**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 For Check Validate Freight

CREATE PROCEDURE sp\_Validate\_Freight

-- inputted customer

@CustomerID nvarchar(10),

-- returned average freight

@AverageFreight money output

AS

BEGIN

SELECT @AverageFreight = AVG(Freight)

FROM Orders

WHERE CustomerID = @CustomerID

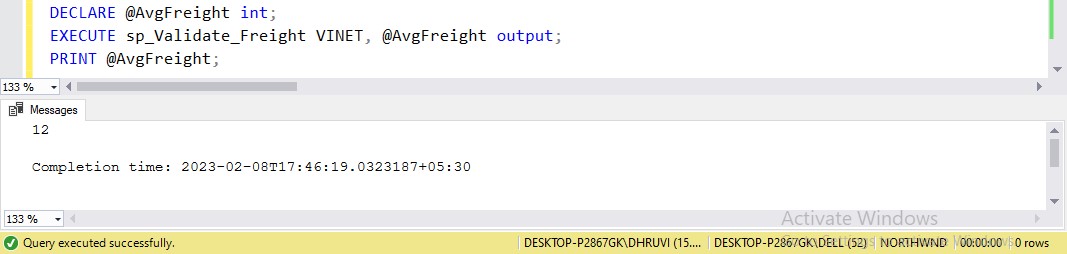
END



DECLARE @AvgFreight int;

EXECUTE sp\_Validate\_Freight 'VINET', @AvgFreight output;

PRINT @AvgFreight;



-- Procedure For Insert Details

CREATE PROCEDURE sp\_Validate\_Freight\_Insert

@OrderID int, @CustomerID nchar(15),

@EmployeeID int, @OrderDate datetime,

@RequiredDate datetime, @ShippedDate datetime,

@ShipVia int, @Freight money, @ShipName nvarchar(50),

@ShipAddress nvarchar(60), @ShipCity nvarchar(20),

@ShipRegion nvarchar(20), @ShipPostalCode nvarchar(20),

@ShipCountry nvarchar(50)

AS

BEGIN

DECLARE @avgfraight money

SET @avgfraight = (

SELECT AVG(Freight) FROM Orders

WHERE CustomerID=@CustomerID )

IF(@avgfraight > @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

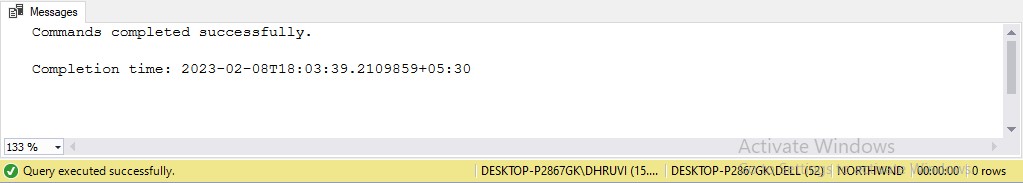
ELSE

BEGIN

RAISERROR ('Inserted Freight is more than customer avg freight',10,1)

END

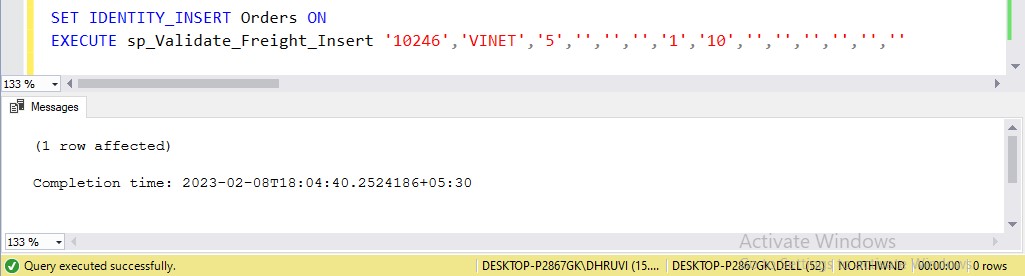
END



-- Insert Data

SET IDENTITY\_INSERT Orders ON

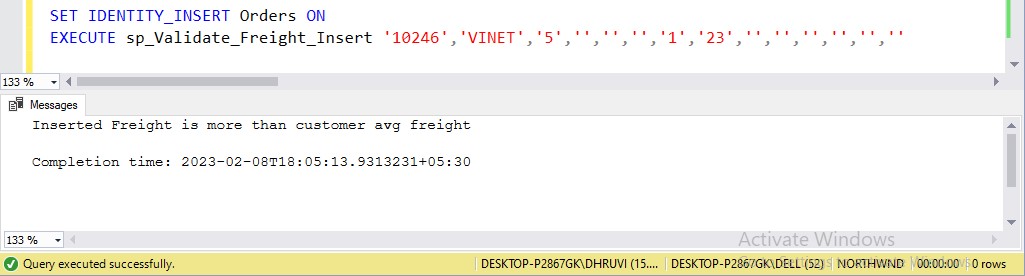
EXECUTE sp\_Validate\_Freight\_Insert '10246','VINET','5','','','','1','10','','','','','',''



