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.

create procedure spValidateFreight

-- inputted customer

@CustomerID nvarchar(5),

-- returned average freight

@AverageFreight money output

as

begin

select @AverageFreight = AVG(Freight)

from Orders

where CustomerID = @CustomerID

end

Declare @AvgFreight int;

execute spValidateFreight VINET, @AvgFreight output;

Print @AvgFreight

Create trigger trVerifyFreightForInsert

on Orders

Instead of insert

as

begin

Declare @AvgFreightOfOrders money

Declare @CustID nchar(5)

Declare @Freight money

Select @CustId=CustomerID from inserted

Select @Freight=Freight from inserted

-- execute stored procedure

exec spValidateFreight @CustID,

@AverageFreight = @AvgFreightOfOrders output

-- check the freight

if @AvgFreightOfOrders is not null

and @AvgFreightOfOrders < @Freight

begin

Raiserror('Invalid data as Freight value exceeds the average freight value',16,1)

return

end

end

Create trigger trVerifyFreightUpdate

on Orders

Instead of update

as

begin

Declare @AvgFreightOfOrders money

Declare @CustID nchar(5)

Declare @Freight money

Select @CustId=CustomerID from inserted

Select @Freight=Freight from inserted

-- execute stored procedure

exec spValidateFreight @CustID,

@AverageFreight = @AvgFreightOfOrders output

-- check the freight

if @AvgFreightOfOrders is not null

and @AvgFreightOfOrders < @Freight

begin

Raiserror('Invalid data as Freight value exceeds the average freight value',16,1)

