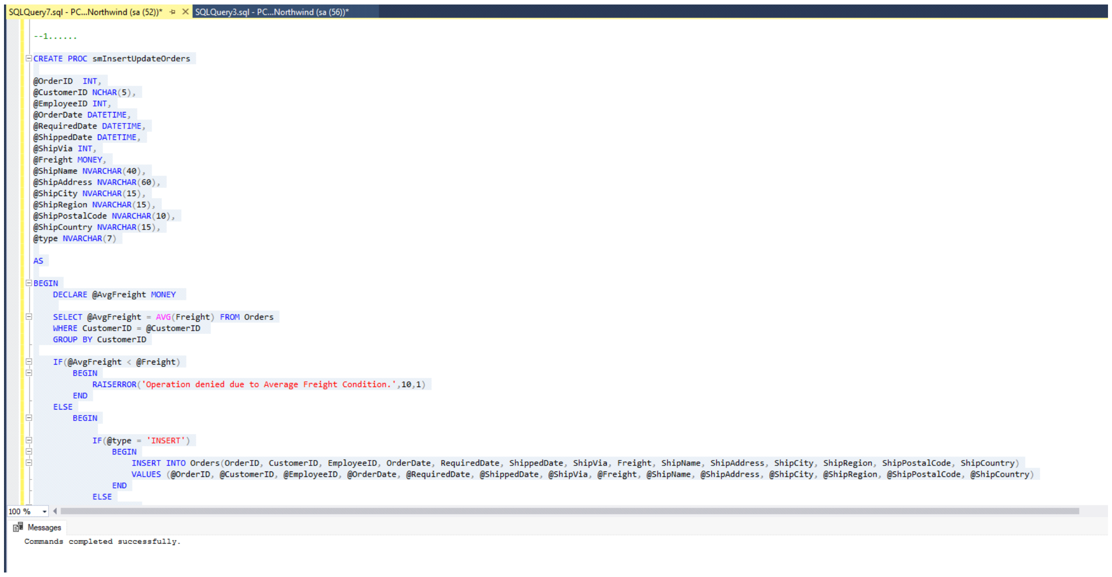
Assignment - 4

Create Stored procedure in Northwind database to insert or update a record in a table

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 PROC smInsertUpdateOrders


@OrderID  INT,

@CustomerID NCHAR(5),

@EmployeeID INT,

@OrderDate DATETIME,

@RequiredDate DATETIME,

@ShippedDate DATETIME,

@ShipVia INT,

@Freight MONEY,

@ShipName NVARCHAR(40),

@ShipAddress NVARCHAR(60),

@ShipCity NVARCHAR(15),
@ShipRegion NVARCHAR(15),
@ShipPostalCode NVARCHAR(10),
@ShipCountry NVARCHAR(15),
@type NVARCHAR(7)

AS

BEGIN
	DECLARE @AvgFreight MONEY

	SELECT @AvgFreight = AVG(Freight) FROM Orders
	WHERE CustomerID = @CustomerID
	GROUP BY CustomerID

	IF(@AvgFreight < @Freight)
		BEGIN
			RAISERROR('Operation denied due to Average Freight
Condition.',10,1)
		END
	ELSE
		BEGIN

			IF(@type = 'INSERT')
				BEGIN
					INSERT INTO Orders(OrderID, CustomerID,
EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia, Freight,
ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode,
ShipCountry)

                           VALUES (@OrderID, @CustomerID,
@EmployeeID, @OrderDate, @RequiredDate, @ShippedDate, @ShipVia,
@Freight, @ShipName, @ShipAddress, @ShipCity, @ShipRegion,
@ShipPostalCode, @ShipCountry)

                      END

                 ELSE

                      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


SET IDENTITY_INSERT Orders ON
```



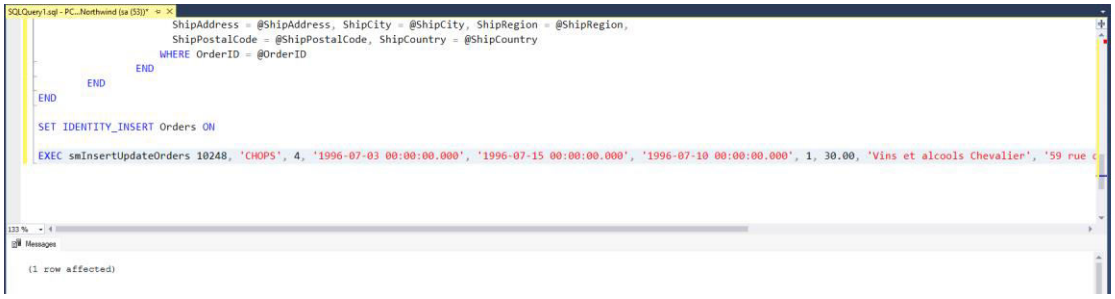
SQLQuery7.sql - PC...Northwind (sa (52))* SQLQuery3.sql - PC...Northwind (sa (56))*

```sql
--1......

