**Assignment : 04**

**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.**

**First , i have created sp\_avg\_freight procedure to calculate average freight of specified customer.**

**alter PROCEDURE sp\_avg\_freight (@customerID INT, @avgval float output)**

**AS**

**BEGIN**

**-- Calculate the average Freight for the specified customer**

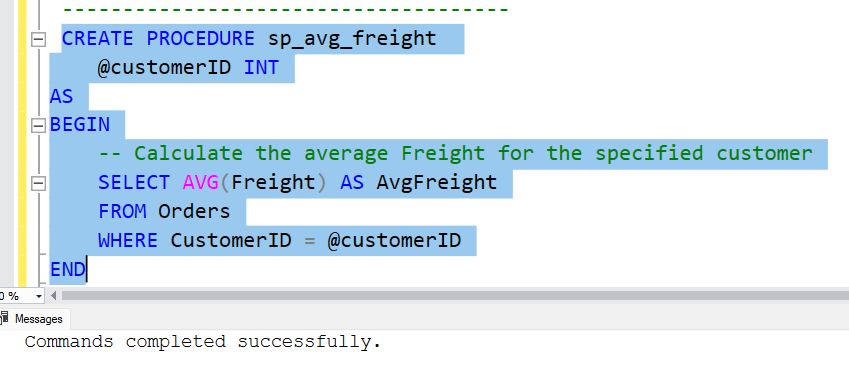
**select @avgval=AVG(Freight)**

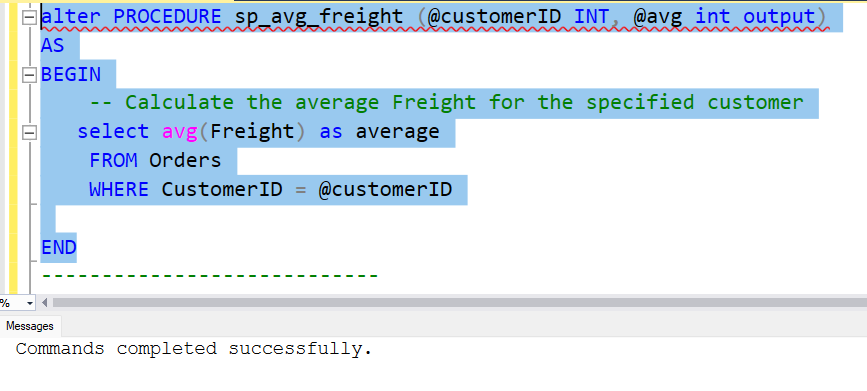
**FROM Orders**

**WHERE CustomerID = @customerID**

**END**

**—-------------------------------------------------------------------------------------**

****

****

**Then I have created trigger which is to implement business rule .**

**ALTER TRIGGER [dbo].[AverageFreightValidation]**

**ON [dbo].