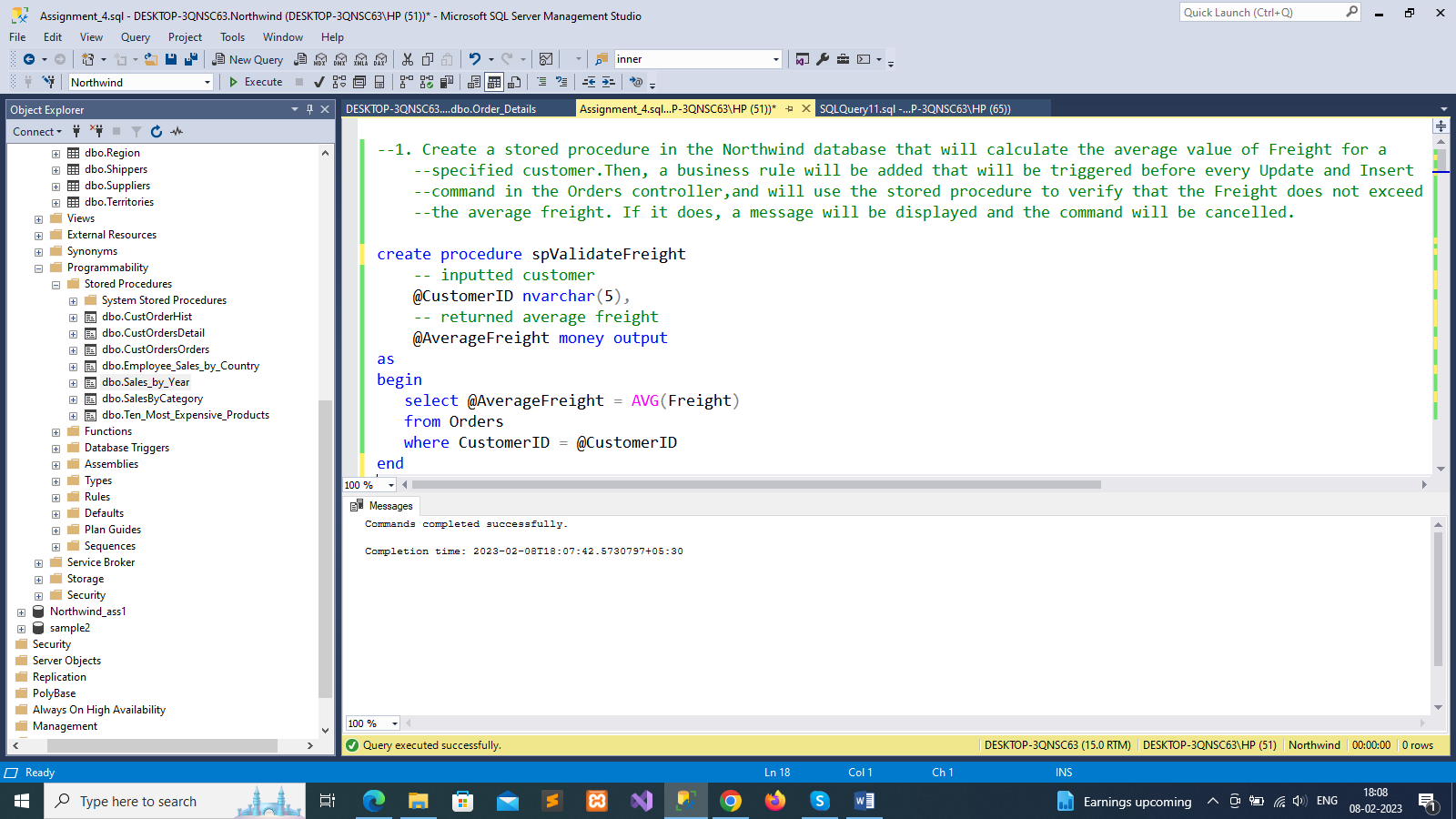
return

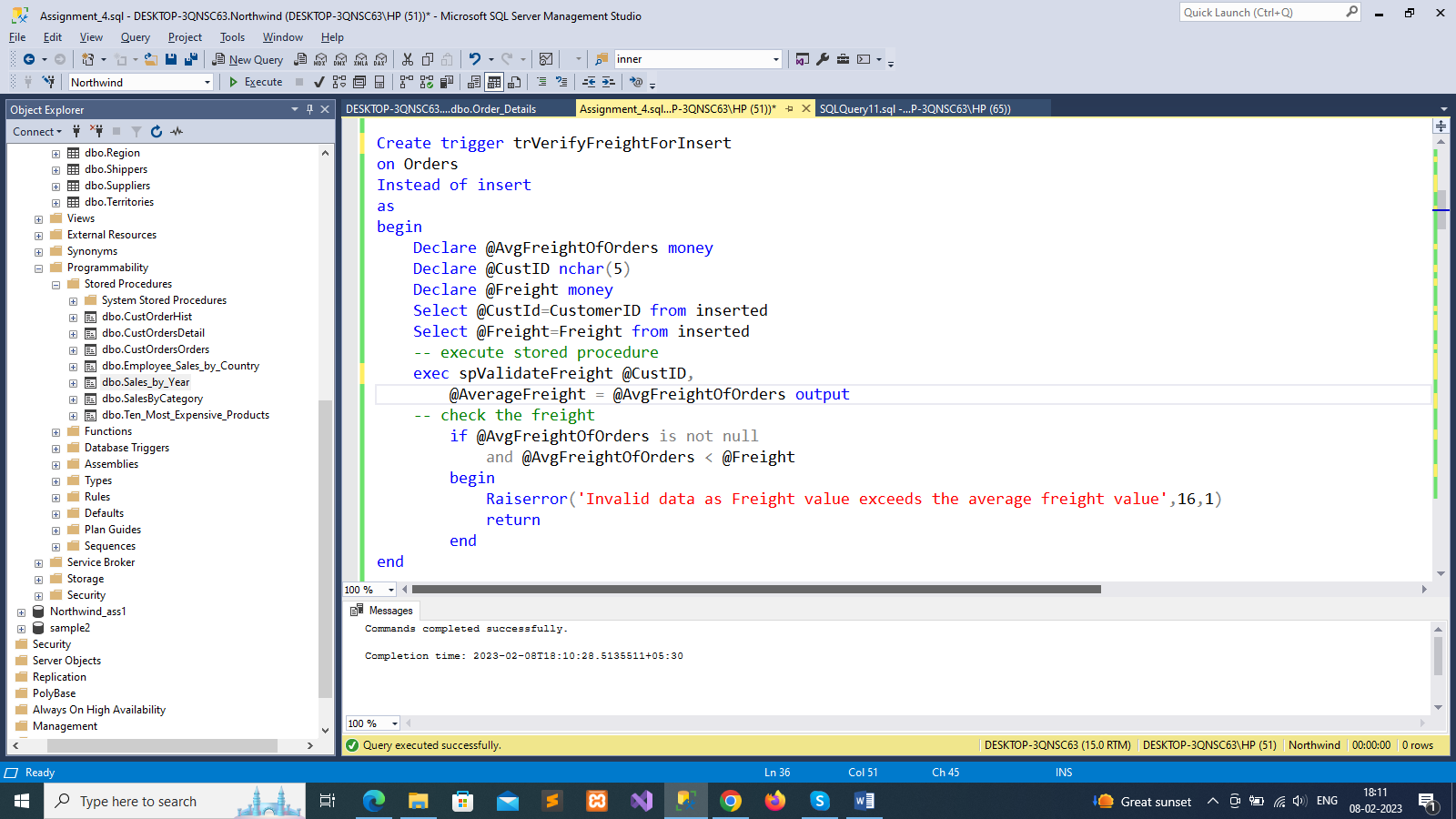
end

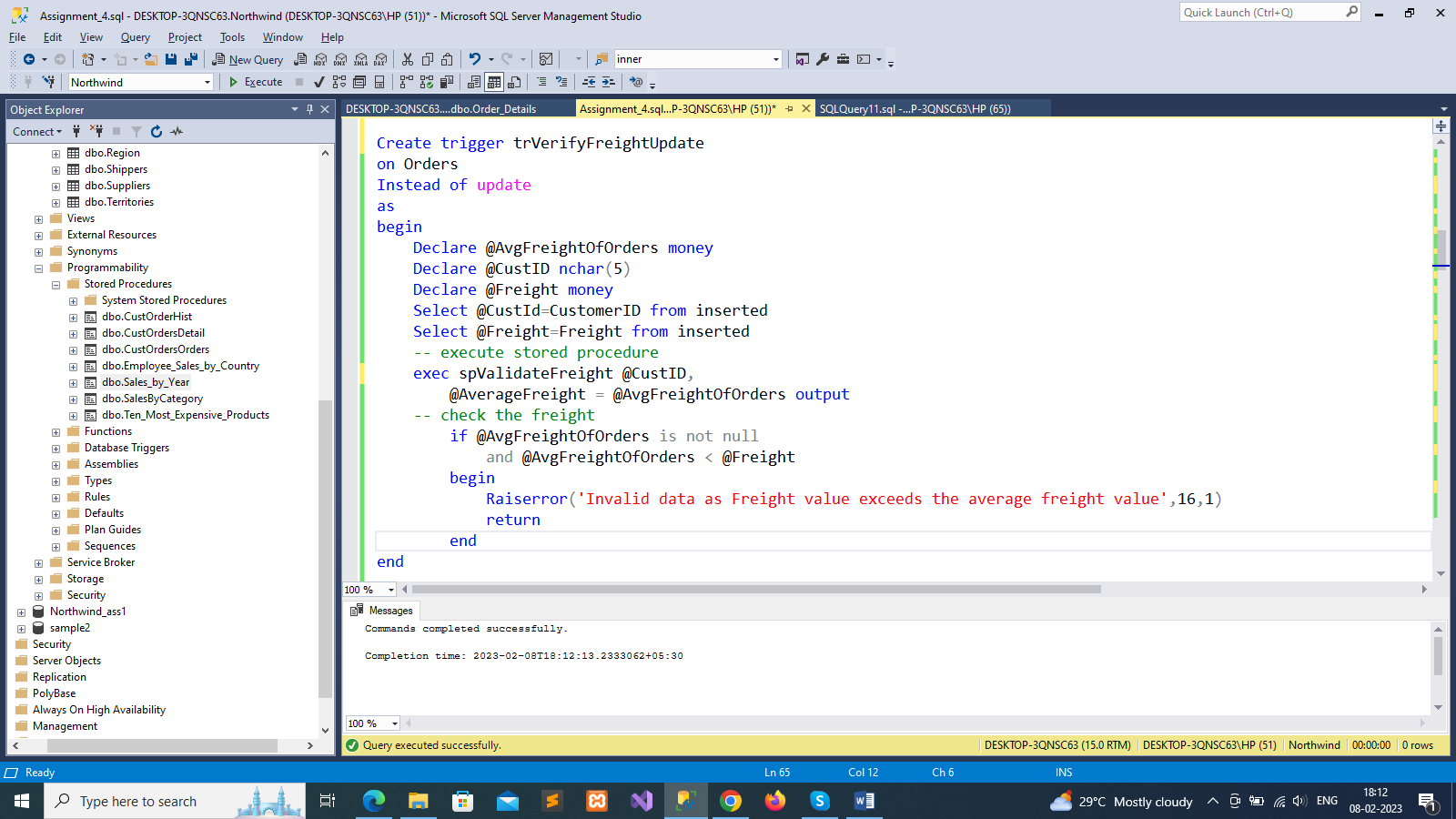
end

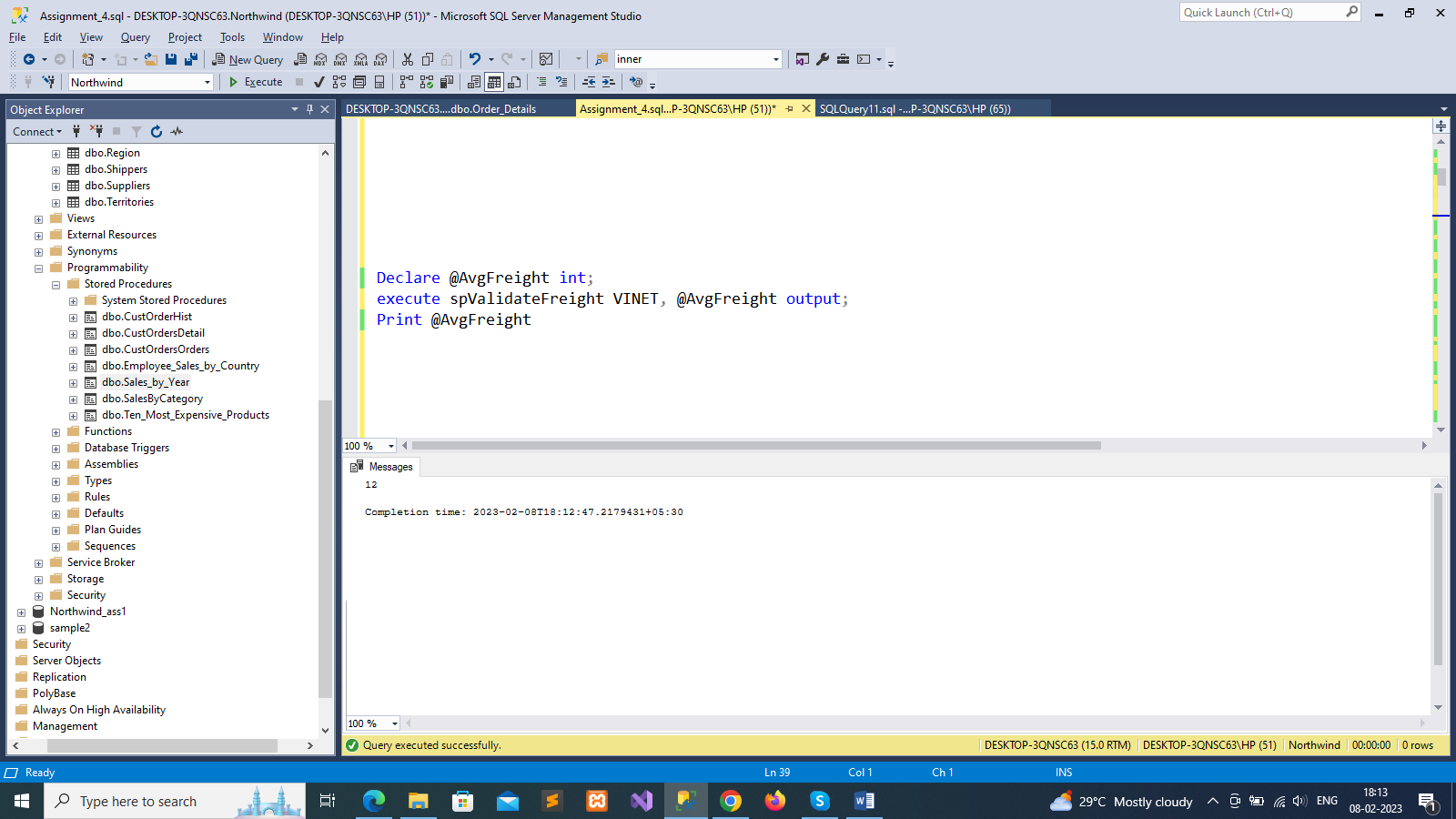
Insert into Orders values('VINET',null,null,null,null,null,10,null,null,null,null,null,null);