CREATE PROC smInsertUpdateOrders

@OrderID  INT,
@CustomerID NCHAR(5),
@EmployeeID INT,
@OrderDate DATETIME,
@RequiredDate DATETIME,
@ShippedDate DATETIME,
@ShipVia INT,
@Freight MONEY,
@ShipName NVARCHAR(40),
@ShipAddress NVARCHAR(60),
@ShipCity NVARCHAR(15),
@ShipRegion NVARCHAR(15),
@ShipPostalCode NVARCHAR(10),
@ShipCountry NVARCHAR(15),
@type NVARCHAR(7)

AS

BEGIN
    DECLARE @AvgFreight MONEY

    SELECT @AvgFreight = AVG(Freight) FROM Orders
    WHERE CustomerID = @CustomerID
    GROUP BY CustomerID

    IF(@AvgFreight < @Freight)
        BEGIN
            RAISERROR('Operation denied due to Average Freight Condition.',10,1)
        END
    ELSE
        BEGIN

            IF(@type = 'INSERT')
                BEGIN
                    INSERT INTO Orders(OrderID, CustomerID, EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry)
                    VALUES (@OrderID, @CustomerID, @EmployeeID, @OrderDate, @RequiredDate, @ShippedDate, @ShipVia, @Freight, @ShipName, @ShipAddress, @ShipCity, @ShipRegion, @ShipPostalCode, @ShipCountry)
                END
            ELSE
```

100 %

Messages

```
Commands completed successfully.
```



SQLQuery1.sql - PC...Northwind (sa (53))*

```sql
                    ShipAddress = @ShipAddress, ShipCity = @ShipCity, ShipRegion = @ShipRegion,
                    ShipPostalCode = @ShipPostalCode, ShipCountry = @ShipCountry
                  WHERE OrderID = @OrderID
                END
        END
END

SET IDENTITY_INSERT Orders ON

EXEC smInsertUpdateOrders 10248, 'CHOPS', 4, '1996-07-03 00:00:00.000', '1996-07-15 00:00:00.000', '1996-07-10 00:00:00.000', 1, 30.00, 'Vins et alcools Chevalier', '59 rue c
```

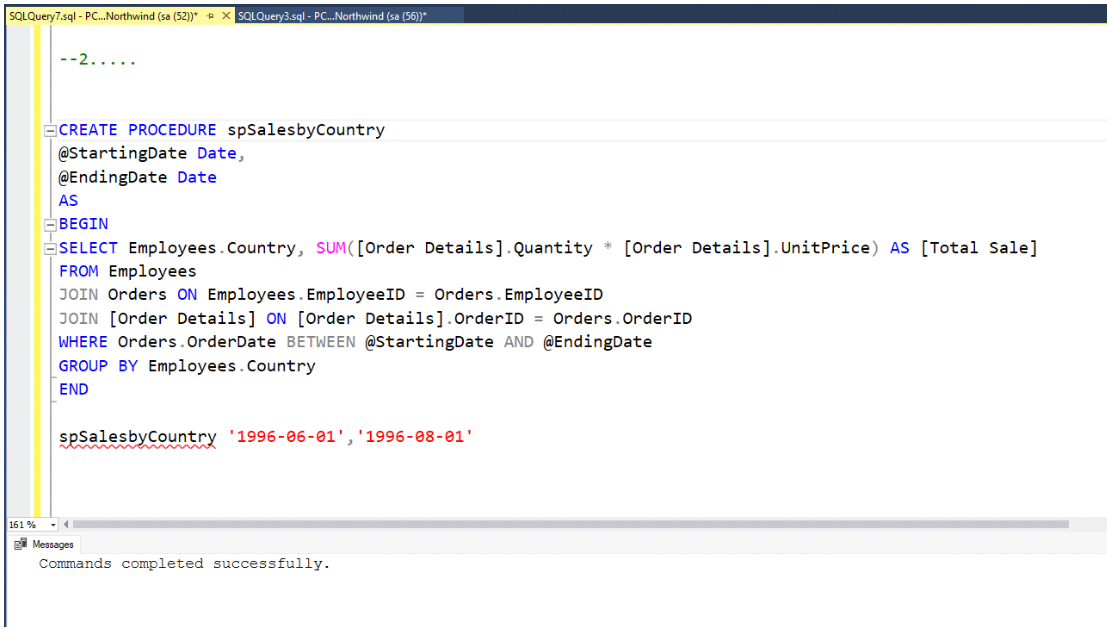
133 %

Messages

```
(1 row affected)
```

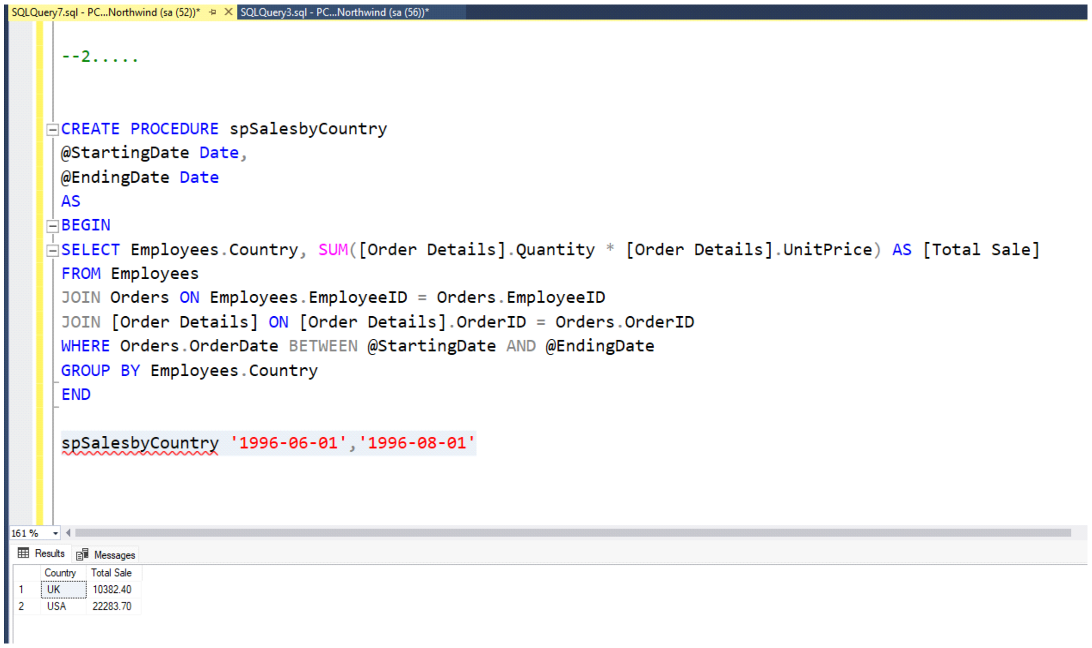
2. write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country

```
CREATE PROCEDURE spSalesbyCountry

@StartingDate Date,

@EndingDate Date

AS

BEGIN

SELECT Employees.Country, SUM([Order Details].Quantity *
[Order Details].UnitPrice) AS [Total Sale]

FROM Employees

JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID

JOIN [Order Details] ON [Order Details].OrderID =
Orders.OrderID

WHERE Orders.OrderDate BETWEEN @StartingDate AND
@EndingDate

GROUP BY Employees.Country

END
```


SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Messages

```sql
--2.....