-- Insert Data

SET IDENTITY\_INSERT Orders ON

EXECUTE sp\_Validate\_Freight\_Insert '10246','VINET','5','','','','1','23','','','','','',''



-- Procedure For Update Details

CREATE PROCEDURE sp\_Validate\_Freight\_Update(

@OrderID int, @CustomerID nchar(15),

@EmployeeID int, @OrderDate datetime,

@RequiredDate datetime, @ShippedDate datetime,

@ShipVia int, @Freight money, @ShipName nvarchar(50),

@ShipAddress nvarchar(60), @ShipCity nvarchar(20),

@ShipRegion nvarchar(20), @ShipPostalCode nvarchar(20),

@ShipCountry nvarchar(50))

AS

BEGIN

DECLARE @avgfraight money

SET @avgfraight = (

SELECT AVG(Freight) FROM Orders

WHERE CustomerID=@CustomerID)

IF(@avgfraight > @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

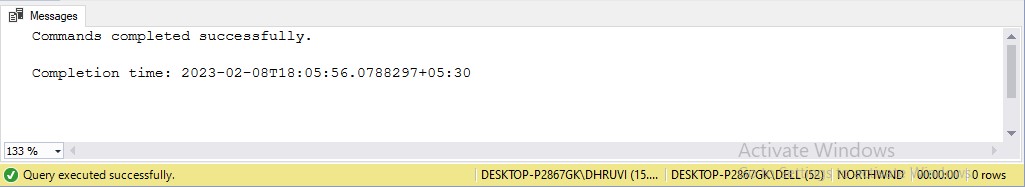
ELSE

BEGIN

RAISERROR('Updated Freight is more than avg freight',10,1)

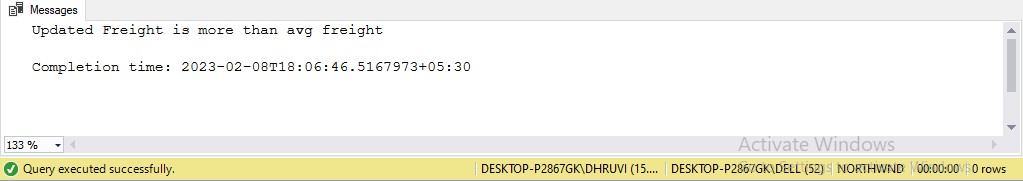
END

END



-- Update Data

EXECUTE sp\_Validate\_Freight\_Update '10246','VINET','5','','','','2','23','','','','','',''



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

CREATE PROCEDURE Employee\_Sales\_By\_Country

@Beginning\_Date DateTime, @Ending\_Date DateTime

AS

BEGIN

SELECT E.Country, E.FirstName,

SUM(CONVERT(money,(OD.UnitPrice \* Quantity \* (1 - Discount)/100))\*100) AS SaleAmount

FROM Employees E

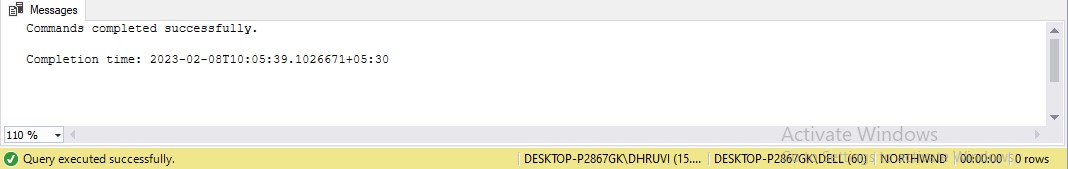
JOIN Orders O ON E.EmployeeID = O.EmployeeID