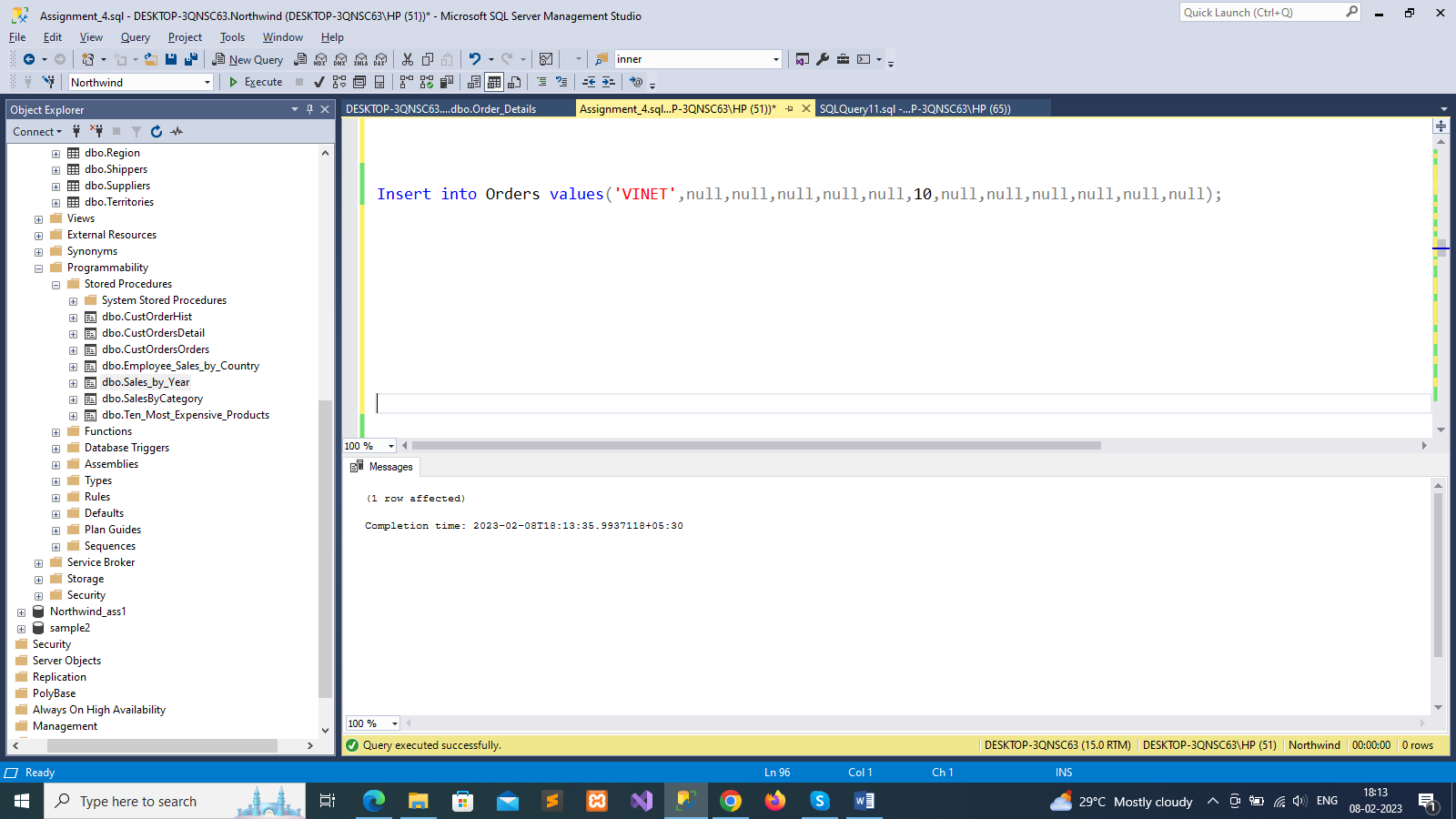
Insert into Orders values('VINET',null,null,null,null,null,26,null,null,null,null,null,null);

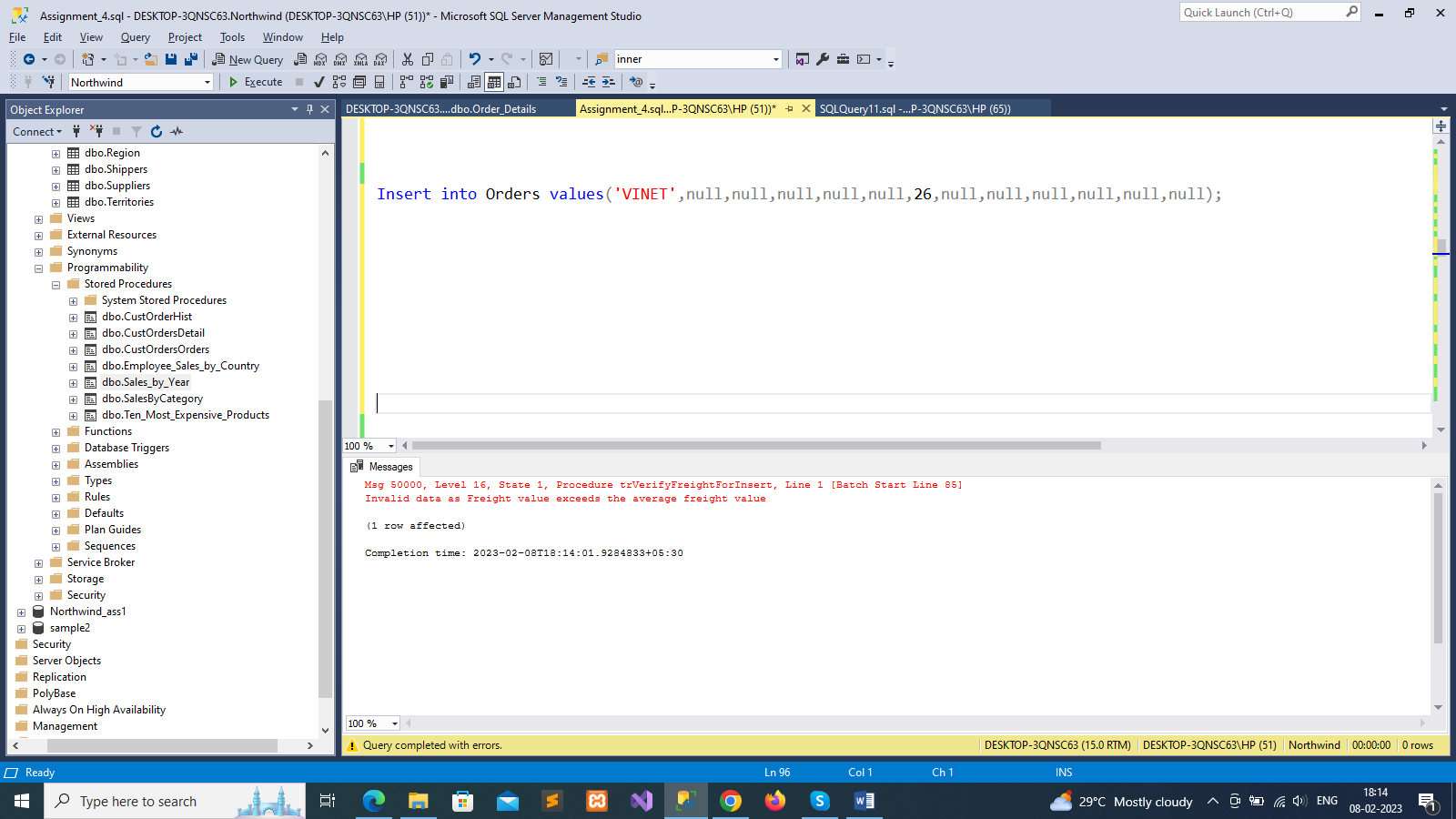












--2. write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country

create procedure spQue\_2

(

@Beginning\_Date Date,

@Ending\_Date Date

)