CREATE PROCEDURE spSalesbyCountry
@StartingDate Date,
@EndingDate Date
AS
BEGIN
SELECT Employees.Country, SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total Sale]
FROM Employees
JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID
JOIN [Order Details] ON [Order Details].OrderID = Orders.OrderID
WHERE Orders.OrderDate BETWEEN @StartingDate AND @EndingDate
GROUP BY Employees.Country
END

spSalesbyCountry '1996-06-01','1996-08-01'
```

```
Commands completed successfully.
```

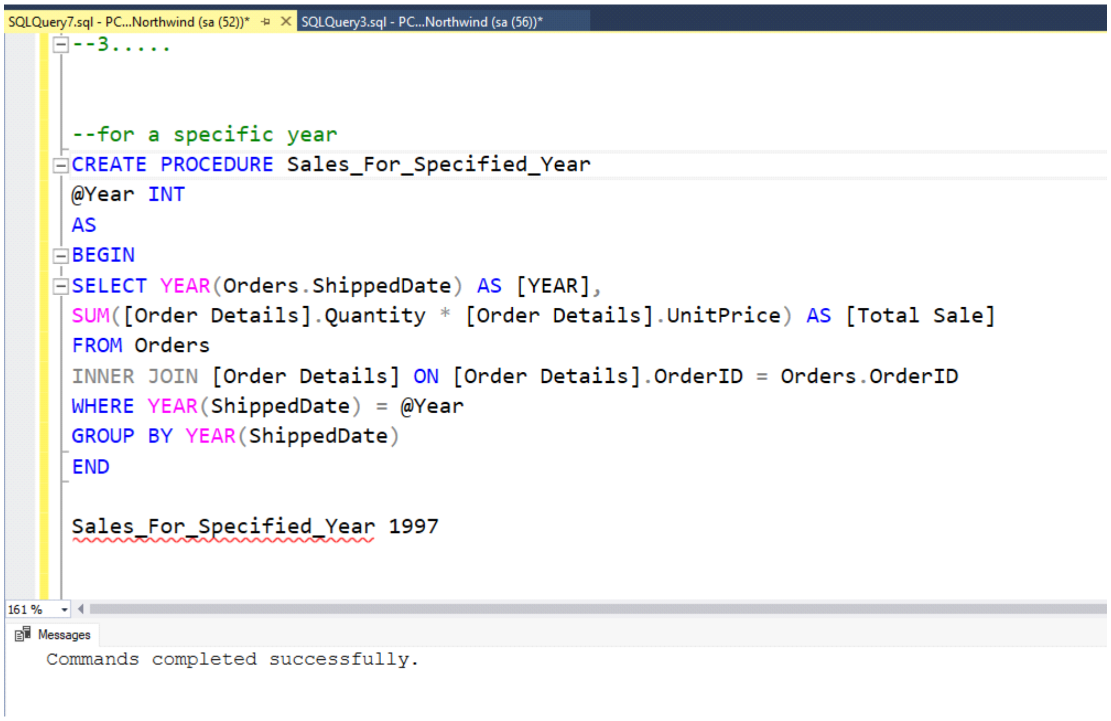

SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Results
Messages

```sql
--2.....

CREATE PROCEDURE spSalesbyCountry
@StartingDate Date,
@EndingDate Date
AS
BEGIN
SELECT Employees.Country, SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total Sale]
FROM Employees
JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID
JOIN [Order Details] ON [Order Details].OrderID = Orders.OrderID
WHERE Orders.OrderDate BETWEEN @StartingDate AND @EndingDate
GROUP BY Employees.Country
END

spSalesbyCountry '1996-06-01','1996-08-01'
```

	Country	Total Sale
1	UK	10382.40
2	USA	22283.70

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

```
--for a specific year

ALTER PROCEDURE Sales_For_Specified_Year

@Year INT

AS

BEGIN

SELECT YEAR(Orders.ShippedDate) AS [YEAR],

SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total
Sale]

FROM Orders

INNER JOIN [Order Details] ON [Order Details].OrderID =
Orders.OrderID

WHERE YEAR(ShippedDate) = @Year

GROUP BY YEAR(ShippedDate)

END


Sales_For_Specified_Year 1997



--for all years

CREATE PROCEDURE Sales_by_Year

AS

BEGIN

SELECT YEAR(Orders.ShippedDate) AS [YEAR],
SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total
Sale]

FROM Orders

INNER JOIN [Order Details] ON [Order Details].OrderID =
Orders.OrderID

GROUP BY YEAR(ShippedDate)

