

--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.

The screenshot shows the SQL Server Enterprise Manager interface with the Northwind database selected. The SQL Query window displays the following code:

```

-- create procedure sp.ValidateFreight
-- -- inputted customer
-- -- returned average freight
-- -- returned average freight
-- @AverageFreight money output
as
begin
    select @AverageFreight = AVG(Freight)
    from Orders
    where CustomerID = @CustomerID
end
go

-- create or alter trigger tr_insteadOfInsert_orders
-- on Orders
-- instead of insert
-- as
-- begin
--     declare @AvgFreight money, @CustomerID nvarchar(5), @insertedFreight money

```

The Results window shows the output of the stored procedure for CustomerID 10248 (VINET), returning an average freight of 32.38. The Properties window on the right shows the connection details for the Northwind database.

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

The screenshot shows the SQL Server Enterprise Manager interface with the Northwind database selected. The SQL Query window displays the following code:

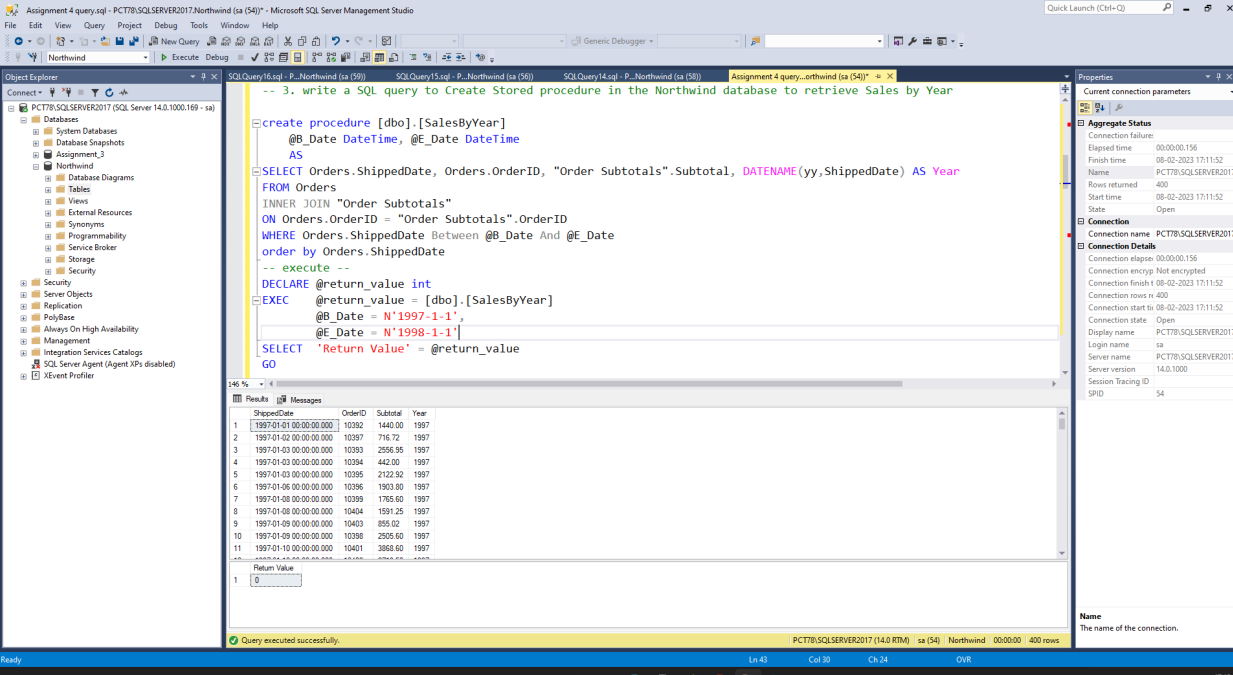
```

--2. write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country
-- alter procedure [dbo].[EmployeeSalesByCountry] @country nvarchar(15)
-- as
-- SELECT Employees.Country, Employees.FirstName, Employees.LastName, Orders.ShippedDate, Orders.OrderID,
-- "Order Subtotals".Subtotal AS SaleAmount
-- FROM Employees INNER JOIN
-- (Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID)
-- ON Employees.EmployeeID = Orders.EmployeeID
-- where Employees.country = @country
-- -- Execute --
-- DECLARE @return_value int
-- EXEC @return_value = [dbo].[EmployeeSalesByCountry]
-- @country = N'UK'
-- SELECT 'Return Value' = @return_value
-- GO

```

The Results window shows the output of the stored procedure for the UK, returning a list of employees and their sales amounts. The Properties window on the right shows the connection details for the Northwind database.

