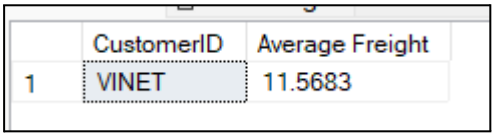
Assignment : 4

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.

Query:

```
create proc sp_avgFreight(@CustId nvarchar(10))
as
begin
	select CustomerID, AVG(Freight) as [Average Freight]
	from orders
	where CustomerID = @CustId
	group by CustomerID
end

exec sp_avgFreight 'VINET'
```



	CustomerID	Average Freight
1	VINET	11.5683

```
create proc sp_insertIntoOrder(@OrderID int, @CustomerID nchar(5),
@EmployeeID int, @OrderDate datetime, @RequiredDate datetime,
@ShippedDate datetime, @ShipVia int, @Freight money, @ShipName
nvarchar(40), @ShipAddress nvarchar(40), @ShipCity nvarchar(40),
@ShipRegion nvarchar(40), @ShipPostalCode nvarchar(40), @ShipCountry
nvarchar(40))
as
begin
	declare @avgFreight money
	set @avgFreight = (select AVG(o.Freight) from Orders as o where
CustomerID = @CustomerID)
	if(@avgFreight>@Freight)
	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
```

```
            raiserror('Inserted Freight is more than the average
freight of Customer', 10, 1)
      end
end

SET IDENTITY_INSERT  Orders ON

exec sp_insertIntoOrder '10225', 'VINET', '5', '', '', '', '1', '11',
'', '', '', '', '', ''

select * from Orders
where CustomerID = 'VINET' and OrderID  = 10225
```

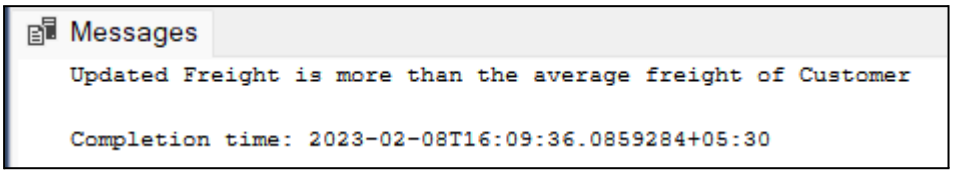
	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName
1	10225	VINET	5	1900-01-01 00:00:00.000	1900-01-01 00:00:00.000	1900-01-01 00:00:00.000	1	11.00	

```
create proc sp_updateIntoOrder(@OrderID int, @CustomerID nchar(5),
@EmployeeID int, @OrderDate datetime, @RequiredDate datetime,
@ShippedDate datetime, @ShipVia int, @Freight money, @ShipName
nvarchar(40), @ShipAddress nvarchar(40), @ShipCity nvarchar(40),
@ShipRegion nvarchar(40), @ShipPostalCode nvarchar(40), @ShipCountry
nvarchar(40))
as
begin
      declare @avgFreight money
      set @avgFreight = (select AVG(o.Freight) from Orders as o where
CustomerID = @CustomerID)
      if(@avgFreight>@Freight)
      begin
            update Orders
            set 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 and CustomerID =
@CustomerID
            end
      else
      begin
            raiserror('Updated Freight is more than the average
freight of Customer', 10, 1)
```

```
      end
end

exec sp_updateIntoOrder '10225', 'VINET', '5', '', '', '', '1', '13',
'', '', '', '', '', ''


select * from Orders
where CustomerID = 'VINET' and OrderID  = 10225
```



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

Query:

```
create procedure sp_empSalesByCountry (@StartDate DATE, @EndDate
DATE)
as
begin
    select
        e.Country, SUM(od.UnitPrice * od.Quantity) as TotalSales
    from
        Employees e
        inner join Orders o ON e.EmployeeID = o.EmployeeID
        inner join [Order Details] od ON o.OrderID = od.OrderID
    where
        o.OrderDate between @StartDate AND @EndDate
    GROUP BY
        e.Country
END

exec sp_empSalesByCountry '1996/07/04' , '1999/07/15'
```

	Country	TotalSales
1	USA	976752.75
2	UK	378025.84

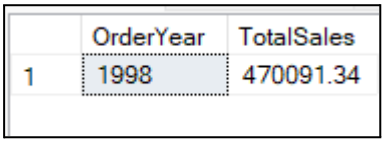
3. write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales by Year

Query:

```
create procedure sp_salesByYear (@Year INT)
```

```
as
begin
    select
        YEAR(OrderDate) as OrderYear, SUM(od.UnitPrice * od.Quantity)
as TotalSales
    from
        Orders o
        inner join [Order Details] od ON o.OrderID = od.OrderID
    where
        YEAR(OrderDate) = @Year
      group by YEAR(OrderDate)
END

exec sp_salesByYear 1998
```



	OrderYear	TotalSales
1	1998	470091.34

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

Query:

```
create procedure sp_salesByCategory (@StartDate DATE, @EndDate DATE)
as
begin
    select
        c.CategoryName, SUM(od.UnitPrice * od.Quantity) as TotalSales
    from
        Categories c
        inner join Products p ON c.CategoryID = p.CategoryID
        inner join [Order Details] od ON p.ProductID = od.ProductID
        inner join Orders o ON od.OrderID = o.OrderID
    where
        o.OrderDate between @StartDate AND @EndDate
    GROUP BY
        c.CategoryName
END

exec sp_salesByCategory '1996/07/04' , '1999/07/15'
```

	CategoryName	TotalSales
1	Seafood	141623.09
2	Meat/Poultry	178188.80
3	Condiments	114014.75
4	Confections	177099.10
5	Produce	105268.60
6	Dairy Products	251330.50
7	Beverages	286526.95
8	Grains/Cereals	100726.80

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

Query:

```
create procedure sp_MostExpensive
as
begin
    select
        TOP 10 p.ProductName, p.UnitPrice
    from
        Products p
    ORDER BY
        p.UnitPrice DESC
END

exec sp_MostExpensive
```

	ProductName	UnitPrice
1	Côte de Blaye	263.50
2	Thüringer Rostbratwurst	123.79
3	Mishi Kobe Niku	97.00
4	Sir Rodney's Marmalade	81.00
5	Carnarvon Tigers	62.50
6	Raclette Courdavault	55.00
7	Manjimup Dried Apples	53.00
8	Tarte au sucre	49.30
9	Ipoh Coffee	46.00
10	Rössle Sauerkraut	45.60

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

Query:

```
create procedure sp_insertOrdrDetail (@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

exec sp_insertOrdrDetail '10248', '5', '25.00', '12', '0.2'

select * from [Order Details]
where OrderID = 11077 and ProductID = 5 and UnitPrice = 25.00
```

	OrderID	ProductID	UnitPrice	Quantity	Discount
1	11077	5	25.00	12	0.2

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

Query:

```
create procedure sp_updateOrdrDetail (@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 sp_updateOrdrDetail '11077', '3', '20.00', '3', '0.12'

select * from [Order Details]
where OrderID = 11077 and ProductID = 3 and UnitPrice = 20.00
```

	OrderID	ProductID	UnitPrice	Quantity	Discount
1	11077	3	20.00	3	0.12