**Assignment 4**

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.

Creating stored procedure:

create procedure spCalculateAvgFreight

@customer\_id varchar(20)

as

begin

declare @avgval as float

select @avgval=avg(freight) from orders where customer\_id=@customer\_id

return @avgval

end

creating Trigger:

create trigger FreightValidation

on orders

for insert,update

as

begin

declare @Freight float,

@customerID varchar(20)

select @customerID=customer\_id from inserted

exec @Freight=spCalculateAvgFreight @customer\_id=@customerID

select @Freight

if exists (select \* from inserted where freight> @Freight)

begin

select freight as new\_freight from inserted

raiserror('freight exceed',16,1)

rollback transaction

end

else

print 'row added'

end

verifying data

update orders set freight =41 where customer\_id = 'ALFKI' and freight= 30

--enter successfully bcz avg freight is 42

insert into orders (freight, ship\_name, customer\_id,order\_id) values (45.00, 'Alfreds Futterkiste', 'ALFKI',11081)

--give error msg freigth exceeed as its value is more then average freight value

insert into orders (freight, ship\_name, customer\_id,order\_id) values (40, 'Alfreds Futterkiste', 'ALFKI',11081)

--inserted row successfully in table

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

Employee Sales by Country

create procedure spEmployeeSalesByCountryUpdated

@employee\_id smallint

AS

select ship\_country,count(order\_id) as[Sales By Country],employee\_id from orders group by ship\_country,employee\_id having employee\_id=@employee\_id

exec spEmployeeSalesByCountryUpdated @employee\_id=6



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

Sales by Year

create procedure spSalesByYear

@year date

As

select year(order\_date),count(order\_id) as [Total Sales Per Year] from orders group by year(order\_date) having year(order\_date)= year(@year)

exec spSalesByYear @year='2000-01-01'



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

Sales By Category

create procedure spSalesByCategory

@category\_name varchar(15)

As

select categories.category\_name,count(order\_id) as [Total Sales] from categories

left outer join products on categories.category\_id=products.category\_id

left outer join order\_details on products.product\_id=order\_details.product\_id

group by categories.category\_name having category\_name=@category\_name

exec spSalesByCategory @category\_name='Condiments'



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

Ten Most Expensive Products

create procedure ExpensiveProduct

As

select top 10 product\_name ,unit\_price from products order by unit\_price desc

exec ExpensiveProduct



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

Customer Order Details

create procedure spInsertOrderDetail

@order\_id smallint,

@product\_id smallint,

@unit\_price real,

@quantity smallint,

@discount real

As

insert into order\_details values (@order\_id,@product\_id,@unit\_price,@quantity,@discount)

exec spInsertOrderDetail @order\_id='10248',@product\_id='',@unit\_price='37.7',@quantity='34',@discount='0.05'





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

Customer Order Details

create procedure spUpdateOrderDetail

@order\_id smallint,

@product\_id smallint,

@unit\_price real,

@quantity smallint,

@discount real

As

update order\_details set unit\_price=@unit\_price,quantity=@quantity,discount=@discount where order\_id=@order\_id and product\_id=@product\_id

exec spUpdateOrderDetail @order\_id='10249',@product\_id='72',@unit\_price='60',@quantity='50',@discount='0.05'



