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 or Alter proc sp\_avgfreightofcstmr

@cust\_id varchar(10),

@cust\_freight float output

As

Begin

Select @cust\_freight = avg(o.Freight)

from Customers c

Inner Join Orders o

On c.CustomerID = o.CustomerID and c.CustomerID = @cust\_id

End

declare @output float

Exec sp\_avgfreightofcstmr 'VINET',@output out

print (@output)

Create or Alter Trigger tr\_Orders\_ForCheckInsert

on Orders

Instead Of Insert , Update

AS

Begin

Declare @CustomerId Varchar(10)

Declare @Freight Float

Declare @avg\_freight\_customer float

Select @CustomerId = CustomerID from inserted

Select @Freight = Freight from inserted

Exec sp\_avgfreightofcstmr @CustomerId,@avg\_freight\_customer out

print @avg\_freight\_customer

if (@avg\_freight\_customer < @Freight)

Begin

Print 'Data Not added to the table'

End

Else

Begin

If Exists(Select \* From inserted)

Begin

Declare @OrderID int

Declare @EmployeeID int

Declare @OrderDate DateTime

Declare @RequiredDate DateTime

Declare @Shippeddate DateTime

Declare @ShipName nvarchar(40)

Declare @ShipVia int

Declare @ShipAddress nvarchar(60)

Declare @ShipCity nvarchar(15)

Declare @ShipRegion nvarchar(15)

Declare @ShipPostalCode nvarchar(10)

Declare @ShipCountry nvarchar(15)

Select @OrderID = OrderID from inserted

Select @EmployeeID = EmployeeID from inserted

Select @OrderDate = OrderDate from inserted

Select @RequiredDate = RequiredDate from inserted

Select @Shippeddate = Shippeddate from inserted

Select @ShipVia = ShipVia from inserted

Select @Freight = Freight from inserted

Select @ShipName = ShipName from inserted

Select @ShipAddress = ShipAddress from inserted

Select @ShipCity = ShipCity from inserted

Select @ShipRegion = ShipRegion from inserted

Select @ShipPostalCode = ShipPostalCode from inserted

Select @ShipCountry = ShipCountry from inserted

If Not Exists(Select \* from deleted)

Begin

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

values (@CustomerId,@EmployeeID,@OrderDate,@RequiredDate,@ShippedDate,@ShipVia,@Freight,@ShipName,@ShipAddress,@ShipCity,@ShipRegion,@ShipPostalCode,@ShipCountry)

End

Else

Begin

If Not Exists (Select OrderID From Orders Where OrderID = @OrderID)

Begin

Print 'OrderID '+@OrderID+' is not exist in Orders Table'

End

Else If Exists (Select OrderID From Orders Where OrderID = @OrderID)