[Orders]**

**after INSERT, UPDATE**

**AS**

**BEGIN**

**DECLARE @avgval float,**

**@customerid int**

**select @customerid = CustomerID**

**from inserted**

**EXEC sp\_avg\_freight @customerID = @customerid, @avgval=@avgval OUTPUT**

**IF EXISTS (SELECT \* FROM Orders WHERE Freight > @avgval)**

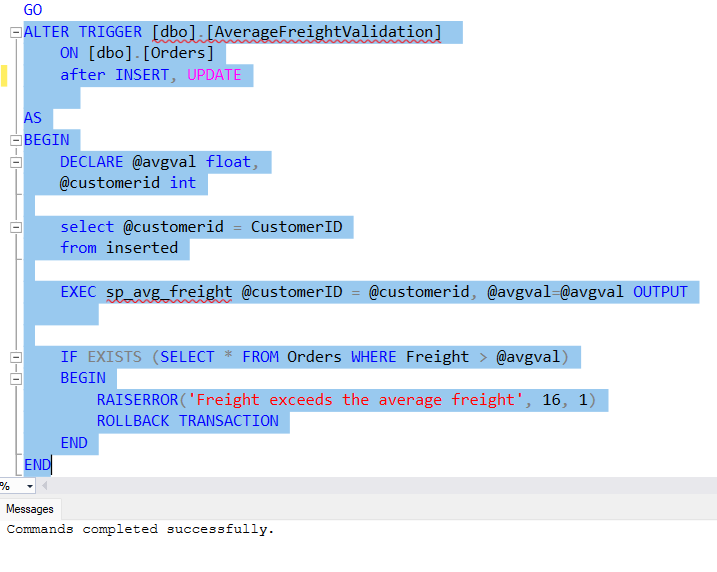
**BEGIN**

**RAISERROR('Freight exceeds the average freight', 16, 1)**

**ROLLBACK TRANSACTION**

**END**

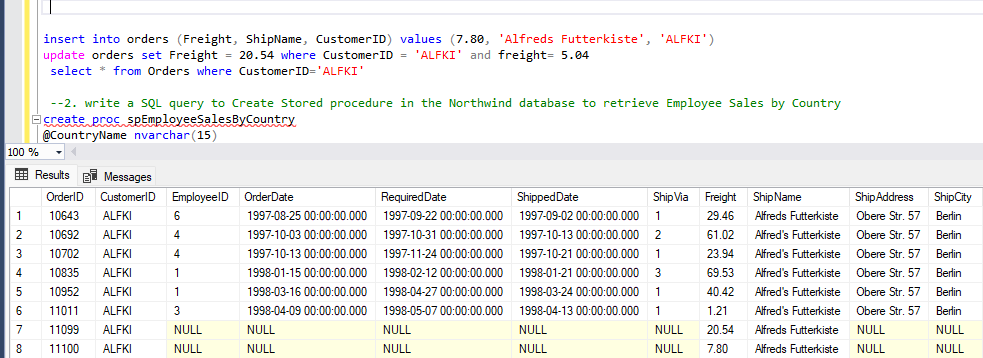
**END**

****

**Then i have tested this business rule to check whether it is working or not.**

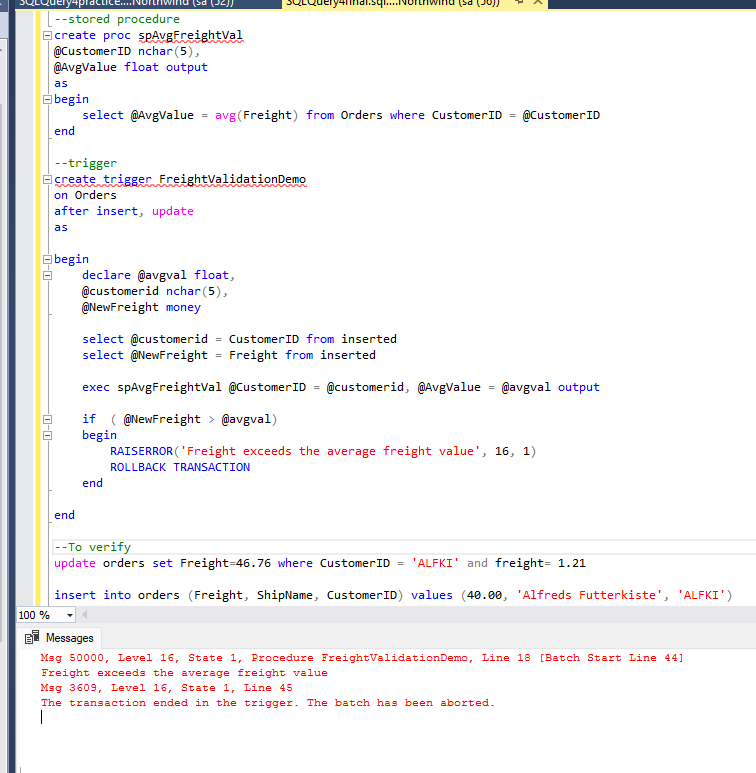
**CASE 1 :**

**Here entered average is not exceeding the limit . So , data is successfully entered.**

****

**CASE 2:**

**Here the inserted freight is exceeding the limit . So trigger is displaying error message and the transaction is rolled back.**

