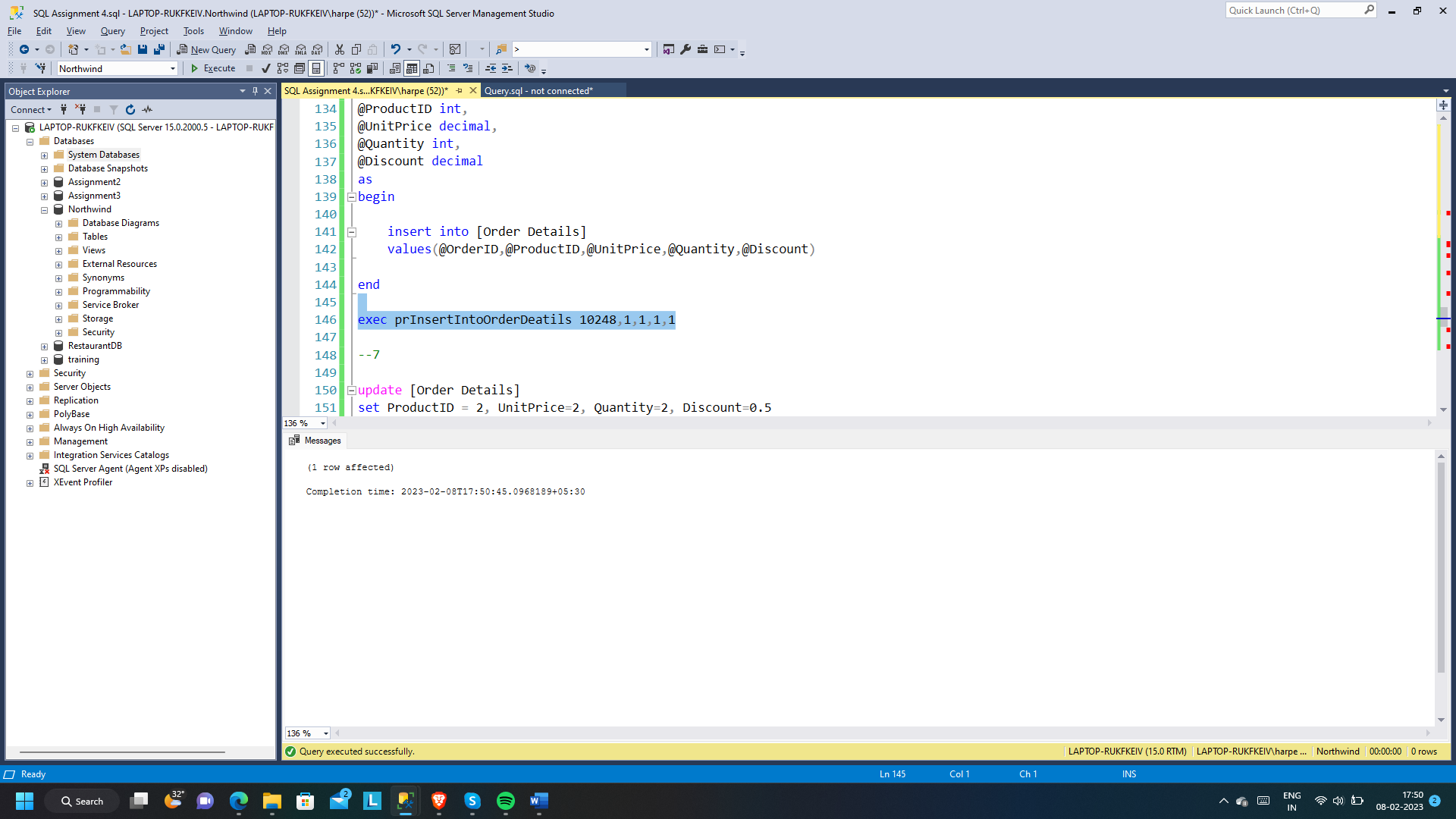
begin

insert into [Order Details]

values(@OrderID,@ProductID,@UnitPrice,@Quantity,@Discount)

end

exec prInsertIntoOrderDeatils 10248,1,1,1,1



1. **write a SQL query to Create Stored procedure in the Northwind database to update Customer Order Details**

update [Order Details]

set ProductID = 2, UnitPrice=2, Quantity=2, Discount=0.5

where OrderID = 10248 and ProductID = 1

alter procedure prUpdateOrderDetails

@OrderID int,

@ProductID int,

@UnitPrice decimal,

@Quantity int,

@Discount decimal

as

begin

update [Order Details]

set UnitPrice=@UnitPrice, Quantity=@Quantity, Discount=@Discount

where OrderID = @OrderID and ProductID=@ProductID

end

exec prUpdateOrderDetails 10248,2,5,65,0.3