Begin

Update Orders

Set CustomerID=@CustomerID,

EmployeeID=@EmployeeID,

OrderDate = @OrderDate,

RequiredDate = @RequiredDate,

ShippedDate = @ShippedDate,

ShipVia = @ShipVia,

Freight = @Freight,

ShipName = @ShipName,

ShipAddress = @ShipAddress,

ShipCity = @ShipCity,

ShipRegion = @ShipRegion,

ShipPostalCode = @ShipPostalCode,

ShipCountry = @ShipCountry

Where OrderId = @OrderID

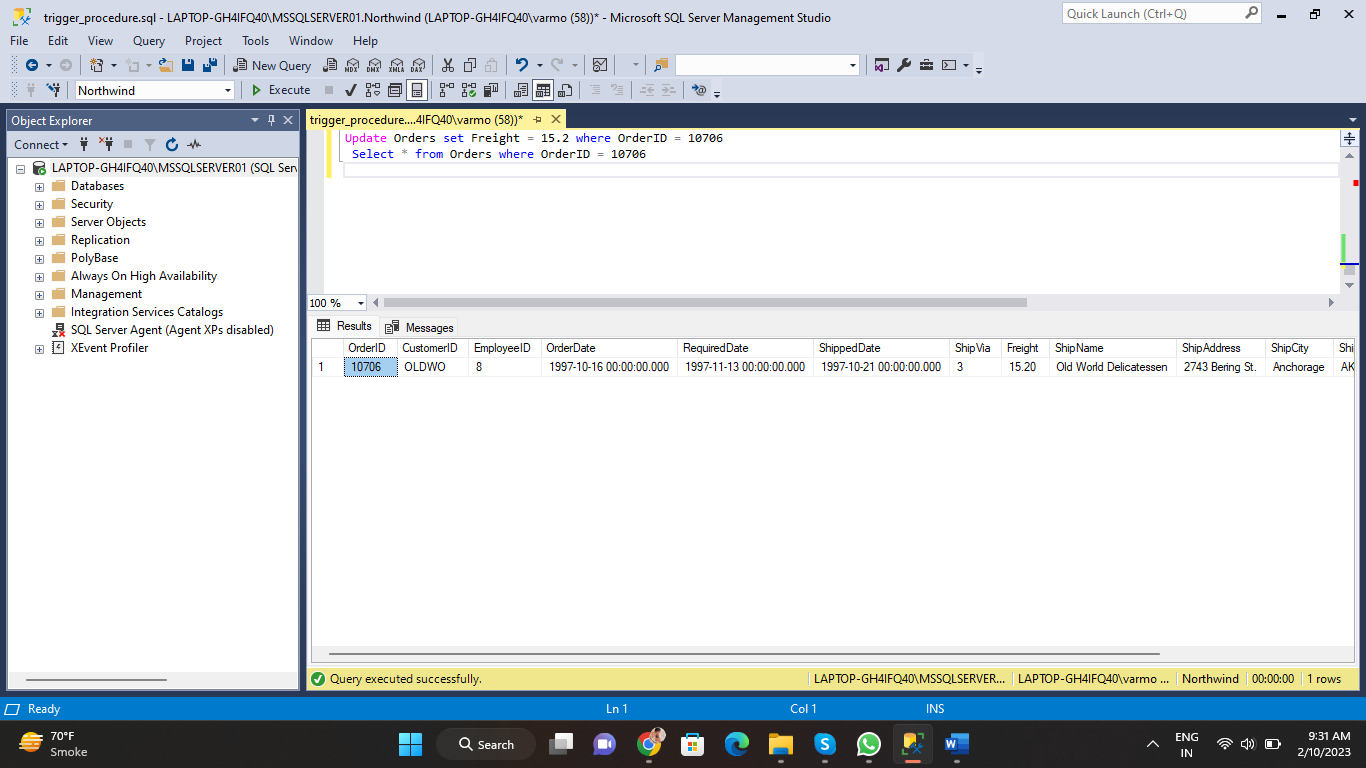
End

End

End

