**SQL 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 PROCEDURE spGetAverageFreight (@CustomerID nchar(5))

AS

BEGIN

SET NOCOUNT ON;

DECLARE @AverageFreight decimal(10, 2);

SELECT @AverageFreight = AVG(Freight)

FROM Orders

WHERE CustomerID = @CustomerID;

SELECT @AverageFreight AS AverageFreight;

END;

CREATE TRIGGER tr\_Orders\_FreightCheck

ON Orders

FOR INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @AverageFreight decimal(10, 2);

DECLARE @Freight decimal(10, 2);

DECLARE @CustomerID nchar(5);

SELECT @CustomerID = i.CustomerID, @Freight = i.Freight

FROM inserted i;

EXEC spGetAverageFreight @CustomerID, @AverageFreight OUTPUT;

IF (@Freight > @AverageFreight)

BEGIN

RAISERROR ('The Freight value exceeds the average Freight for the specified customer.', 16, 1);

ROLLBACK TRANSACTION;

END;

END;

**2] write a SQL query to Create Stored procedure in the Northwind database to retrieve**

**Employee Sales by Country**

CREATE PROCEDURE spGetEmployeeSalesbyCountry (@Startdate DATETIME,@Enddate DATETIME)

AS

BEGIN

SELECT Employees.Country , SUM([Order Details].UnitPrice \* [Order Details].Quantity)

FROM Employees

INNER JOIN Orders

ON Employees.EmployeeID = Orders.EmployeeID

INNER JOIN [Order Details]

ON Orders.OrderID = [Order Details].OrderID

WHERE Orders.OrderDate BETWEEN @Startdate AND @Enddate

GROUP BY Employees.Country

END;

**3] write a SQL query to Create Stored procedure in the Northwind database to retrieve**

**Sales by Year**

CREATE PROCEDURE spGetSalesByYear (@Startdate DATETIME,@Enddate DATETIME)

AS

BEGIN

SELECT Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal, DATENAME(yy,ShippedDate) AS Year

FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID

WHERE Orders.ShippedDate Between @Startdate And @Enddate

END;

**4] write a SQL query to Create Stored procedure in the Northwind database to retrieve**

**Sales By Category**

CREATE PROCEDURE spGetSalesByCategory (@CategoryName nvarchar(15), @OrdYear nvarchar(4))

AS

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

BEGIN

SELECT @OrdYear = '1998'

END

BEGIN

SELECT @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 C.CategoryName = @CategoryName

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

GROUP BY ProductName

END;

**5] write a SQL query to Create Stored procedure in the Northwind database to retrieve**

**Ten Most Expensive Products**

CREATE PROCEDURE spGetTopTenMostExpensiveProducts

AS

BEGIN

SET ROWCOUNT 10

SELECT Products.ProductName AS TenMostExpensiveProducts, Products.UnitPrice

FROM Products

ORDER BY Products.UnitPrice DESC

END;

**6] write a SQL query to Create Stored procedure in the Northwind database to insert**

**Customer Order Details**

CREATE PROC spGetsmInsertOrderDetails

@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

**7] write a SQL query to Create Stored procedure in the Northwind database to update**

**Customer Order Details**

CREATE PROC spGetsmUpdateOrderDetails

@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