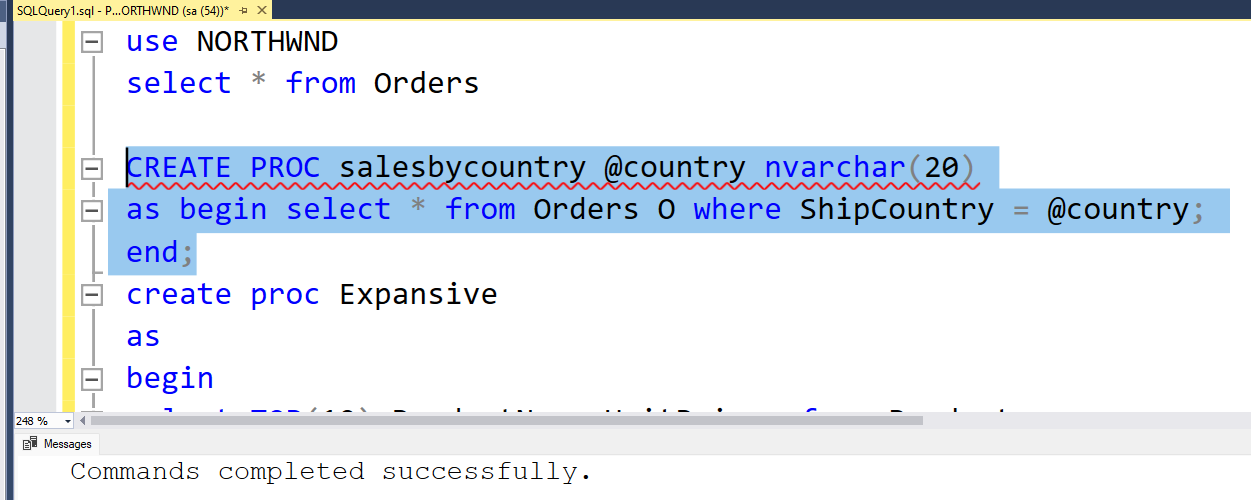
****

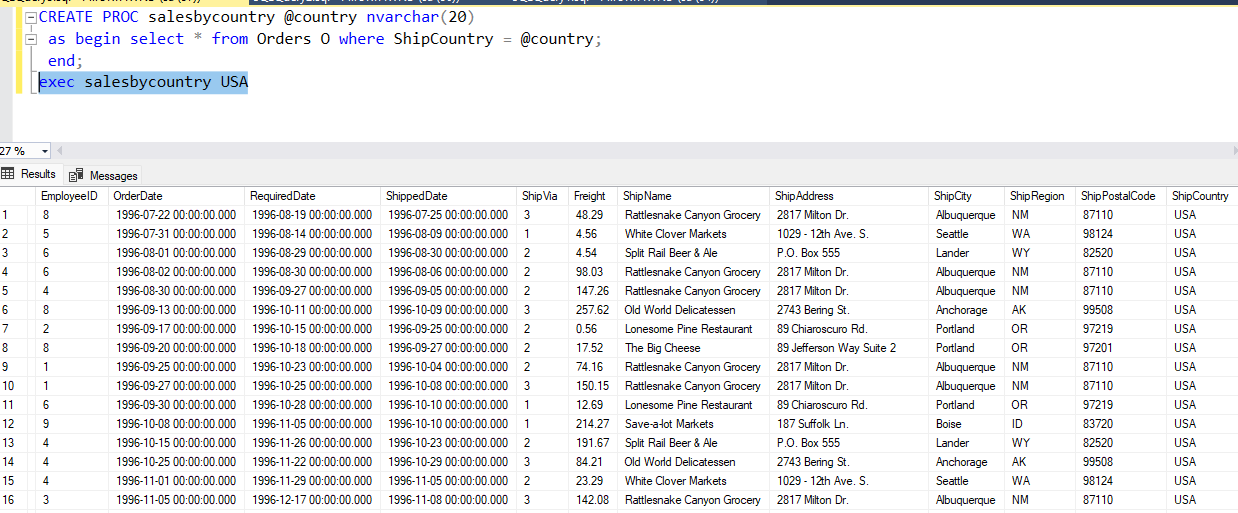
**2. write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country**