END


Sales_by_Year
```



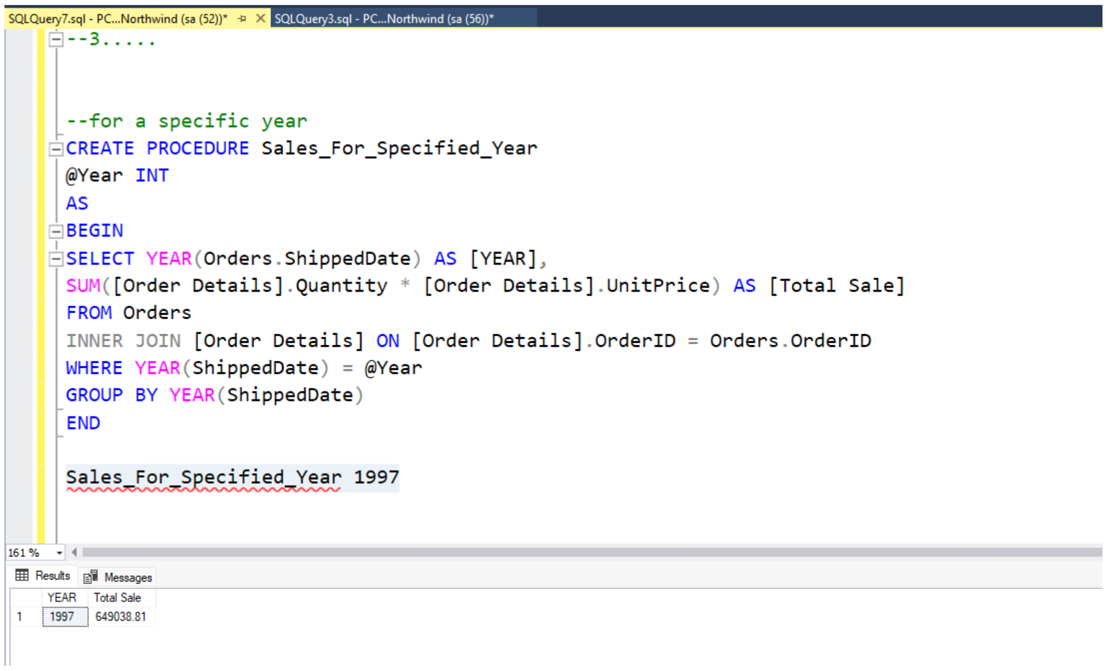
SQLQuery7.sql - PC...Northwind (sa (52))* | SQLQuery3.sql - PC...Northwind (sa (56))*

```
--3.....


--for a specific year
CREATE PROCEDURE Sales_For_Specified_Year
@Year INT
AS
BEGIN
SELECT YEAR(Orders.ShippedDate) AS [YEAR],
SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total Sale]
FROM Orders
INNER JOIN [Order Details] ON [Order Details].OrderID = Orders.OrderID
WHERE YEAR(ShippedDate) = @Year
GROUP BY YEAR(ShippedDate)
END

Sales_For_Specified_Year 1997
```

161 %

Messages

```
Commands completed successfully.
```


SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Results
Messages

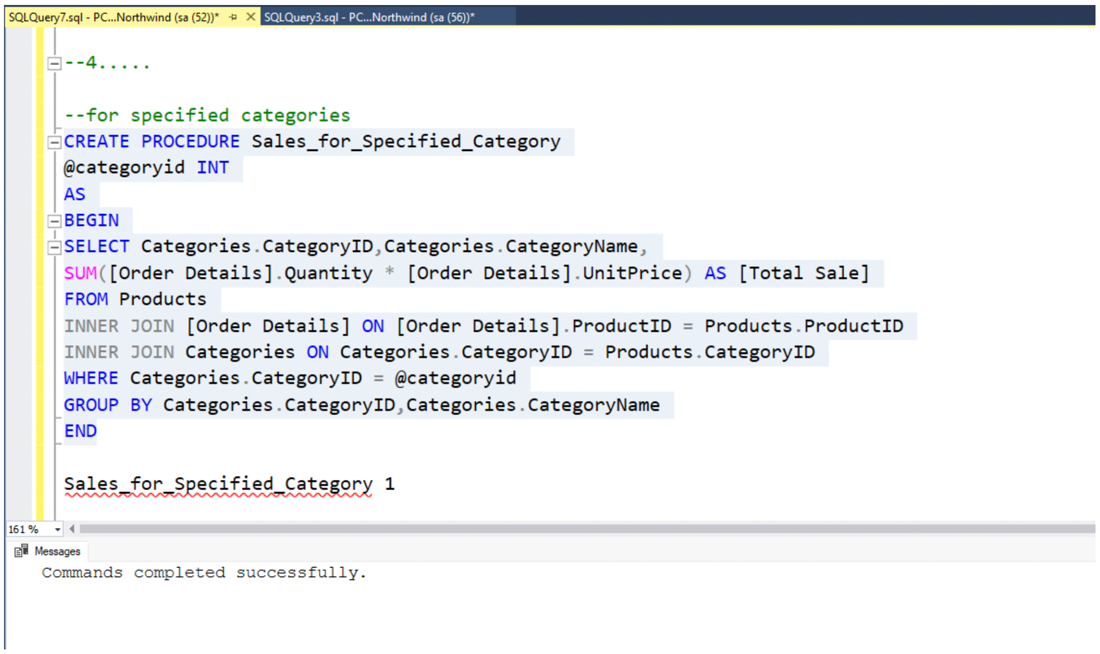
```
--3.....