as

begin

select Employees.Country, Employees.FirstName,Employees.LastName, Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal AS SaleAmount

from Employees

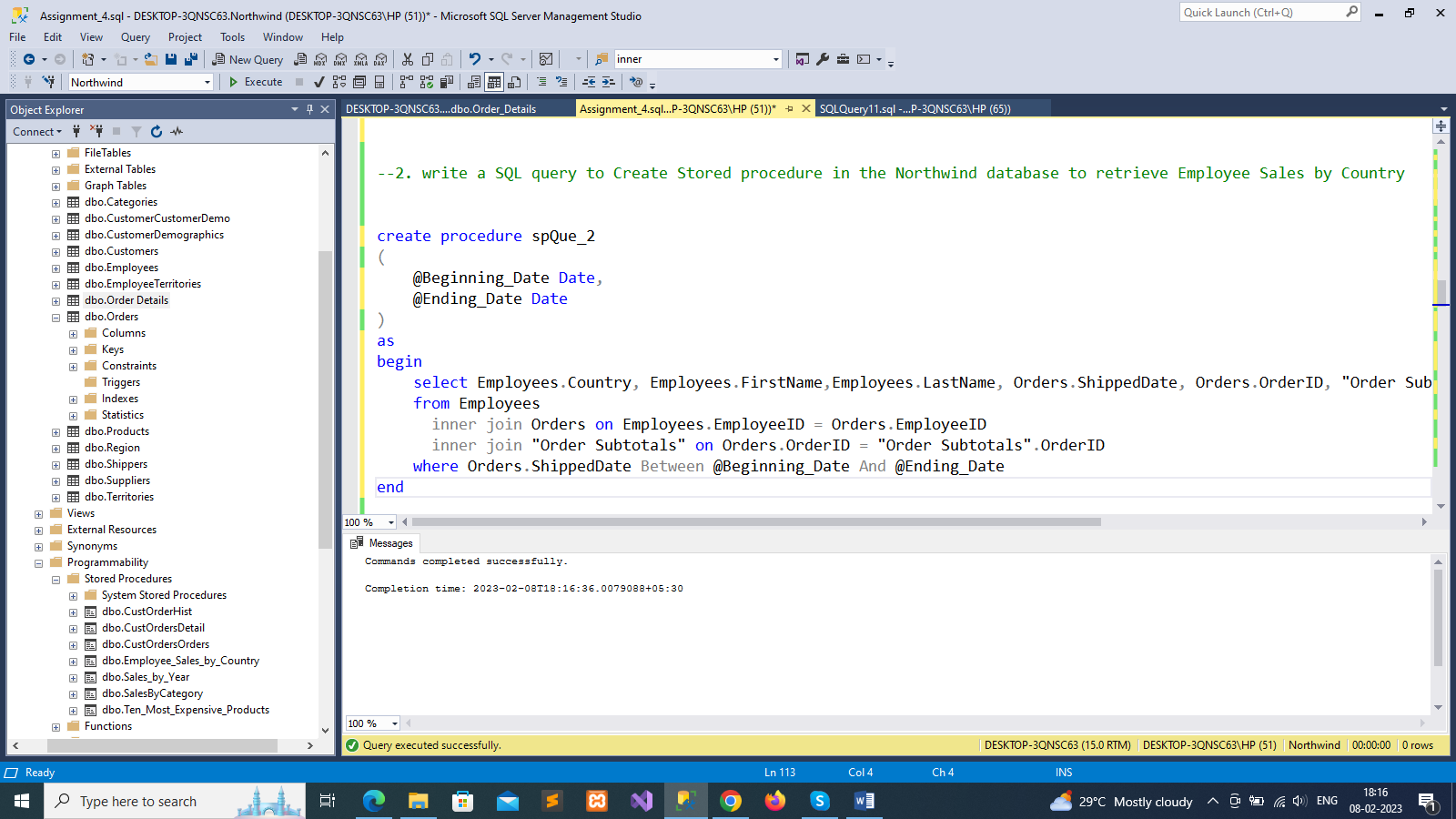
inner join Orders on Employees.EmployeeID = Orders.EmployeeID

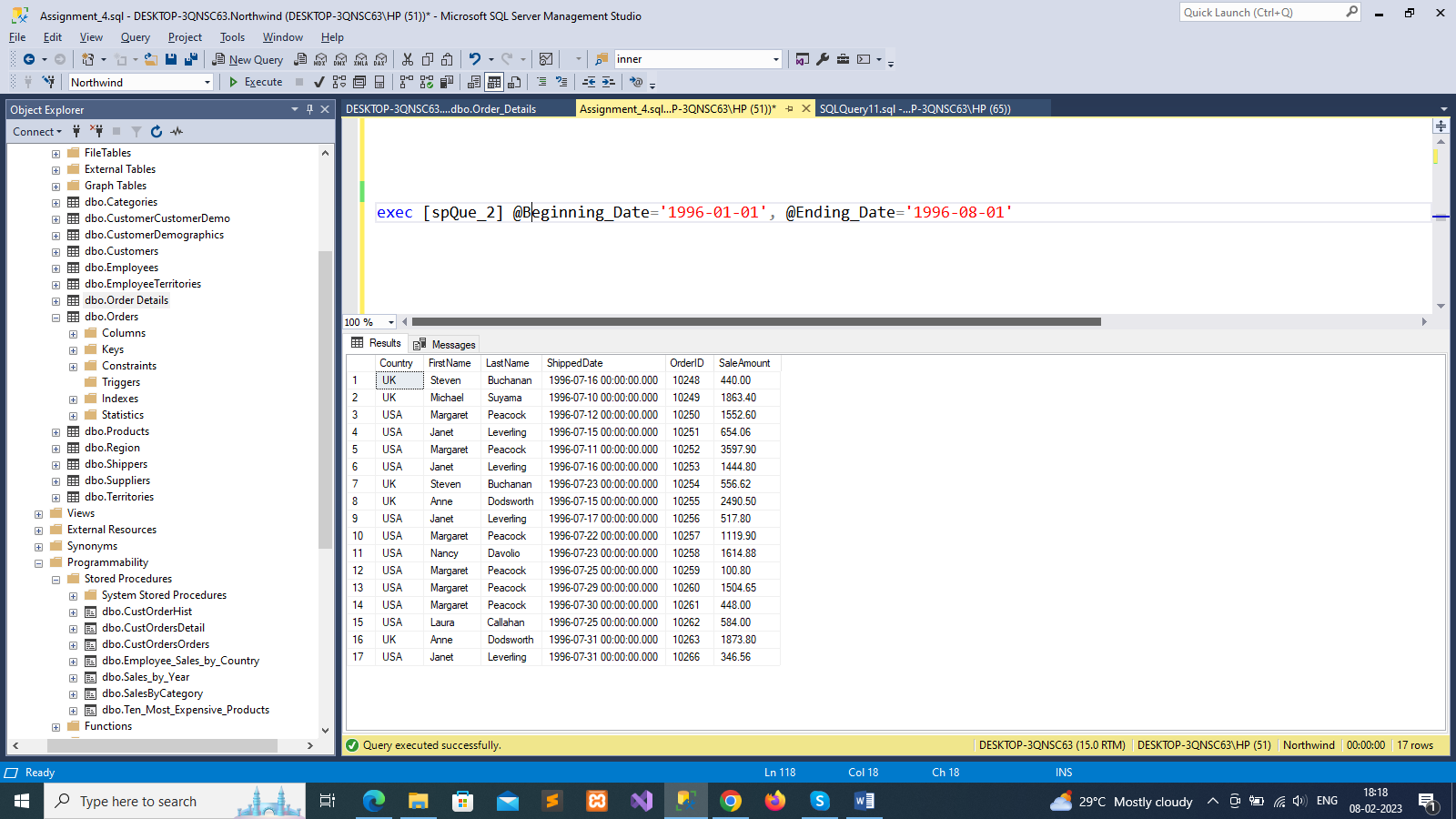
inner join "Order Subtotals" on Orders.OrderID = "Order Subtotals".OrderID

where Orders.ShippedDate Between @Beginning\_Date And @Ending\_Date

end

exec [spQue\_2] @Beginning\_Date='1996-01-01', @Ending\_Date='1996-08-01'





--3. write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales by Year

create procedure spQue\_3

(

@Beginning\_Date Date,

@Ending\_Date Date

)

as

begin

select Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal, DATENAME(yy,ShippedDate) AS Year