End

End



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

CREATE PROCEDURE usp\_Employees\_Sales\_By\_Country

AS

BEGIN

SELECT e.EmployeeID AS 'ID',

e.FirstName + ' ' + e.LastName AS 'Employee Name',

o.ShipCountry AS 'Country',

ROUND(SUM(od.UnitPrice \* od.quantity - (od.UnitPrice \* od.quantity \* od.discount)), 2) AS 'Total Sale'

FROM Employees e

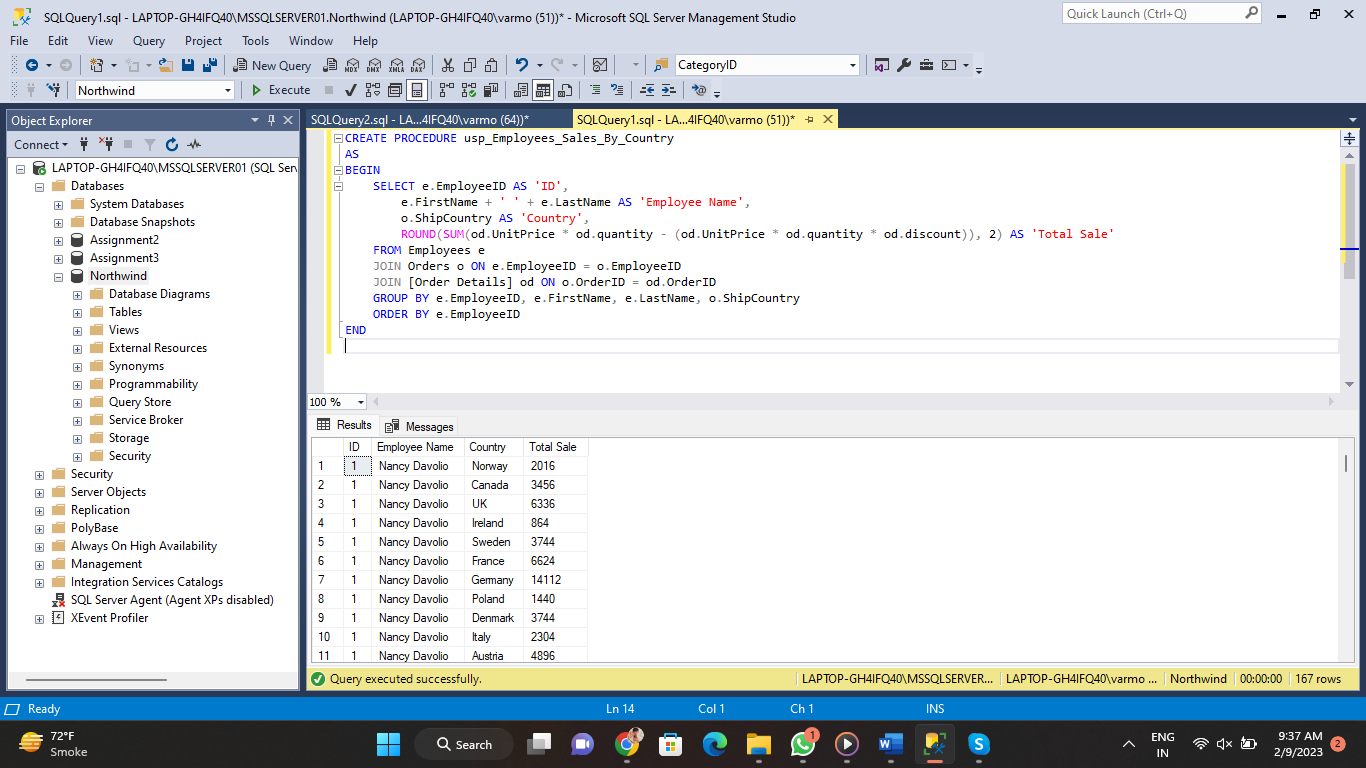
JOIN Orders o ON e.EmployeeID = o.EmployeeID