**CREATE PROC salesbycountry @country nvarchar(20)**

**as begin select \* from Orders O where ShipCountry = @country;**

**end;**

****

****

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

**CREATE PROC salesviayear @year int**

**as begin**

**select \* from Orders O right join [Order Details]**

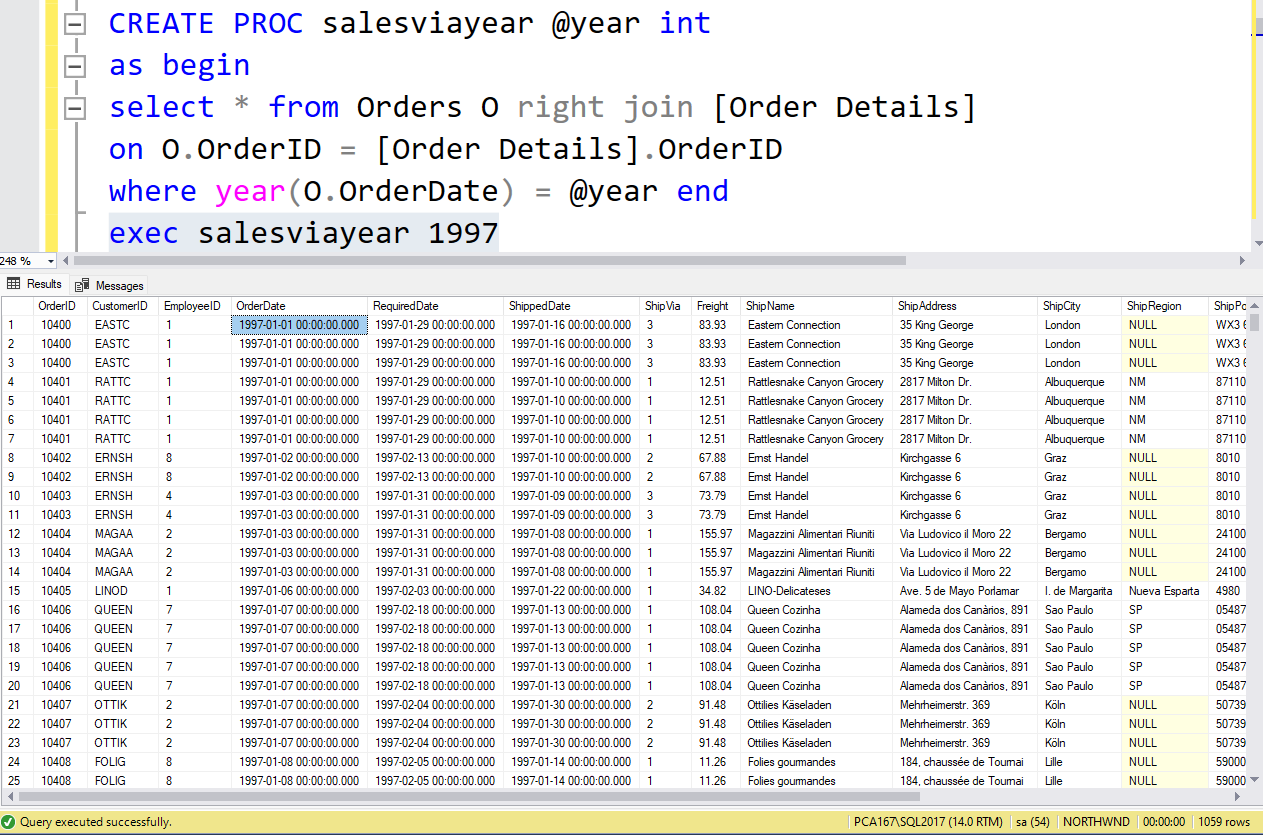
**on O.OrderID = [Order Details].OrderID**