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



```

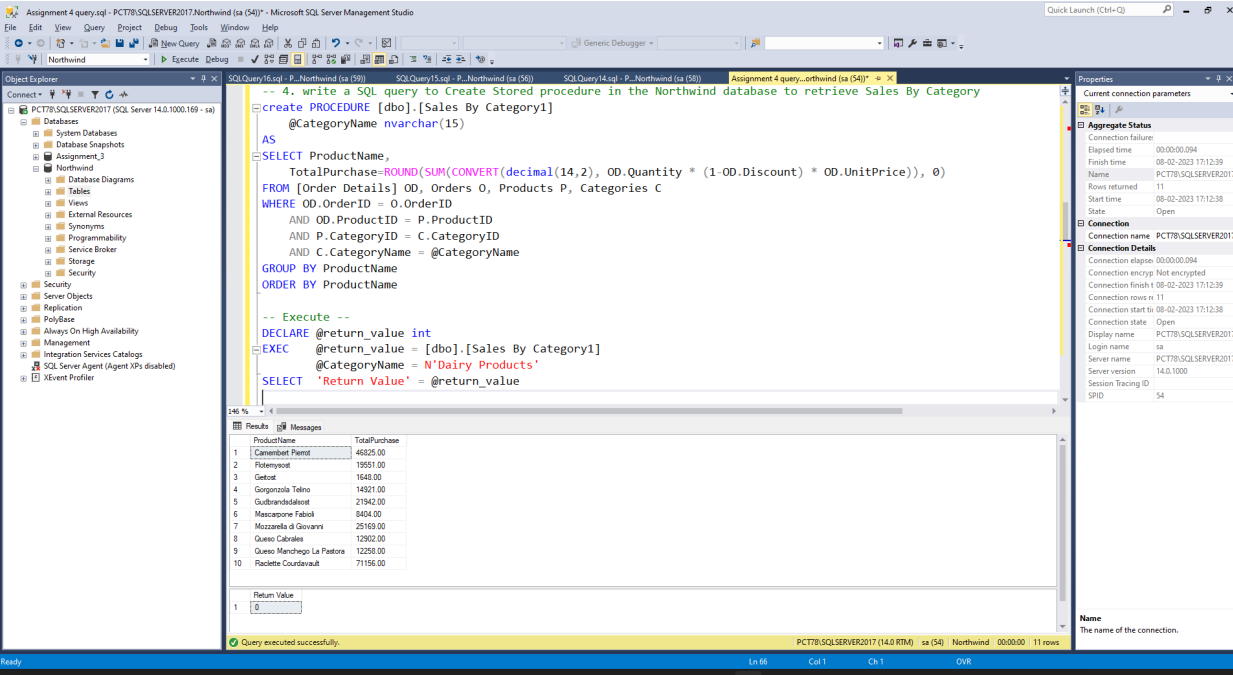
-- 3. write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales by Year

create procedure [dbo].[SalesByYear]
    @B_Date DateTime, @E_Date DateTime
AS
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 @B_Date And @E_Date
order by Orders.ShippedDate
-- execute --
DECLARE @return_value int
EXEC     @return_value = [dbo].[SalesByYear]
        @B_Date = N'1997-1-1',
        @E_Date = N'1998-1-1'
SELECT   'Return Value' = @return_value
GO

```

ShippedDate	OrderID	Subtotal	Year
1997-01-01 00:00:00.000	10392	1440.00	1997
1997-01-02 00:00:00.000	10397	716.72	1997
1997-01-03 00:00:00.000	10393	2556.95	1997
1997-01-03 00:00:00.000	10394	442.00	1997
1997-01-03 00:00:00.000	10395	2122.92	1997
1997-01-06 00:00:00.000	10396	1903.80	1997
1997-01-08 00:00:00.000	10399	1765.60	1997
1997-01-08 00:00:00.000	10404	1591.25	1997
1997-01-09 00:00:00.000	10403	895.02	1997
1997-01-09 00:00:00.000	10398	2950.60	1997
1997-01-10 00:00:00.000	10401	3868.60	1997

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



```

-- 4. write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales By Category
@CategoryName nvarchar(15)
AS
SELECT ProductName,
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
GROUP BY ProductName
ORDER BY ProductName
-- Execute --
DECLARE @return_value int
EXEC     @return_value = [dbo].[Sales By Category1]
        @CategoryName = N'Dairy Products'
SELECT   'Return Value' = @return_value
GO