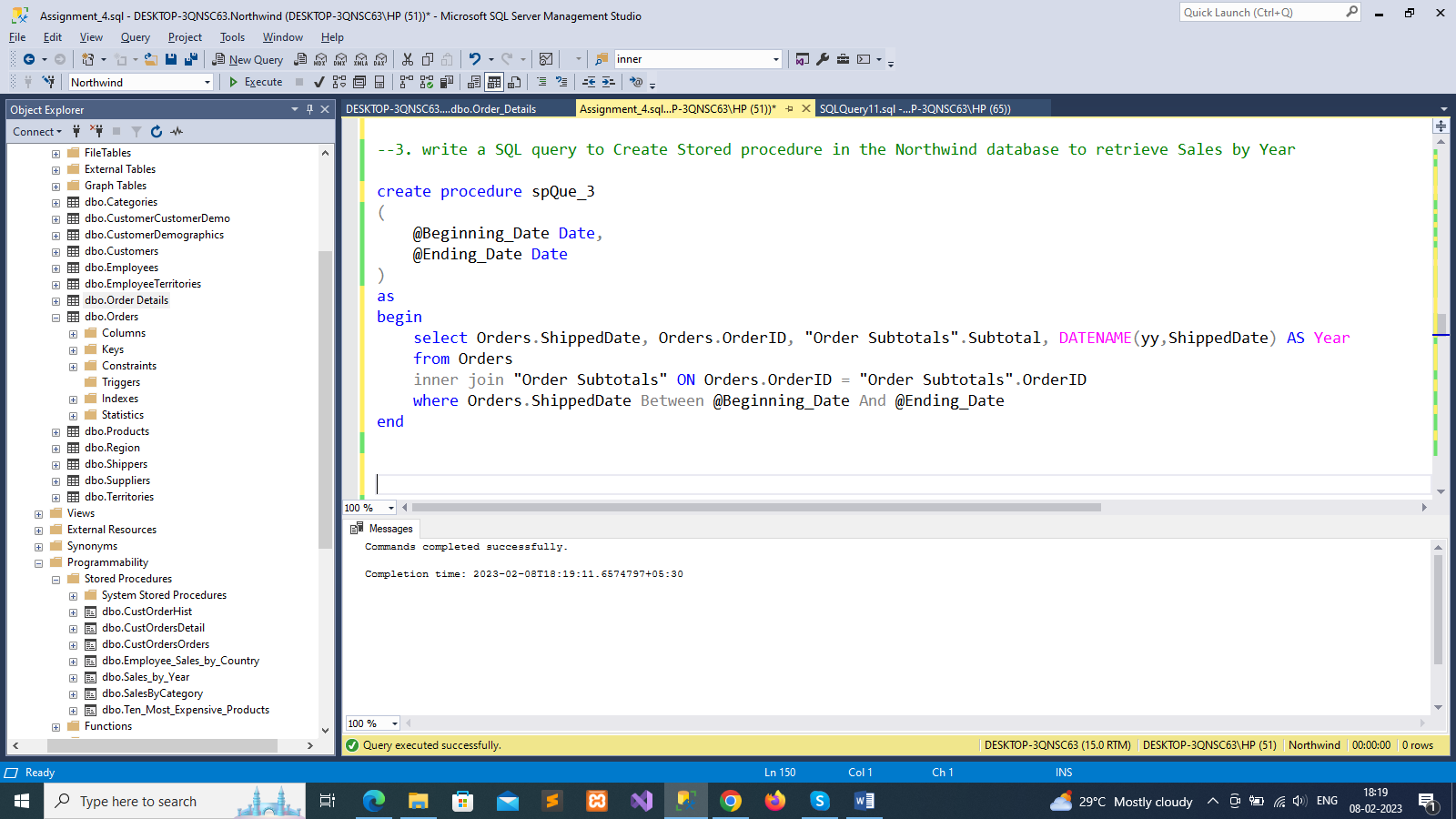
from Orders

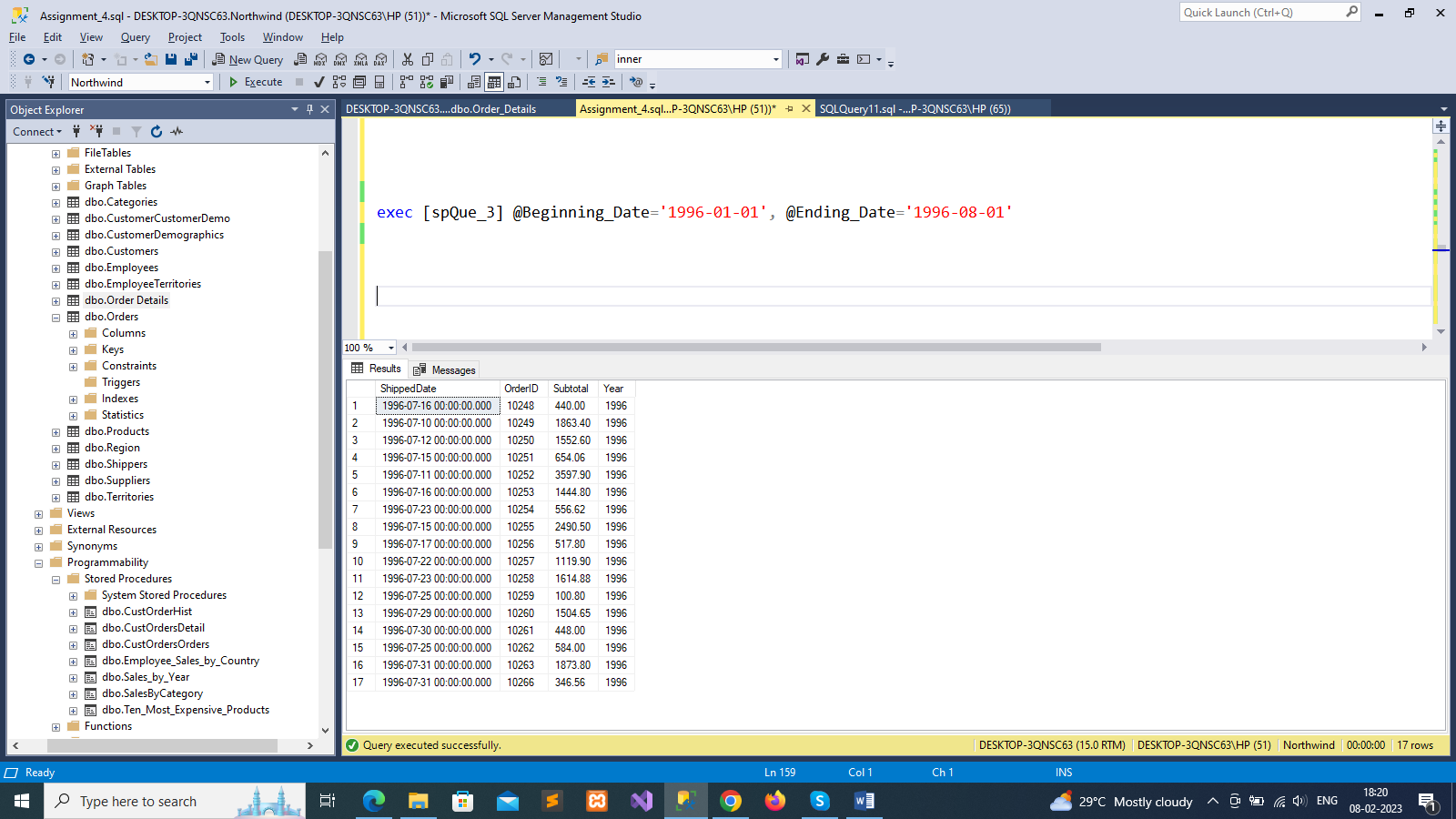
inner join "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID

where Orders.ShippedDate Between @Beginning\_Date And @Ending\_Date

end

exec [spQue\_3] @Beginning\_Date='1996-01-01', @Ending\_Date='1996-08-01'





--4. write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales By Category

create procedure spQue\_4

(

@CategoryName nvarchar(15),

@OrdYear nvarchar(4) = '1998'