--for a specific year
CREATE PROCEDURE Sales_For_Specified_Year
@Year INT
AS
BEGIN
SELECT YEAR(Orders.ShippedDate) AS [YEAR],
SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total Sale]
FROM Orders
INNER JOIN [Order Details] ON [Order Details].OrderID = Orders.OrderID
WHERE YEAR(ShippedDate) = @Year
GROUP BY YEAR(ShippedDate)
END

Sales_For_Specified_Year 1997
```

	YEAR	Total Sale
1	1997	649038.81

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

```
--for specified categories

CREATE PROCEDURE Sales_for_Specified_Category

@categoryid INT

AS

BEGIN

SELECT Categories.CategoryID,Categories.CategoryName,

SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total
Sale]

FROM Products

INNER JOIN [Order Details] ON [Order Details].ProductID =
Products.ProductID

INNER JOIN Categories ON Categories.CategoryID = Products.CategoryID

WHERE Categories.CategoryID = @categoryid

GROUP BY Categories.CategoryID,Categories.CategoryName

END


Sales_for_Specified_Category 1




--for all categories

CREATE PROCEDURE Sales_by_Category

AS

BEGIN

SELECT Categories.CategoryID,Categories.CategoryName,
SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total
Sale]

FROM Products

INNER JOIN [Order Details] ON [Order Details].ProductID =
Products.ProductID

INNER JOIN Categories ON Categories.CategoryID = Products.CategoryID

GROUP BY Categories.CategoryID,Categories.CategoryName

END
```

Sales_by_Category



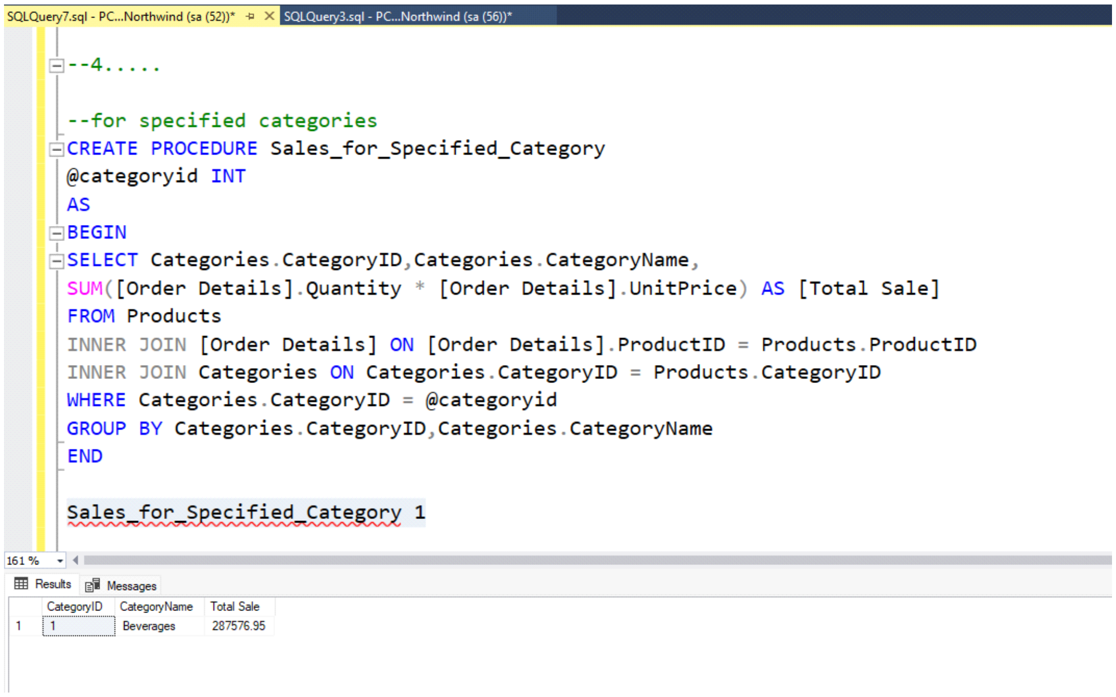
SQLQuery7.sql - PC...Northwind (sa (52))* SQLQuery3.sql - PC...Northwind (sa (56))*

```
--4.....

--for specified categories
CREATE PROCEDURE Sales_for_Specified_Category
@categoryid INT
AS
BEGIN
SELECT Categories.CategoryID,Categories.CategoryName,
SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total Sale]
FROM Products
INNER JOIN [Order Details] ON [Order Details].ProductID = Products.ProductID
INNER JOIN Categories ON Categories.CategoryID = Products.CategoryID
WHERE Categories.CategoryID = @categoryid
GROUP BY Categories.CategoryID,Categories.CategoryName
END

Sales_for_Specified_Category 1
```

161 %

Messages

```
Commands completed successfully.
```

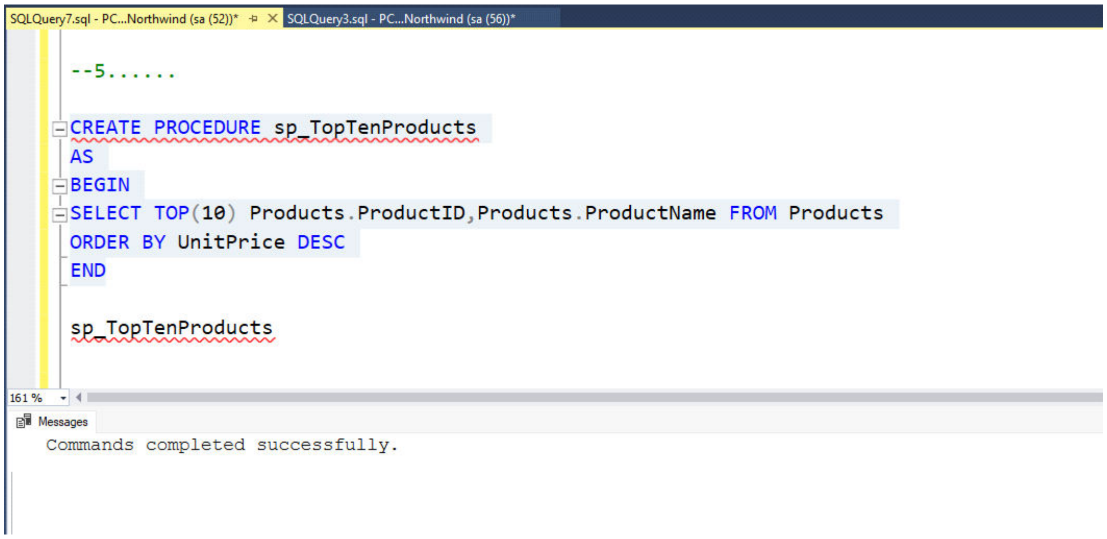

SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Results
Messages

```
--4.....