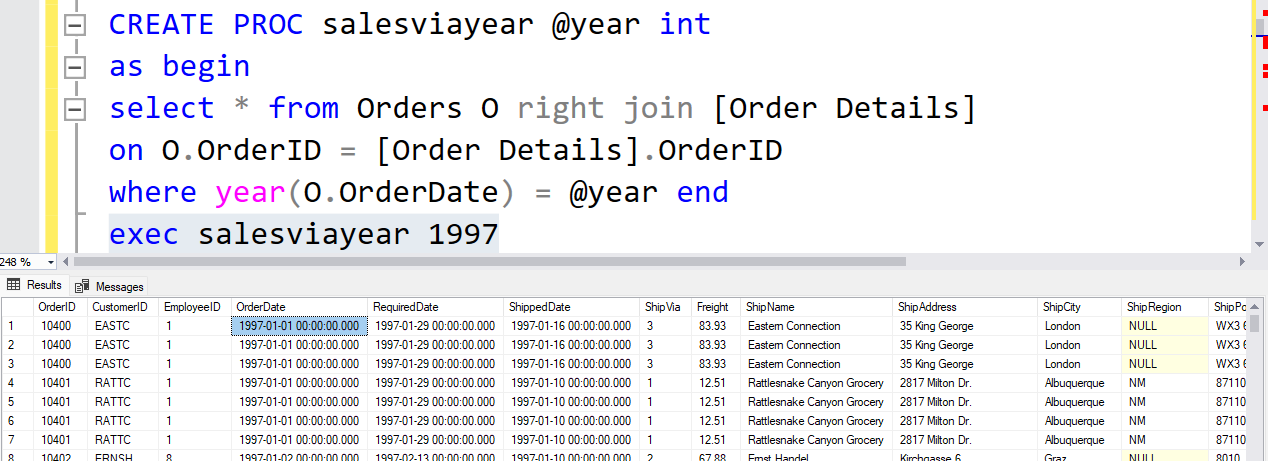
**where year(O.OrderDate) = @year end**

**Then i checked whether procedure is working properly or not.**

**exec salesviayear 1997**

**Here is the result.**

****

****

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

**CREATE PROC saleswithcategory @category nvarchar(20)**

**as begin select \* from Products P**

**left join Suppliers S on S.SupplierID=P.SupplierID**

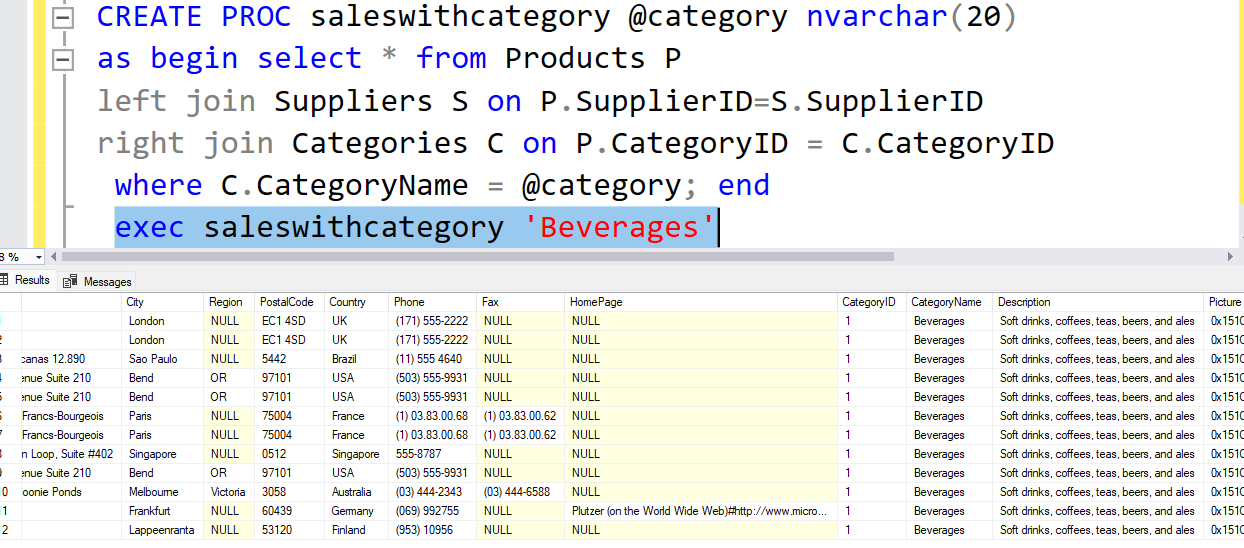
**right join Categories C on C.CategoryID = P.CategoryID**