)

as

begin

if @OrdYear != '1996' AND @OrdYear != '1997' AND @OrdYear != '1998'

begin

select @OrdYear = '1998'

end

select ProductName,TotalPurchase=round(sum(convert(decimal(14,2), OD.Quantity \* (1-OD.Discount) \* OD.UnitPrice)), 0)

from [Order Details] OD, Orders O, Products P, Categories C

where OD.OrderID = O.OrderID

and OD.ProductID = P.ProductID

and P.CategoryID = C.CategoryID

and C.CategoryName = @CategoryName

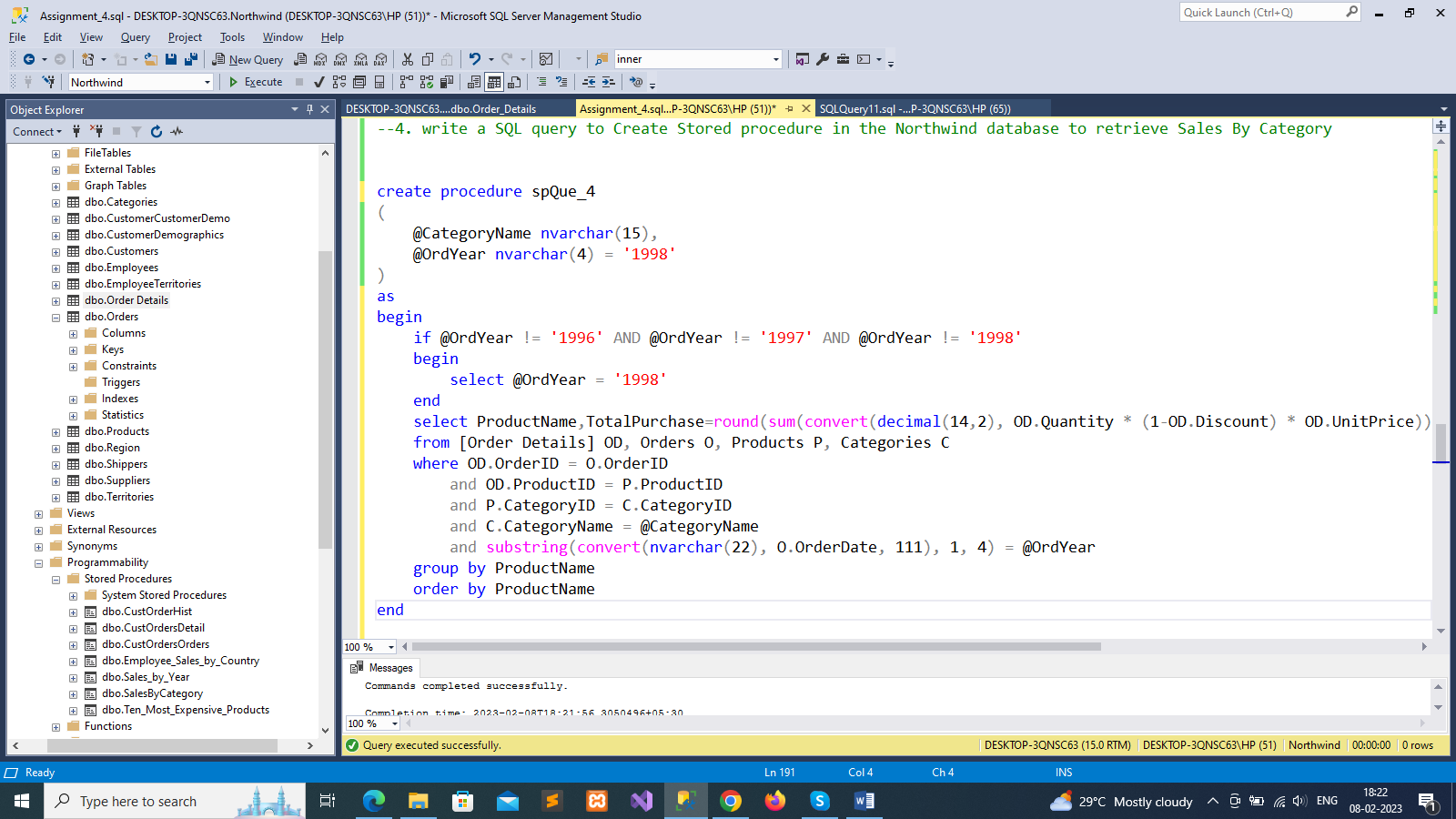
and substring(convert(nvarchar(22), O.OrderDate, 111), 1, 4) = @OrdYear

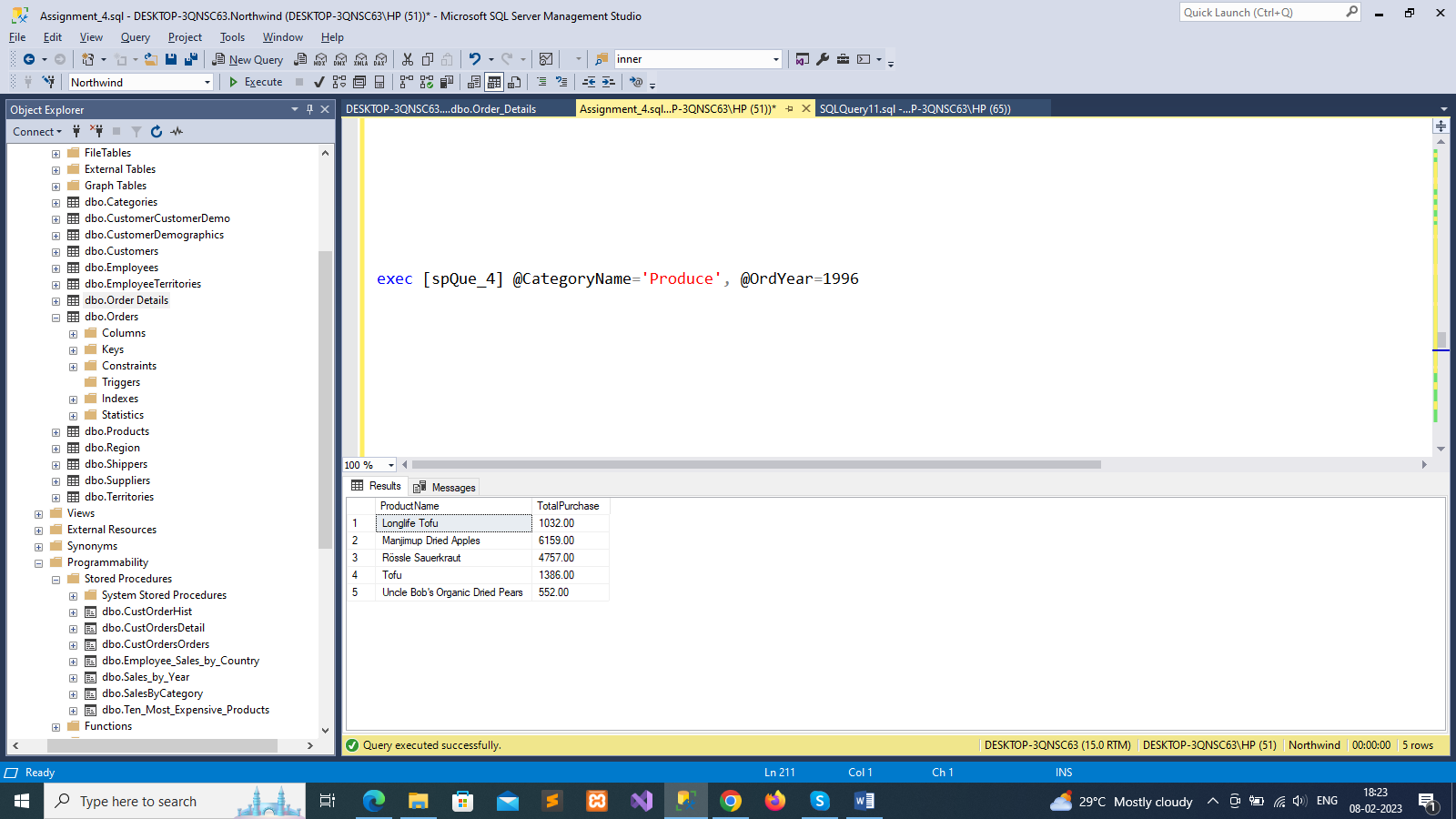
group by ProductName

order by ProductName

end

exec [spQue\_4] @CategoryName='Produce', @OrdYear=1996





--5. write a SQL query to Create Stored procedure in the Northwind database to retrieve Ten Most Expensive Products

