**Assignment-3**

--1. write a SQL query to find Employees who have the biggest salary in their Department

select d.d\_name, ms.maxSalary

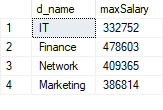
from department d join

(select department\_id, MAX(salary) as maxSalary

from employee

group by department\_id) as ms

on d.d\_id=ms.department\_id;



--2. write a SQL query to find Departments that have less than 3 people in it

select dc.dcount, d.d\_name

from department d join(

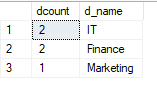
select department\_id,count(department\_id) as dcount

from employee

group by department\_id

having count(department\_id)<3) as dc

on d.d\_id=dc.department\_id;



--3. write a SQL query to find All Department along with the number of people there

select dc.dcount, d.d\_name

from department d join(

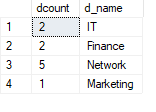
select department\_id,count(department\_id) as dcount

from employee

group by department\_id

) as dc

on d.d\_id=dc.department\_id;



--4. write a SQL query to find All Department along with the total salary there

select d.d\_name, ms.ss

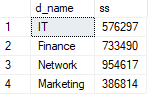
from department d join

(select sum(salary)as ss, department\_id

from employee

group by department\_id) as ms

on d.d\_id=ms.department\_id;



**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 CalculateAverageFreight

@CustomerID NVARCHAR(5)

AS

BEGIN

DECLARE @AverageFreight MONEY;

SELECT @AverageFreight = AVG(Freight)

FROM Orders

WHERE CustomerID = @CustomerID;

SELECT @AverageFreight AS AverageFreight;

END;

CREATE TRIGGER FreightValidation

ON Orders

INSTEAD OF INSERT, UPDATE

AS

BEGIN

IF (SELECT COUNT(\*)

FROM inserted) > 0

BEGIN

DECLARE @CustomerID NVARCHAR(5);

DECLARE @Freight MONEY;

SELECT @CustomerID = CustomerID, @Freight = Freight

FROM inserted;

DECLARE @AverageFreight MONEY;

--EXEC CalculateAverageFreight @CustomerID, @AverageFreight OUTPUT;

IF (@Freight > @AverageFreight)

BEGIN

RAISERROR('Freight exceeds average. Operation cancelled.', 5, 1)

ROLLBACK TRANSACTION;

RETURN;

END;

END;

-- If Freight is within limits or no records in inserted table, proceed with the original command.

INSERT INTO Orders (CustomerID,

EmployeeID,

OrderDate,

RequiredDate,

ShippedDate,

ShipVia,

Freight,

ShipName,

ShipAddress,

ShipCity,

ShipRegion,

ShipPostalCode,

ShipCountry)

SELECT CustomerID,

EmployeeID,

OrderDate,

RequiredDate,

ShippedDate,

ShipVia,

Freight,

ShipName,

ShipAddress,

ShipCity,

ShipRegion,

ShipPostalCode,

ShipCountry FROM inserted;

END;

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

CREATE PROCEDURE spSalesByCountry

AS

BEGIN

select sum(alldet.ptotal) as TotalSales,e.FirstName, alldet.ShipCountry

from Employees e join

(select o.OrderID,od.UnitPrice \* od.Quantity as ptotal ,EmployeeID,ShipCountry

from Orders o join [Order Details] od

on o.OrderID=od.OrderID) alldet

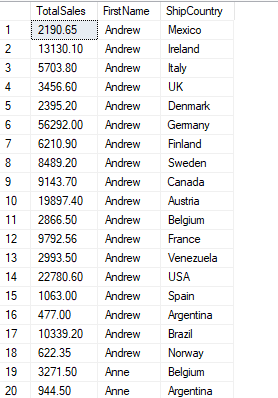
on e.EmployeeID=alldet.EmployeeID

group by e.FirstName, alldet.ShipCountry

order by e.FirstName;

END

GO



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

CREATE PROCEDURE spSalesByYear

AS

BEGIN

select sum(alldet.ptotal) as TotalSales,e.FirstName, YEAR(alldet.OrderDate)

from Employees e join

(select o.OrderID,od.UnitPrice \* od.Quantity as ptotal ,EmployeeID,o.OrderDate

from Orders o join [Order Details] od

on o.OrderID=od.OrderID) alldet

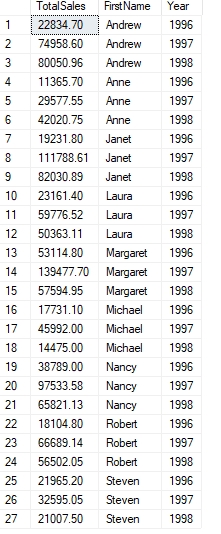
on e.EmployeeID=alldet.EmployeeID

group by e.FirstName, YEAR(alldet.OrderDate)

order by e.FirstName, YEAR(alldet.OrderDate);

END

GO



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

CREATE PROCEDURE spSalesByCategories

AS

BEGIN

select c.CategoryName , sum(alldet.ptotal) as TotalSales

from Categories c join

(select od.UnitPrice \* od.Quantity as ptotal , p.CategoryID

from Products p join [Order Details] od

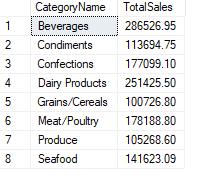
on p.ProductID=od.ProductID) alldet

on c.CategoryID=alldet.CategoryID

group by c.CategoryName;

END

GO



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

Create proc [dbo].[sp5TenMostExpensiveProducts]

as

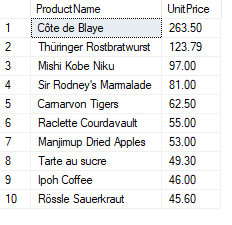
begin

select top 10 ProductName, UnitPrice

from Products

order by UnitPrice desc;

End



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

CREATE PROC SpInsertOrderDetail

@OrderID INTEGER,

@ProductID INTEGER,

@UnitPrice INTEGER,

@Quantity INTEGER,

@Discount INTEGER

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 spUpdateOrderDetail

@OrderID INTEGER,

@ProductID INTEGER,

@UnitPrice MONEY,

@Quantity SMALLINT,

@Discount REAL

AS

BEGIN

UPDATE [Order Details]

SET Quantity = @Quantity,

ProductID = @ProductID,

UnitPrice = @UnitPrice,

Discount = @Discount

WHERE OrderID = @OrderID

END