--for specified categories
CREATE PROCEDURE Sales_for_Specified_Category
@categoryid INT
AS
BEGIN
SELECT Categories.CategoryID,Categories.CategoryName,
SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total Sale]
FROM Products
INNER JOIN [Order Details] ON [Order Details].ProductID = Products.ProductID
INNER JOIN Categories ON Categories.CategoryID = Products.CategoryID
WHERE Categories.CategoryID = @categoryid
GROUP BY Categories.CategoryID,Categories.CategoryName
END

Sales_for_Specified_Category 1
```

	CategoryID	CategoryName	Total Sale
1	1	Beverages	287576.95

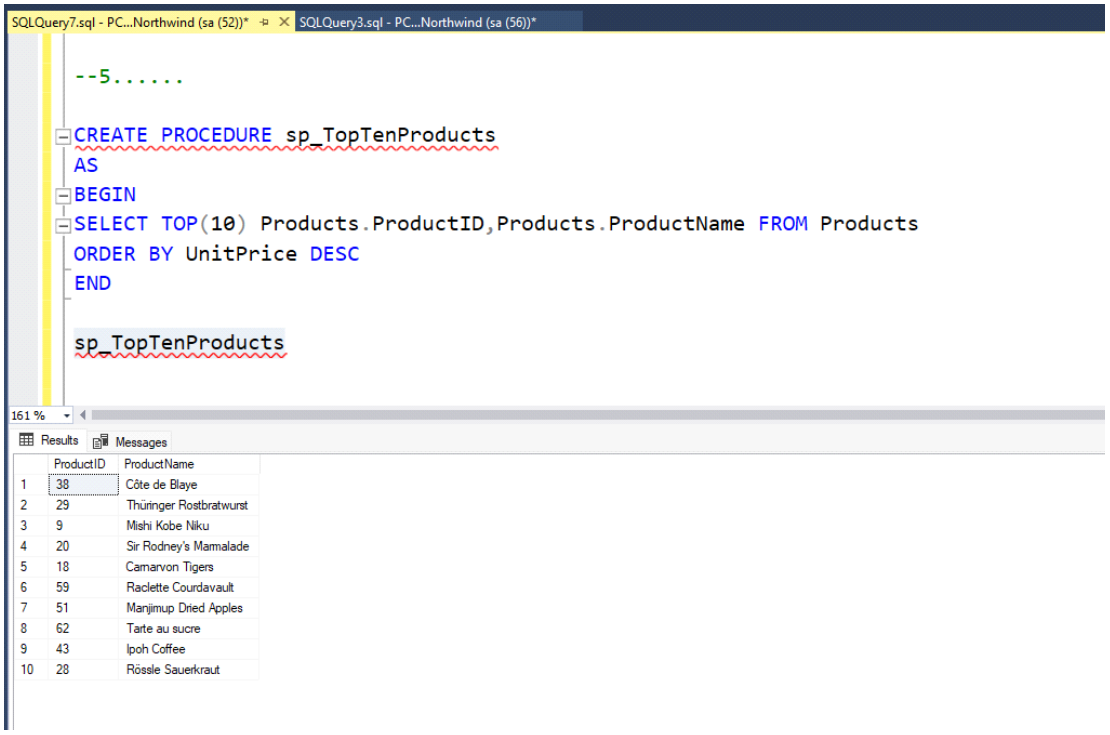
5. write a SQL query to Create Stored procedure in the Northwind database to retrieve Ten Most Expensive Products

```
CREATE PROCEDURE sp_TopTenProducts

AS

BEGIN

SELECT TOP(10) Products.ProductID,Products.ProductName FROM
Products

ORDER BY UnitPrice DESC

END


sp_TopTenProducts
```


SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Messages

```
--5......

CREATE PROCEDURE sp_TopTenProducts
AS
BEGIN
SELECT TOP(10) Products.ProductID,Products.ProductName FROM Products
ORDER BY UnitPrice DESC
END

sp_TopTenProducts
```

```
Commands completed successfully.
```



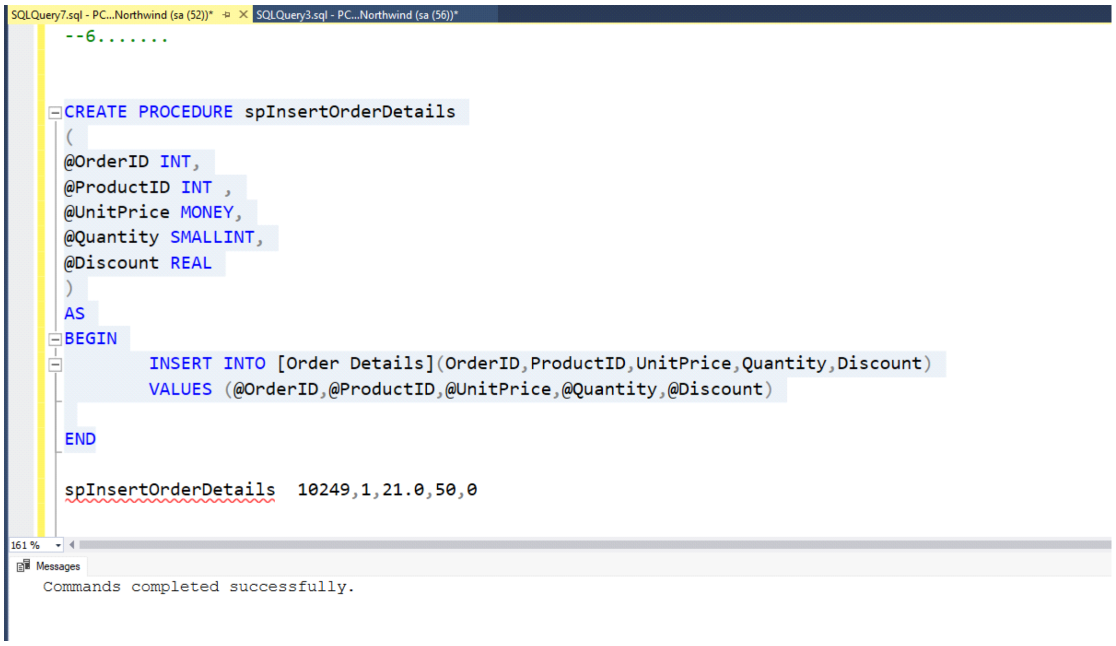
SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Results
Messages