**where C.CategoryName = @category; end**

**Then i checked whether procedure is working properly or not.**

**exec saleswithcategory 'Beverages'**

**Here is the result.**

****

**I have also created procedure which is printing total sales amount and items taking category name.**

**CREATE PROCEDURE spSalesByCategory**

**@CategoryName VARCHAR(15)**

**AS**

**select \* from Categories**

**BEGIN**

**SELECT C.CategoryName, COUNT(C.CategoryName) AS "Total Sale items" ,**

**SUM(OD.Quantity\*(1-OD.Discount)\*OD.UnitPrice) AS "Total sales amount"**

**FROM [Order Details] OD**

**INNER JOIN Orders O ON OD.OrderID = O.OrderID**

**INNER JOIN Products P ON OD.ProductID = P.ProductID**

**INNER JOIN Categories C ON P.CategoryID= C.CategoryID**

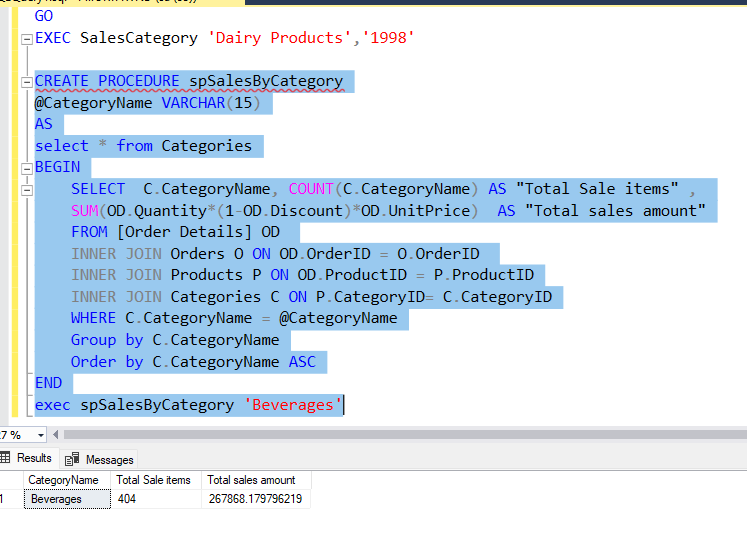
**WHERE C.CategoryName = @CategoryName**