create procedure spQue\_5

as

begin

set rowcount 10

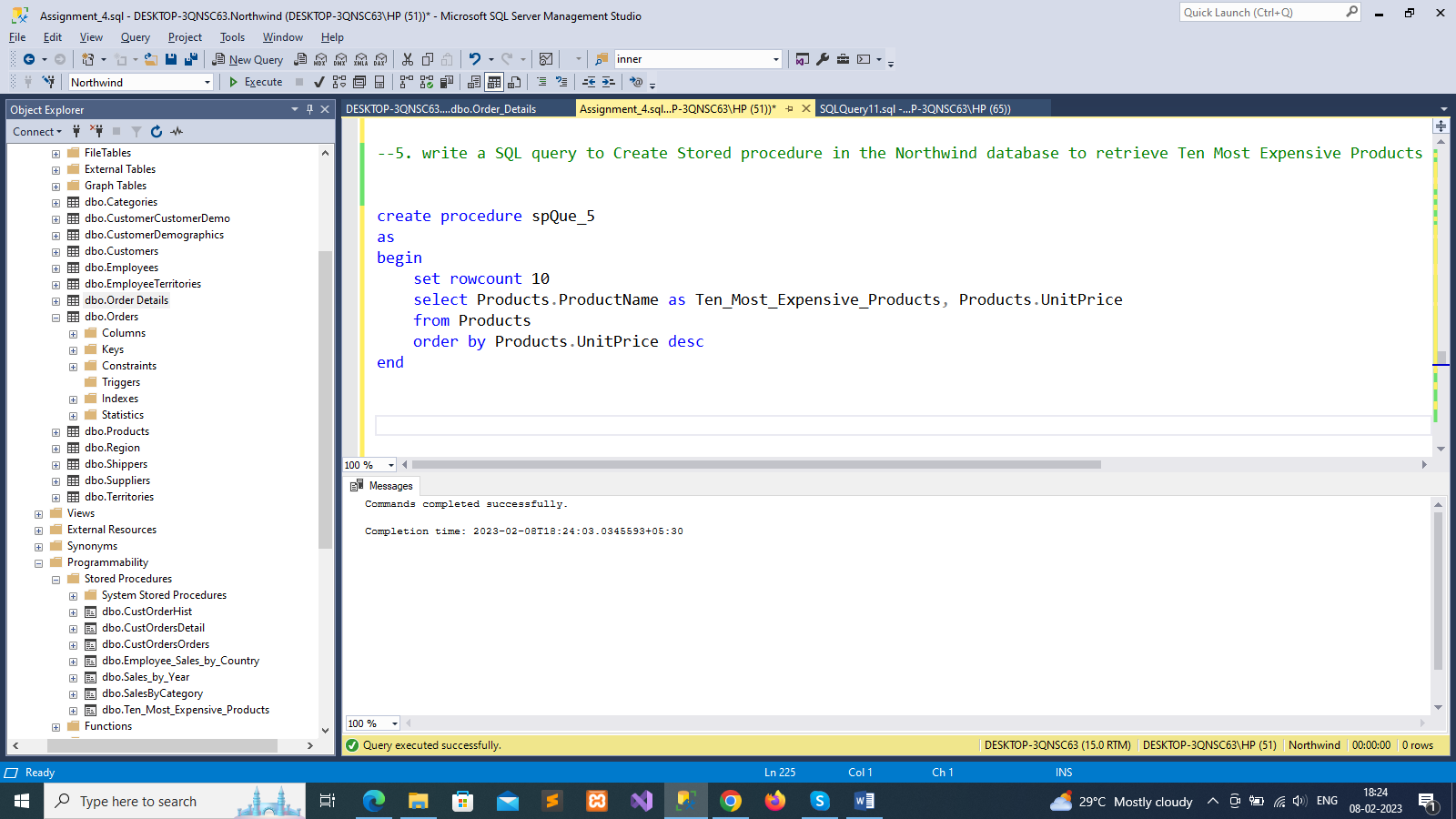
select Products.ProductName as Ten\_Most\_Expensive\_Products, Products.UnitPrice

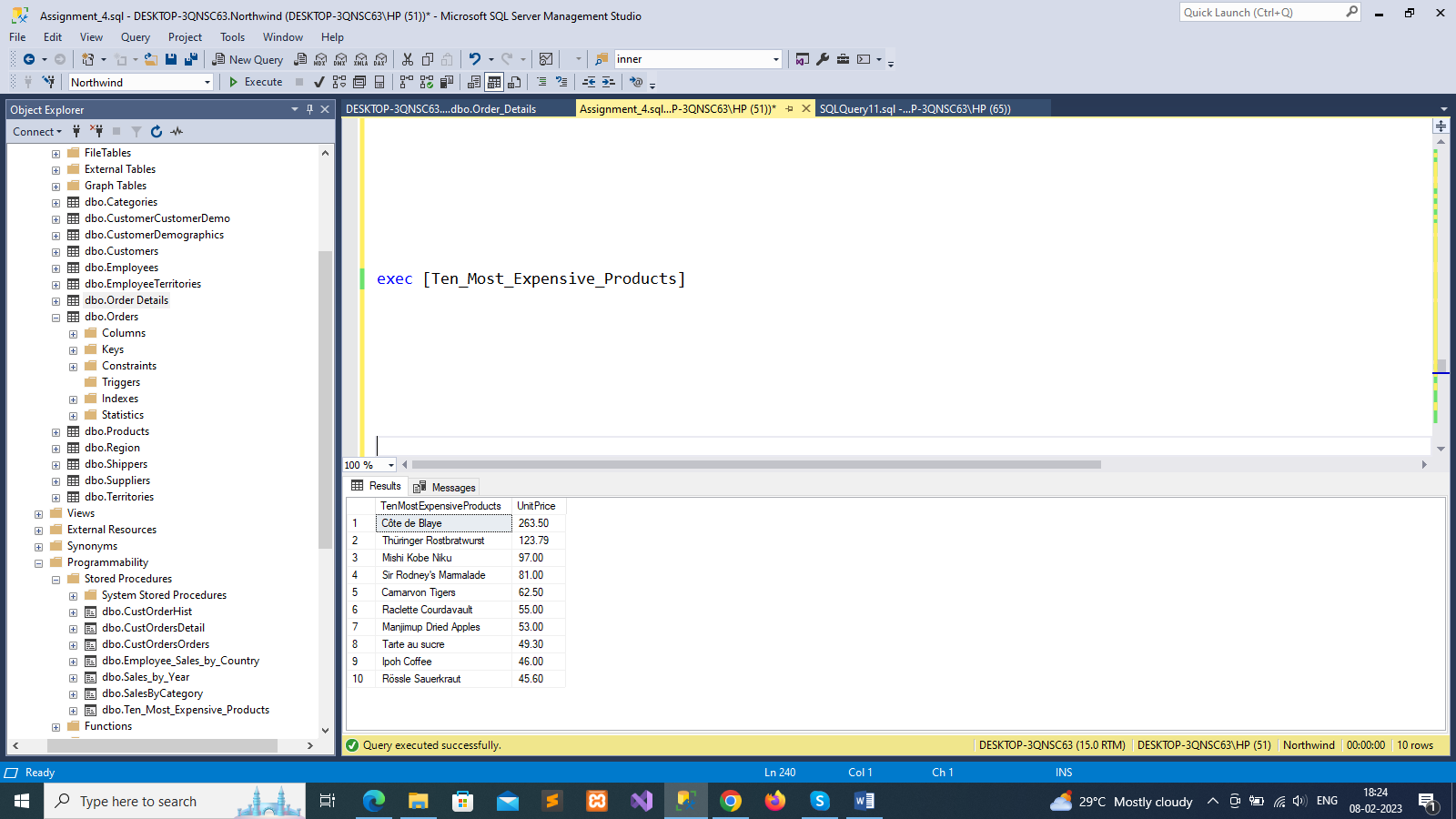
from Products

order by Products.UnitPrice desc

end

exec [Ten\_Most\_Expensive\_Products]





--6. write a SQL query to Create Stored procedure in the Northwind database to insert Customer Order Details

create or alter procedure spQue\_6

(

@OrderID int,

@ProductID int,

@UnitPrice money,

@Quantity smallint,

@Discount real

)