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

GROUP BY e.EmployeeID, e.FirstName, e.LastName, o.ShipCountry

ORDER BY e.EmployeeID

END



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

Create or Alter Procedure sp\_SalesByCountry

As

Begin

Select Datename(Year,o.ShippedDate),

Round(Sum(od.UnitPrice \* od.Quantity - (od.UnitPrice \* od.Quantity \* od.Discount)), 2) As "Total Sale"

From Orders o

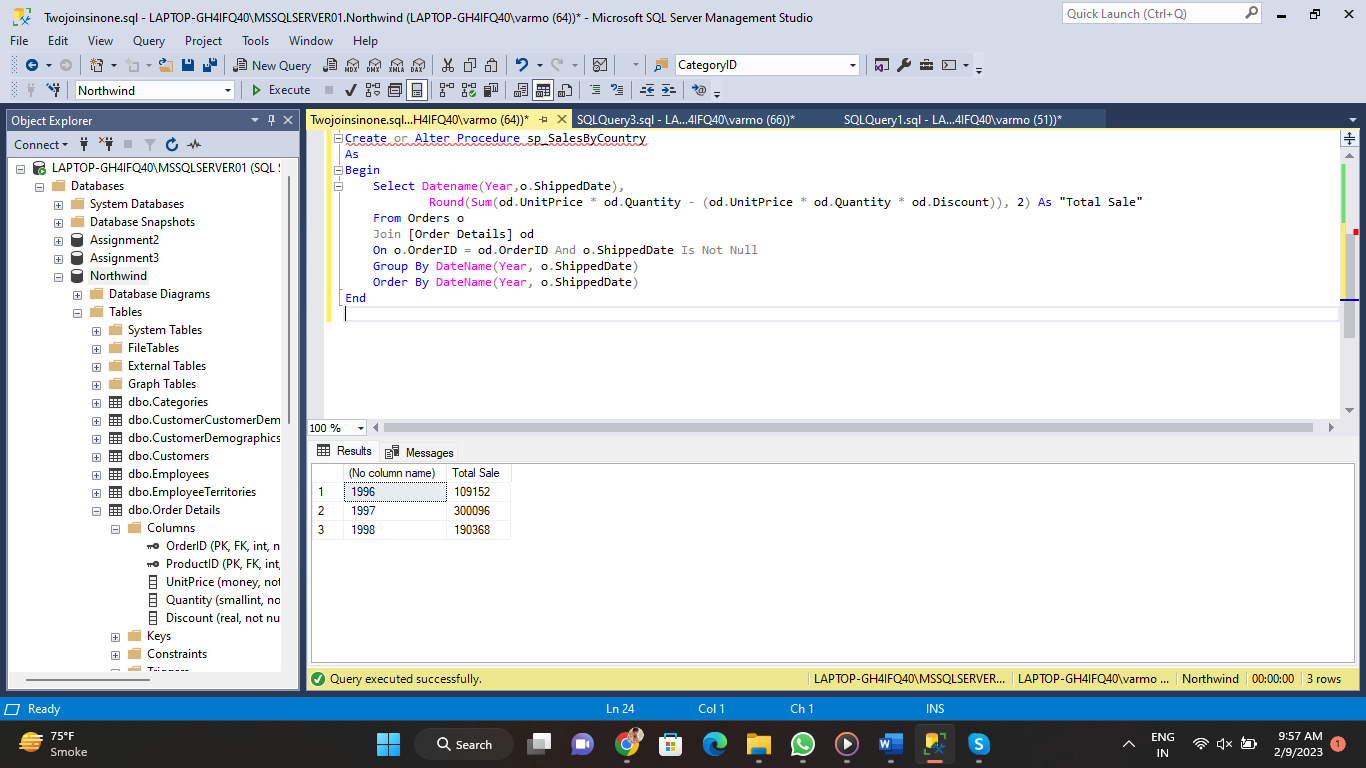
Join [Order Details] od

On o.OrderID = od.OrderID And o.ShippedDate Is Not Null

Group By DateName(Year, o.ShippedDate)

Order By DateName(Year, o.ShippedDate)

End



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

CREATE PROCEDURE usp\_Sales\_By\_Category

AS

BEGIN

SELECT c.CategoryID, c.CategoryName,

ROUND(SUM(od.UnitPrice \* od.Quantity - (od.UnitPrice \* od.Quantity \* od.Discount)), 2) AS 'Total Sale'

FROM Categories c

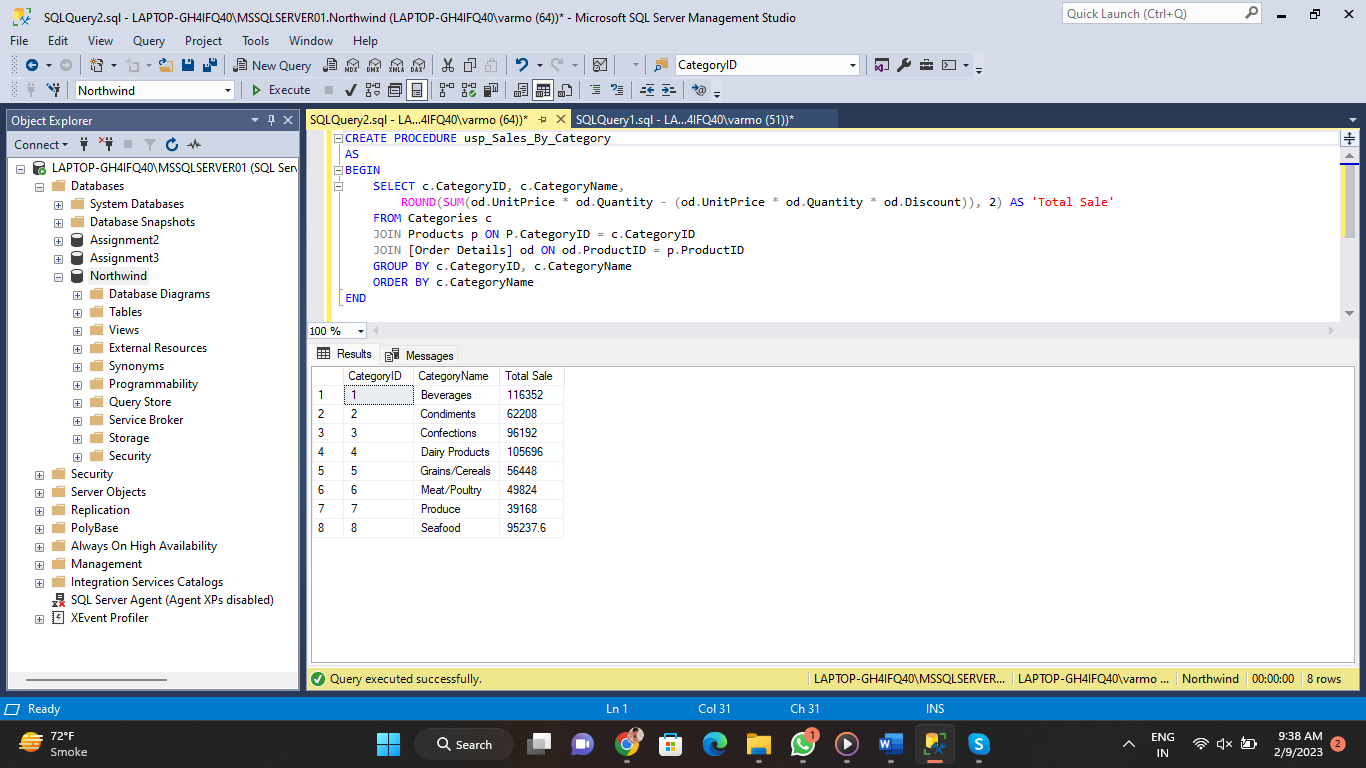
JOIN Products p ON P.CategoryID = c.CategoryID