JOIN [Order Details] OD ON O.OrderID = OD.OrderID

GROUP BY E.Country,E.FirstName, O.ShippedDate

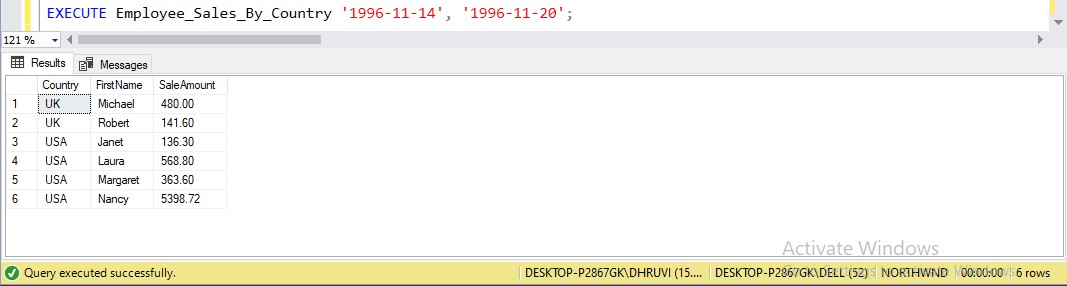
HAVING O.ShippedDate BETWEEN @Beginning\_Date AND @Ending\_Date;

END



-- Execute Store Procedure Employee\_Sales\_By\_Country

EXECUTE Employee\_Sales\_By\_Country '1996-11-14', '1996-11-20';



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

CREATE PROCEDURE Sales\_By\_Year

AS

BEGIN

SELECT YEAR(O.ShippedDate) AS Year,

SUM(CONVERT(money,(OD.UnitPrice \* Quantity \* (1 - Discount)/100))\*100) AS SaleAmount

FROM Orders O

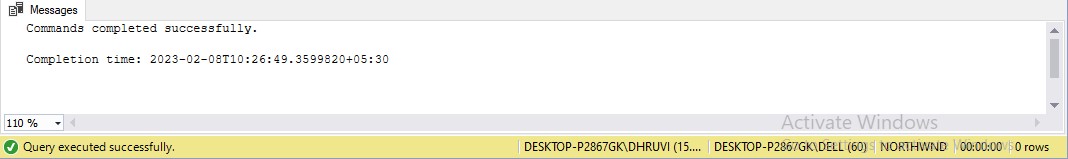
JOIN [Order Details] OD

ON O.OrderID = OD.OrderID

GROUP BY YEAR(O.ShippedDate)

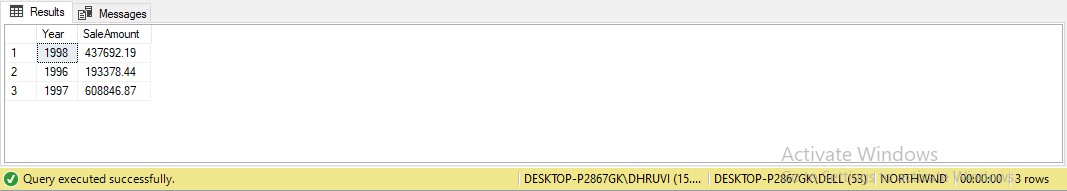
HAVING YEAR(O.ShippedDate) IS NOT NULL;

END



-- Execute Store Procedure Sales\_By\_year

EXECUTE Sales\_By\_Year



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

ALTER PROCEDURE Sales\_By\_Category

@OrdYear nvarchar(4) = '1998'

AS

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

BEGIN

DECLARE @msg nvarchar(50)

SET @msg = 'Please Select Year among 1996, 1997, 1998'