```
--5......

CREATE PROCEDURE sp_TopTenProducts
AS
BEGIN
SELECT TOP(10) Products.ProductID,Products.ProductName FROM Products
ORDER BY UnitPrice DESC
END

sp_TopTenProducts
```

	ProductID	ProductName
1	38	Côte de Blaye
2	29	Thüringer Rostbratwurst
3	9	Mishi Kobe Niku
4	20	Sir Rodney's Marmalade
5	18	Carnarvon Tigers
6	59	Raclette Courdavault
7	51	Manjimup Dried Apples
8	62	Tarte au sucre
9	43	Ipoh Coffee
10	28	Rössle Sauerkraut

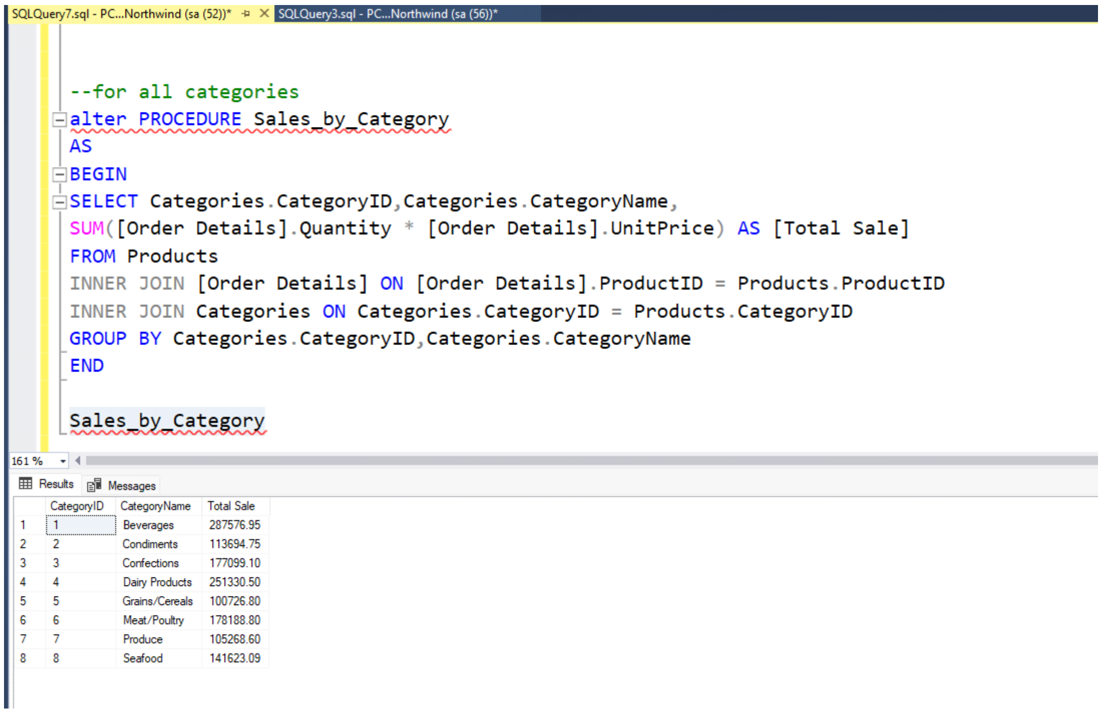
6. write a SQL query to Create Stored procedure in the Northwind database to insert Customer Order Details

```
CREATE PROCEDURE spInsertOrderDetails

(

@OrderID INT,

@ProductID INT ,

@UnitPrice MONEY,

@Quantity SMALLINT,

@Discount REAL

)

AS

BEGIN

           INSERT INTO [Order
Details](OrderID,ProductID,UnitPrice,Quantity,Discount)

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


END
```


SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Messages

```
--6.......


CREATE PROCEDURE spInsertOrderDetails
(
@OrderID INT,
@ProductID INT ,
@UnitPrice MONEY,
@Quantity SMALLINT,
@Discount REAL
)
AS
BEGIN
        INSERT INTO [Order Details](OrderID,ProductID,UnitPrice,Quantity,Discount)
        VALUES (@OrderID,@ProductID,@UnitPrice,@Quantity,@Discount)