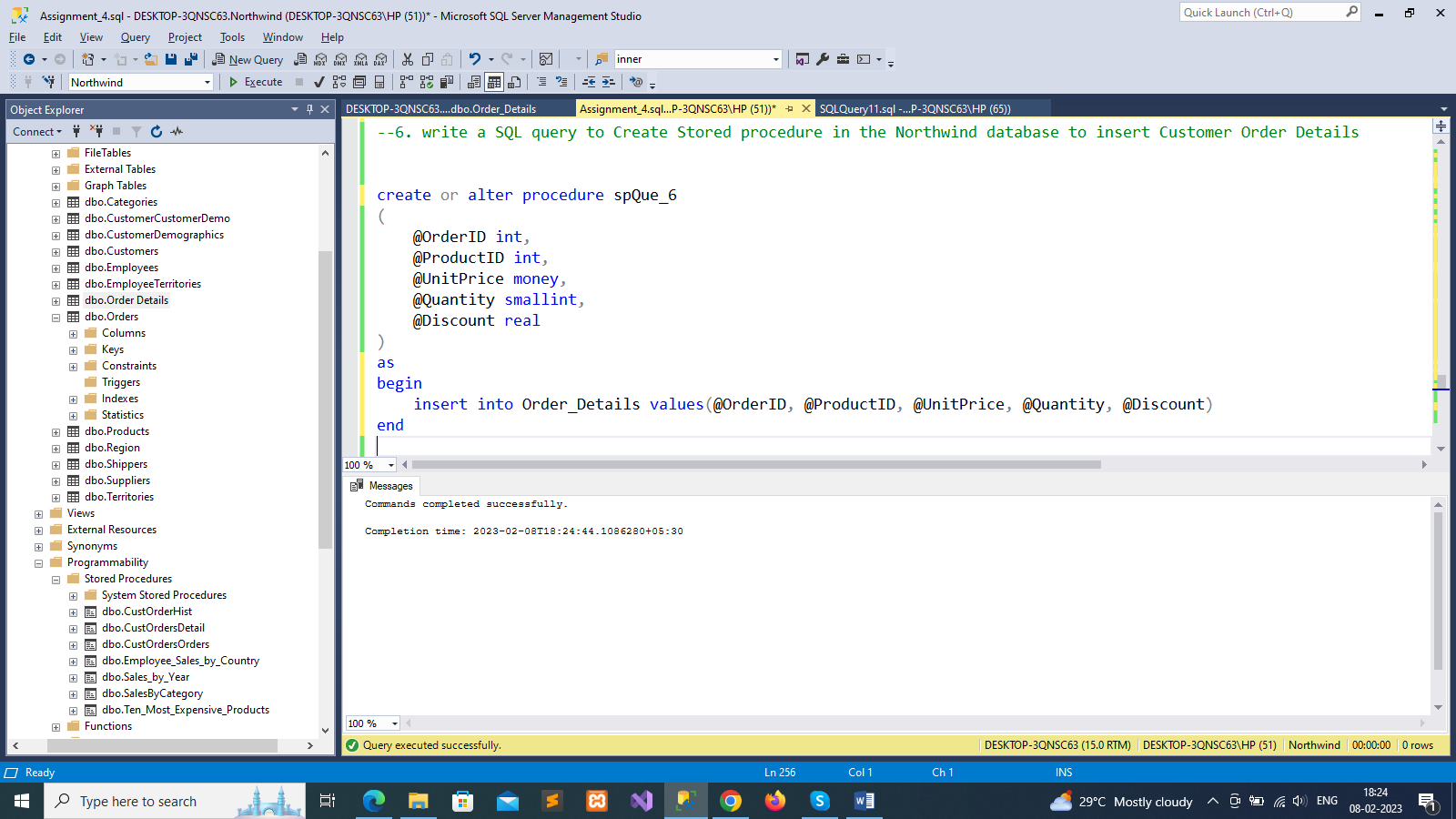
as

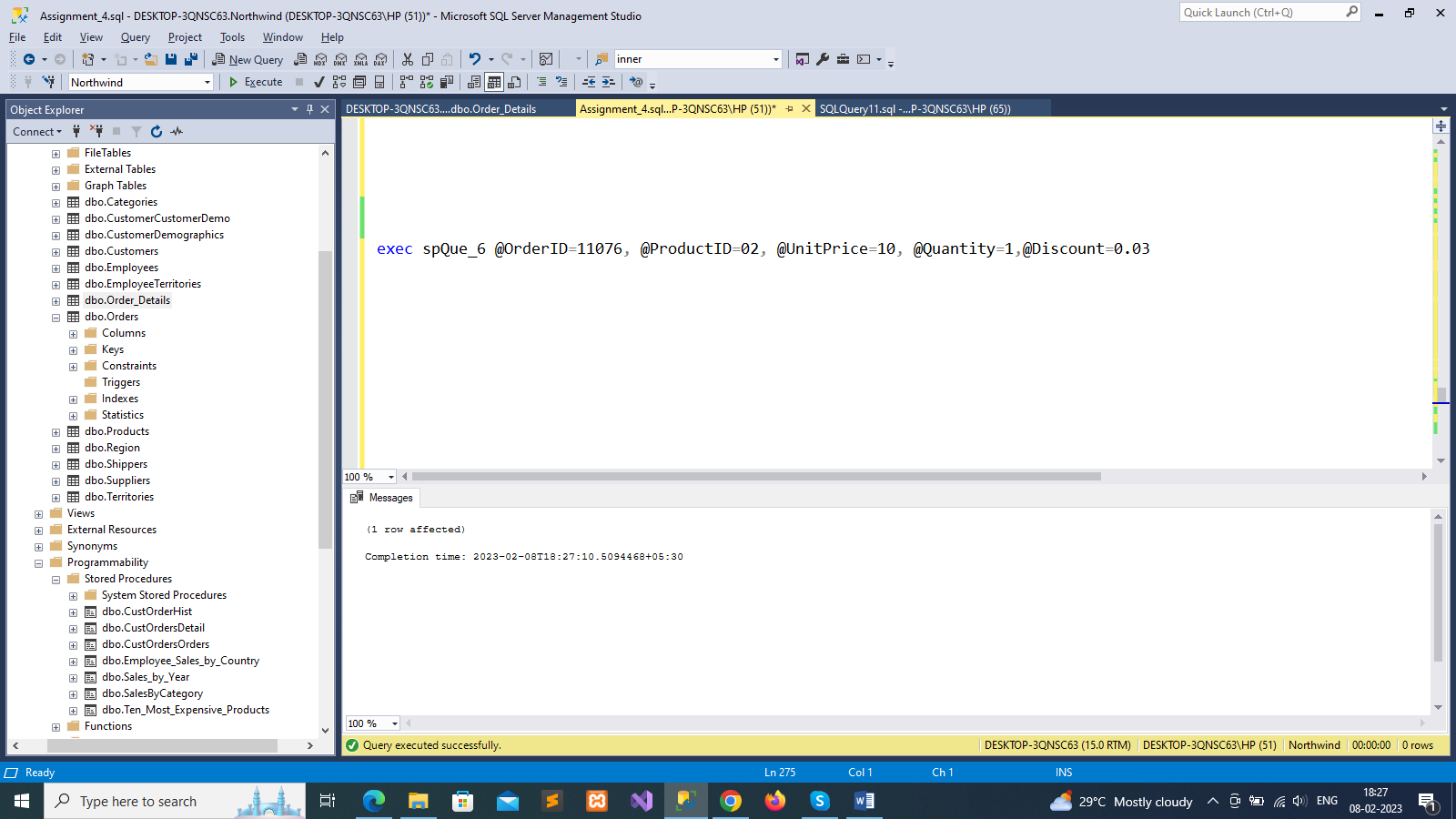
begin

insert into Order\_Details values(@OrderID, @ProductID, @UnitPrice, @Quantity, @Discount)

end

exec spQue\_6 @OrderID=11076, @ProductID=02, @UnitPrice=10, @Quantity=1,@Discount=0.03





--7. write a SQL query to Create Stored procedure in the Northwind database to update Customer Order Details

create or alter procedure spQue\_7

(

@OrderID int,

@ProductID int,

@UnitPrice money,

@Quantity smallint,

@Discount real

)

as

begin

update Order\_Details set UnitPrice=@UnitPrice, Quantity=@Quantity, Discount=@Discount

where OrderID=@OrderID and ProductID=@ProductID

end

exec spQue\_7 @OrderID=11076, @ProductID=02, @UnitPrice=11, @Quanti