PRINT @msg

END

ELSE

BEGIN

SELECT C.CategoryName,

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

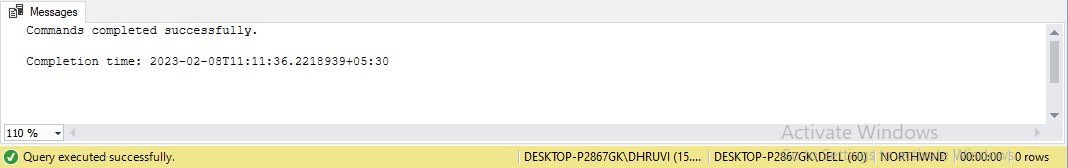
AND P.CategoryID = C.CategoryID

AND SUBSTRING(CONVERT(nvarchar(22), O.OrderDate, 111), 1, 4) = @OrdYear

GROUP BY C.CategoryName

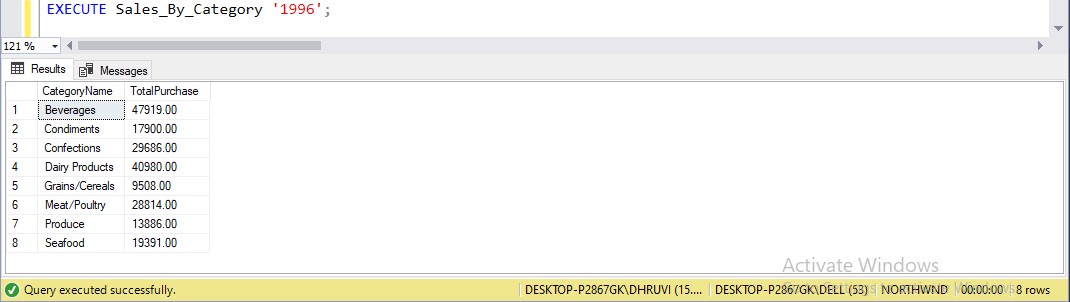
ORDER BY C.CategoryName;

END



-- Execute Store Procedure Sales\_By\_Category

EXECUTE Sales\_By\_Category '1996';



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

CREATE PROCEDURE Ten\_Most\_Expensive\_Products

AS

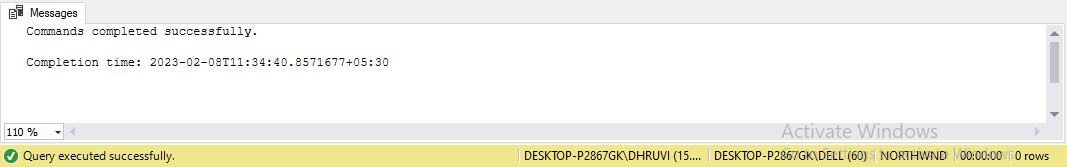
BEGIN

SELECT TOP 10 ProductName AS Ten\_Most\_Expensive\_Products, UnitPrice

FROM Products

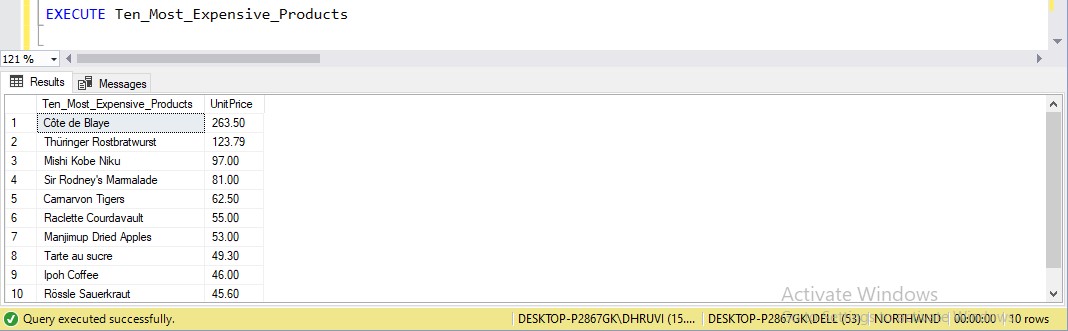
ORDER BY UnitPrice DESC

END



-- Execute Store Procedure Ten\_Most\_Expensive\_Products

EXECUTE Ten\_Most\_Expensive\_Products



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

CREATE PROCEDURE Insert\_Cust\_Order\_Details

@OrderId int, @ProductId int, @UnitPrince decimal(10,2),

@Quantity int, @Discouunt float

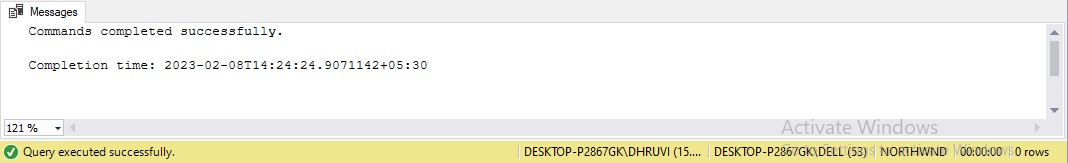
AS

BEGIN

INSERT INTO [Order Details]