END

spInsertOrderDetails  10249,1,21.0,50,0
```

```
Commands completed successfully.
```

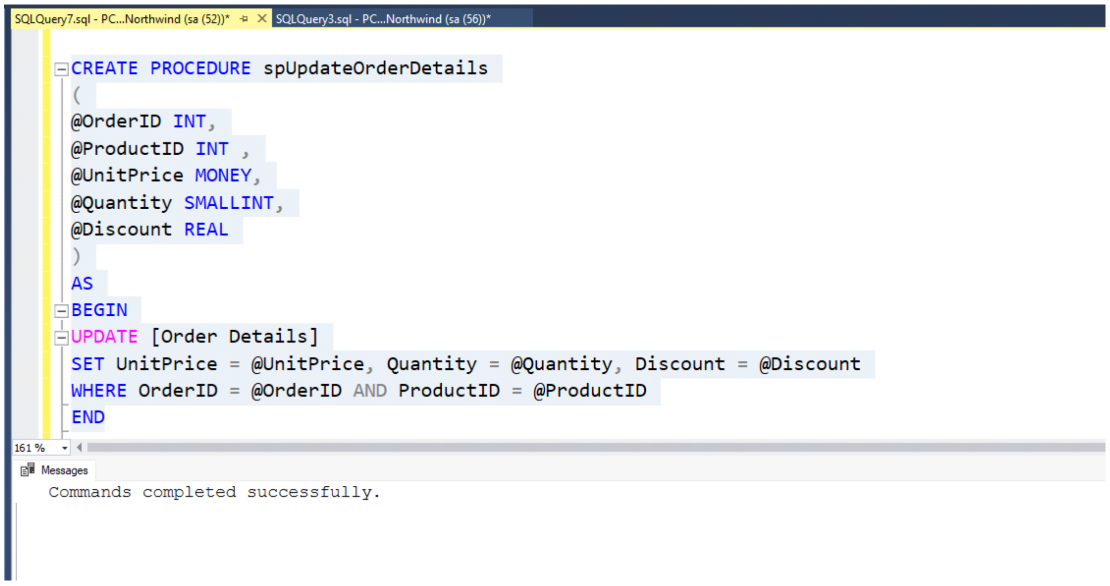

SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Results
Messages

```
--for all categories
alter PROCEDURE Sales_by_Category
AS
BEGIN
SELECT Categories.CategoryID,Categories.CategoryName,
SUM([Order Details].Quantity * [Order Details].UnitPrice) AS [Total Sale]
FROM Products
INNER JOIN [Order Details] ON [Order Details].ProductID = Products.ProductID
INNER JOIN Categories ON Categories.CategoryID = Products.CategoryID
GROUP BY Categories.CategoryID,Categories.CategoryName
END

Sales_by_Category
```

	CategoryID	CategoryName	Total Sale
1	1	Beverages	287576.95
2	2	Condiments	113694.75
3	3	Confections	177099.10
4	4	Dairy Products	251330.50
5	5	Grains/Cereals	100726.80
6	6	Meat/Poultry	178188.80
7	7	Produce	105268.60
8	8	Seafood	141623.09

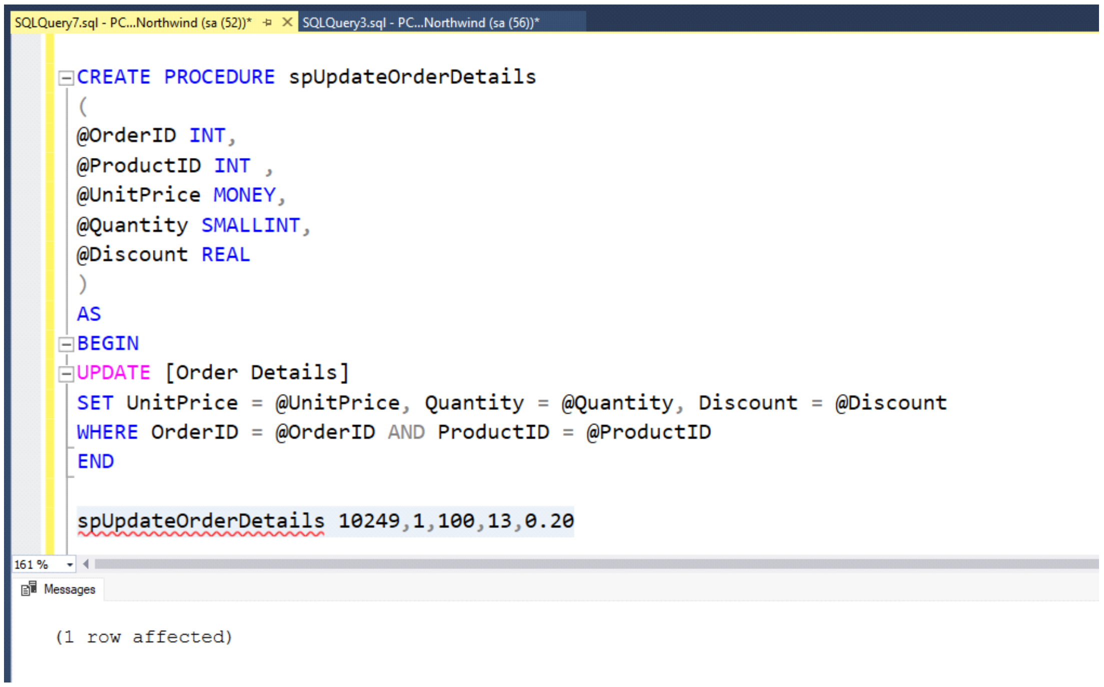
7. write a SQL query to Create Stored procedure in the Northwind database to update Customer Order Details

```
CREATE PROCEDURE spUpdateOrderDetails

(

@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


spUpdateOrderDetails 10249,1,100,13,0.20
```


SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Messages

```
CREATE PROCEDURE spUpdateOrderDetails
(
@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
```

```
Commands completed successfully.
```


SQLQuery7.sql - PC...Northwind (sa (52))*
SQLQuery3.sql - PC...Northwind (sa (56))*
161 %
Messages

```
CREATE PROCEDURE spUpdateOrderDetails
(
@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

spUpdateOrderDetails 10249,1,100,13,0.20
```

```
(1 row affected)
```