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

GROUP BY c.CategoryID, c.CategoryName

ORDER BY c.CategoryName

END



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

Create or Alter Proc sp\_TopTenProduct

As

Begin

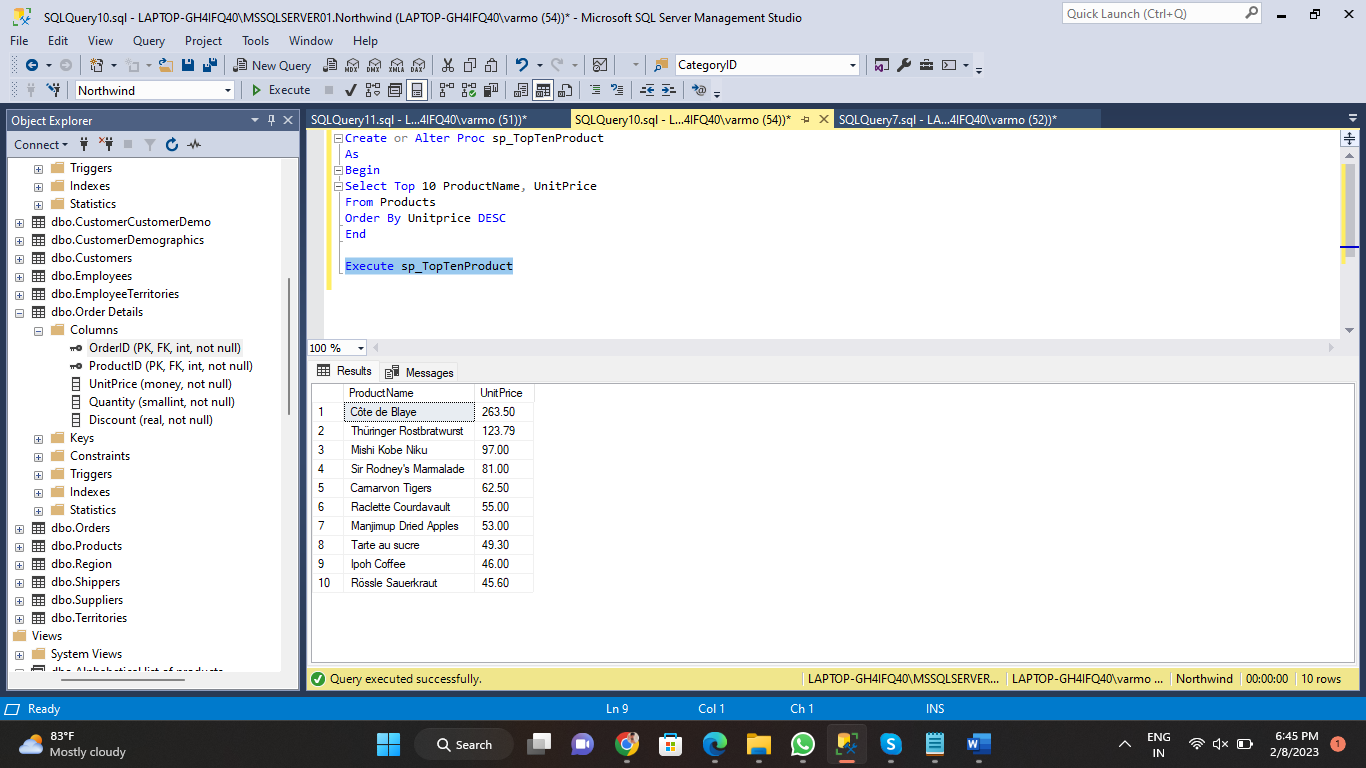
Select Top 10 ProductName, UnitPrice

From Products

Order By Unitprice DESC

End

Execute sp\_TopTenProduct



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

Create or Alter Proc sp\_OrderDetails\_insert

@Order\_ID Int,

@Product\_ID Int,

@Unit\_Price Float,

@Quntity Int,

@Discount Int

As

Begin

If((Select Count(@Order\_ID) From Orders) <> 1)