```

ProductName	TotalPurchase
Camembert Pierrot	46825.00
Foie gras	19591.00
Gruyere	1648.00
Gorgonzola Telino	14921.00
Gudbrandsdalsost	21942.00
Mascarpone Fabozzi	8404.00
Mozzarella di Giovanni	25169.00
Queso Cabrales	12962.00
Queso Manchego La Pastora	12258.00
Raclette Courdavault	71156.00

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

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the Northwind database selected. The central pane contains a SQL query window with the following code:

```
-- 5. write a SQL query to Create Stored procedure in the Northwind database to retrieve Ten Most Expensive Products
create PROCEDURE topTenCoastlyProduct
as
select top 10 [ProductName], [UnitPrice] from dbo.Products order by UnitPrice DESC;
select* from Products

-- Execute --
DECLARE @return_value int
EXEC @return_value = [dbo].[topTenCoastlyProduct]
SELECT 'Return Value' = @return_value
```

The right pane shows the Properties window for the connection. The bottom status bar indicates the query was executed successfully, returning 11 rows.

ProductID	ProductName	UnitPrice
1	Côte de Boeuf	263.50
2	Thüringer Rostbraten	123.75
3	Mahy Kobe Niku	97.00
4	Sir Rodney's Marmalade	81.00
5	Caravanon Tigris	62.50
6	Rivellate Crouteau	55.00
7	Manjup Dried Apples	53.00
8	Tarte au sucre	49.30
9	Isoh Coffee	46.00
10	Pissot Steakrout	45.00

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

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the Northwind database selected. The central pane contains a SQL query window with the following code:

```
-- 6. write a SQL query to Create Stored procedure in the Northwind database to insert Customer Order Details
create procedure insertNewOrder
@OrderID int,
@ProductID int,
@UnitPrice money,
@Quantity smallInt,
@Discount real
as
insert into [dbo].[Order Details] values (@OrderID, @ProductID, @UnitPrice, @Quantity, @Discount)

--Execute --
DECLARE @return_value int
EXEC @return_value = [dbo].[insertNewOrder]
@OrderID = 10249,
@ProductID = 74,
@UnitPrice = 34.80,
@Quantity = 4,
@Discount = 0.00
SELECT 'Return Value' = @return_value
```

The right pane shows the Properties window for the connection. The bottom status bar indicates the query was executed successfully, returning 2158 rows.

OrderID	ProductID	UnitPrice	Quantity	Discount	
1	10248	11	14.00	12	0
2	10248	42	9.80	3	0
3	10248	72	34.80	5	0
4	10249	14	18.60	9	0
5	10249	51	42.40	40	0
6	10249	72	34.80	4	0
7	10249	74	34.80	4	0
8	10250	41	7.70	10	0
9	10250	51	42.40	35	0.15
10	10250	65	16.80	15	0.15
11	10250	72	38.00	4	0.15
12	10251	22	16.80	6	0.05
13	10251	57	15.60	15	0.05
14	10251	65	16.80	20	0

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

The screenshot displays the Microsoft SQL Server Management Studio interface. The central pane shows a SQL query titled "Assignment 4 query...northwind (sa [54])". The query is as follows:

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

create procedure UpdateOrder
@OrderID int ,
@ProductID int,
@Quantity smallint
as
update [dbo].[Order Details]
set
Quantity = @Quantity
where OrderId = @OrderID AND ProductId = @ProductID

--execute --
DECLARE @return_value int
EXEC @return_value = [dbo].[UpdateOrder]
        @OrderID = 10249,
        @ProductID = 74,
        @Quantity = 2

SELECT 'Return Value' = @return_value
```

The bottom pane shows the results of the query execution, displaying a table with 16 rows and 5 columns: OrderID, ProductID, UnitPrice, Quantity, and Discount. The 7th row is highlighted, showing OrderID 10249, ProductID 74, UnitPrice 34.80, Quantity 2, and Discount 0.

	OrderID	ProductID	UnitPrice	Quantity	Discount
1	10248	11	14.00	12	0
2	10248	42	9.80	3	0
3	10248	72	34.80	5	0
4	10248	14	18.60	9	0
5	10249	51	42.40	40	0
6	10249	72	34.80	4	0
7	10249	74	34.80	2	0
8	10250	41	7.70	10	0
9	10250	51	42.40	35	0.15
10	10250	65	16.80	15	0.15
11	10250	72	38.00	4	0.15
12	10251	22	16.80	6	0.05
13	10251	57	15.60	15	0.05
14	10251	65	16.80	20	0
15	10252	20	64.80	40	0.05
16	10252	33	2.00	25	0.05

The right-hand pane shows the Properties window for the current connection, displaying details such as Connection name, Connection string, and Connection state.