**Group by C.CategoryName**

**Order by C.CategoryName ASC**

**END**

**exec spSalesByCategory 'Beverages'**

****

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

**create proc Expansive**

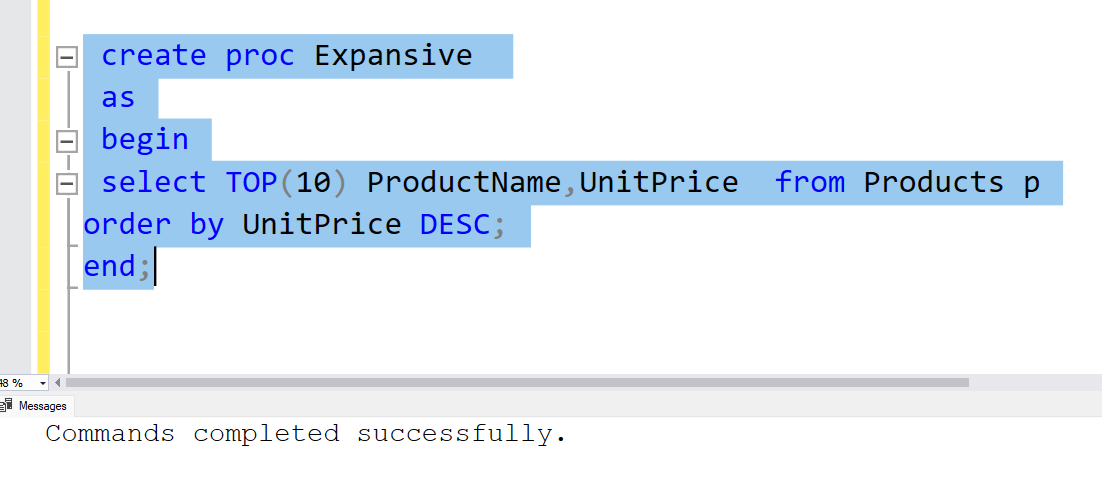
**as**

**begin**

**select TOP(10) ProductName,UnitPrice from Products p**

**order by UnitPrice DESC;**

**end;**

****

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

**create procedure Customerdata @cid nchar(5),**

**@companyname nvarchar(40), @cname varchar(30)=Null,**

**@ctitle nvarchar(30)=Null, @address nvarchar(60)=Null,**

**@city nvarchar(15)=null, @region nvarchar(15)=null,**

**@pcode nvarchar(10)=null, @country nvarchar(15)=null,**

**@phone nvarchar(24)=null, @fax nvarchar(24)=null**

**as**

**begin**

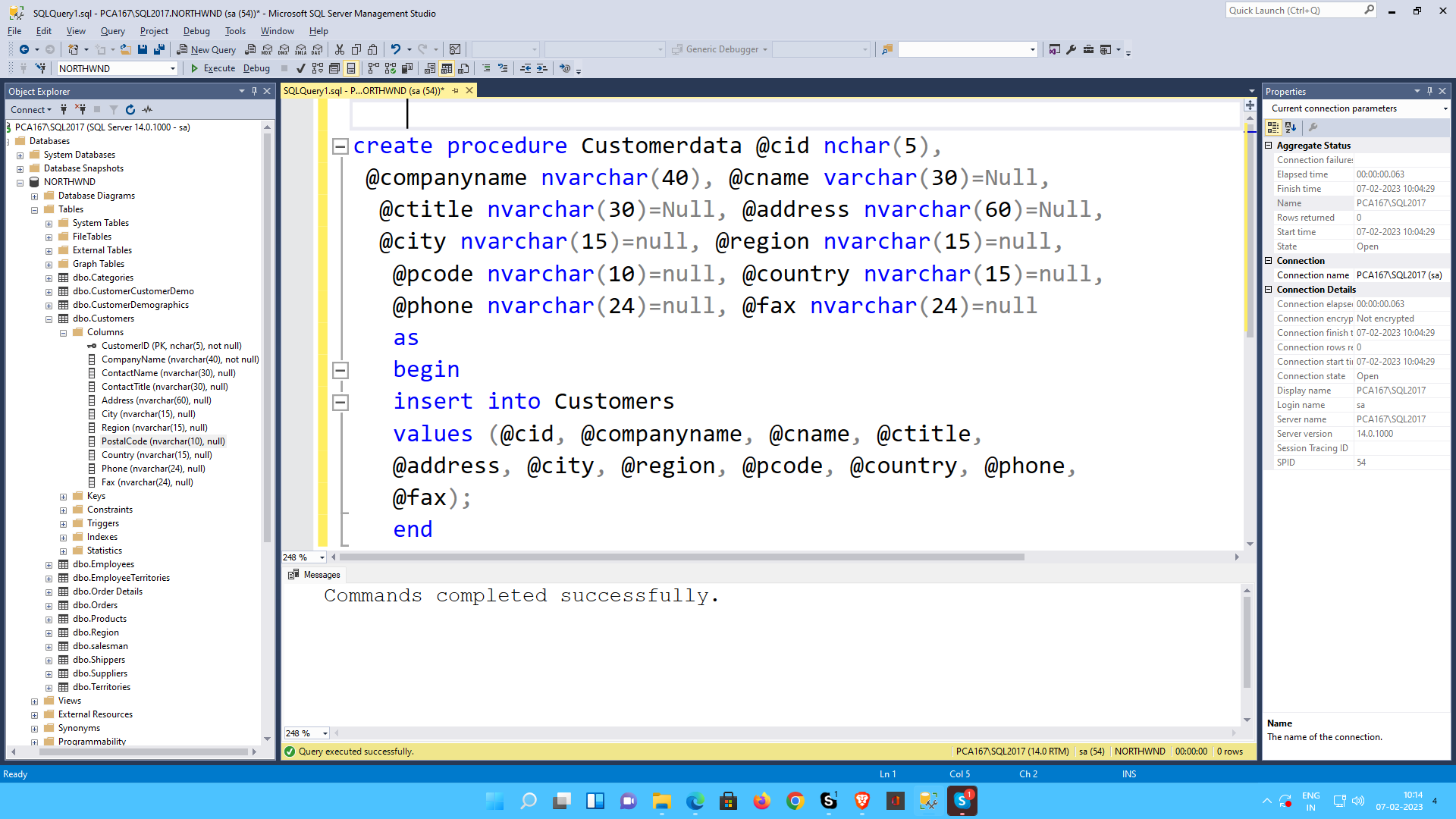