Begin

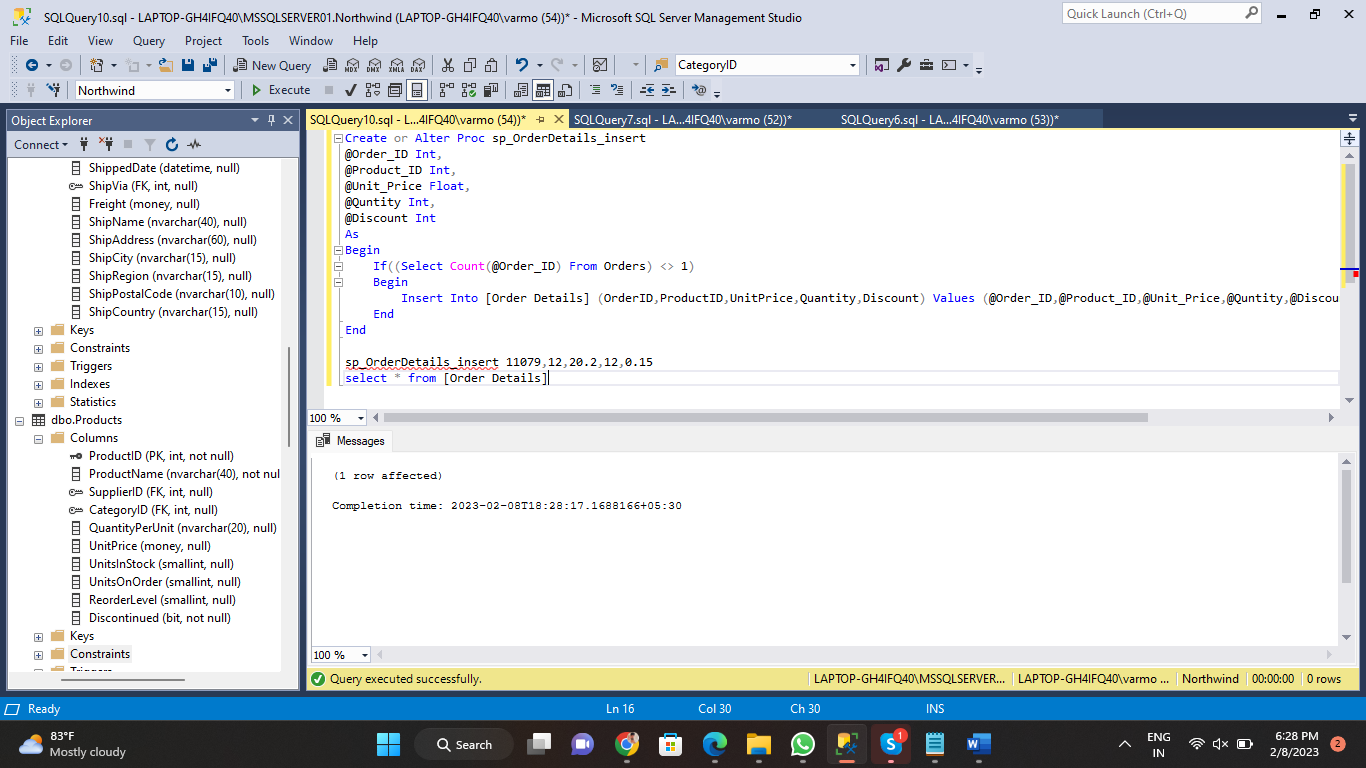
Insert Into [Order Details] (OrderID,ProductID,UnitPrice,Quantity,Discount) Values (@Order\_ID,@Product\_ID,@Unit\_Price,@Quntity,@Discount)

End

End

sp\_OrderDetails\_insert 11079,12,20.2,12,0.15

select \* from [Order Details]



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

Create or Alter Proc sp\_OrderDetails\_update

@Order\_ID Int,

@Product\_ID Int,

@Unit\_Price Float,

@Quntity Int,

@Discount Int

As

Begin

If((Select Count(@Order\_ID) From Orders) <> 0)

Begin

Update [Order Details] Set UnitPrice = @Unit\_Price, Quantity = @Quntity, Discount = @Discount where OrderID = @Order\_ID

End

End

sp\_OrderDetails\_update 11079, 13, 15.2, 13, 0.15