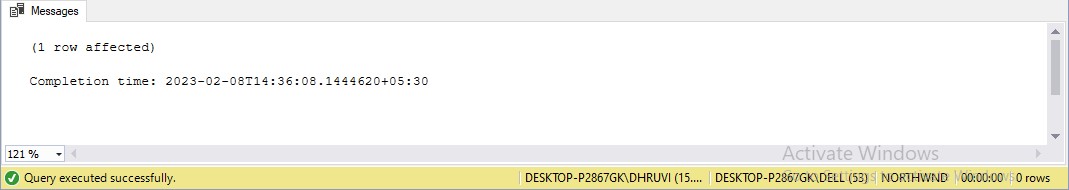
VALUES (@OrderId,@ProductId,@UnitPrince,@Quantity,@Discouunt);

END



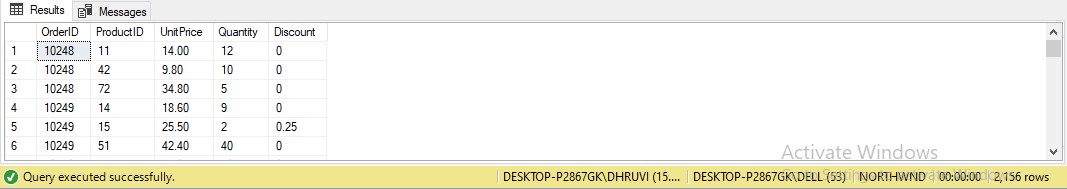
-- Execute Store Procedure Insert\_Cust\_Order\_Details

EXECUTE Insert\_Cust\_Order\_Details '10249','15','25.50','2','0.25'



-- Verify

SELECT \* FROM [Order Details]



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

CREATE PROCEDURE Update\_Cust\_Order\_Details

@OrderID int = NULL, @ProductId int = NULL, @UnitPrice Decimal(10,2) = NULL,

@Quantity smallint = NULL, @Discount real = NULL

AS

BEGIN

UPDATE [Order Details]

SET

OrderID = CASE WHEN @OrderID IS NOT NULL AND @OrderID <> OrderID THEN @OrderID ELSE OrderID END,

ProductId = CASE WHEN @ProductId IS NOT NULL AND @ProductId <> ProductID THEN @ProductId ELSE ProductId END,

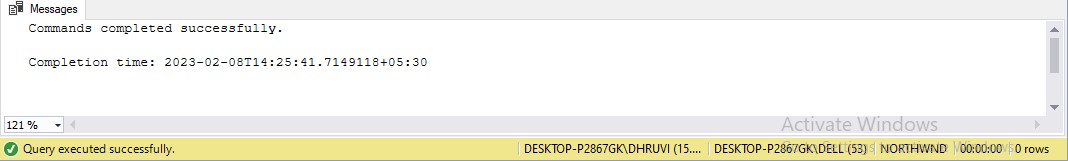
UnitPrice = CASE WHEN @UnitPrice IS NOT NULL AND @UnitPrice <> UnitPrice THEN @UnitPrice ELSE UnitPrice END,

Quantity = CASE WHEN @Quantity IS NOT NULL AND @Quantity <> Quantity THEN @Quantity ELSE Quantity END,

Discount = CASE WHEN @Quantity IS NOT NULL AND @Quantity <> Discount THEN @Quantity ELSE Discount END

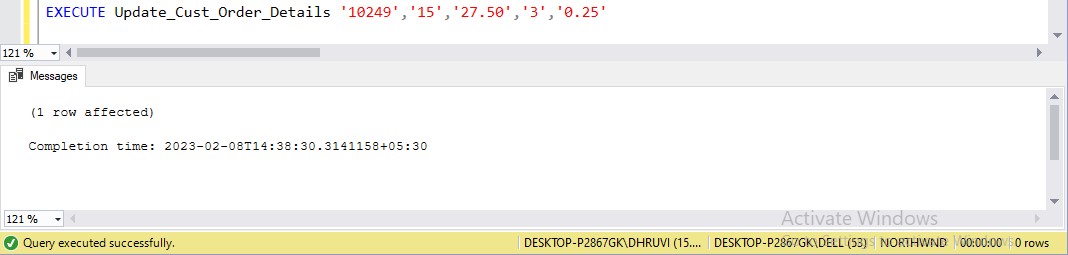
WHERE OrderID = @OrderID AND ProductID = @ProductId;

END



-- Execute Store Procedure Update\_Cust\_Order\_Details

EXECUTE Update\_Cust\_Order\_Details '10249','15','27.50','3','0.25'



-- Verify

SELECT \* FROM [Order Details]