**insert into Customers**

**values (@cid, @companyname, @cname, @ctitle,**

**@address, @city, @region, @pcode, @country, @phone,**

**@fax);**

**end**

****

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

**CREATE PROCEDURE UpdateOrderDetail**

**@OrderID INT,**

**@ProductID INT,**

**@UnitPrice DECIMAL,**

**@Quantity INT,**

**@Discount DECIMAL**

**AS**

**BEGIN**

**UPDATE [Order Details]**

**SET UnitPrice = @UnitPrice,**

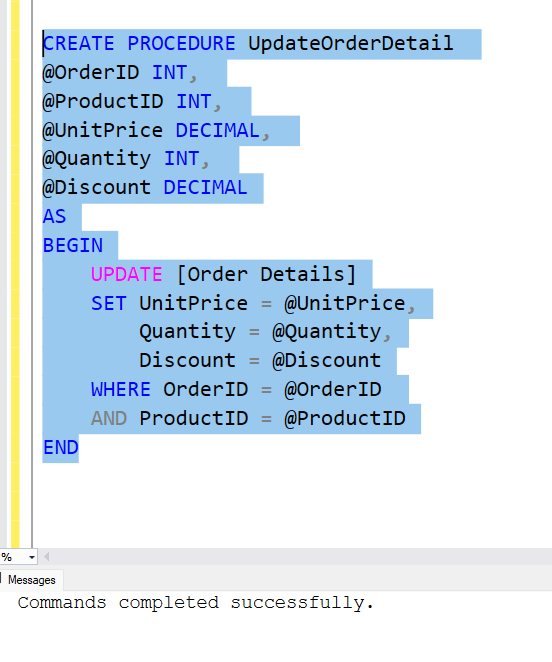
**Quantity = @Quantity,**

**Discount = @Discount**

**WHERE OrderID = @OrderID**

**AND ProductID = @ProductID**

**END**

****