**Week 2: Advanced SQL, Nunit and Moq**

**SQL Exercises: Advanced SQL Concepts**

**Exercise 1: RANKING AND WINDOW FUNCTIONS**

**QUERY:**

WITH RankedProducts AS (

SELECT

ProductID,

ProductName,

Category,

Price,

ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum,

RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankNum,

DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRankNum

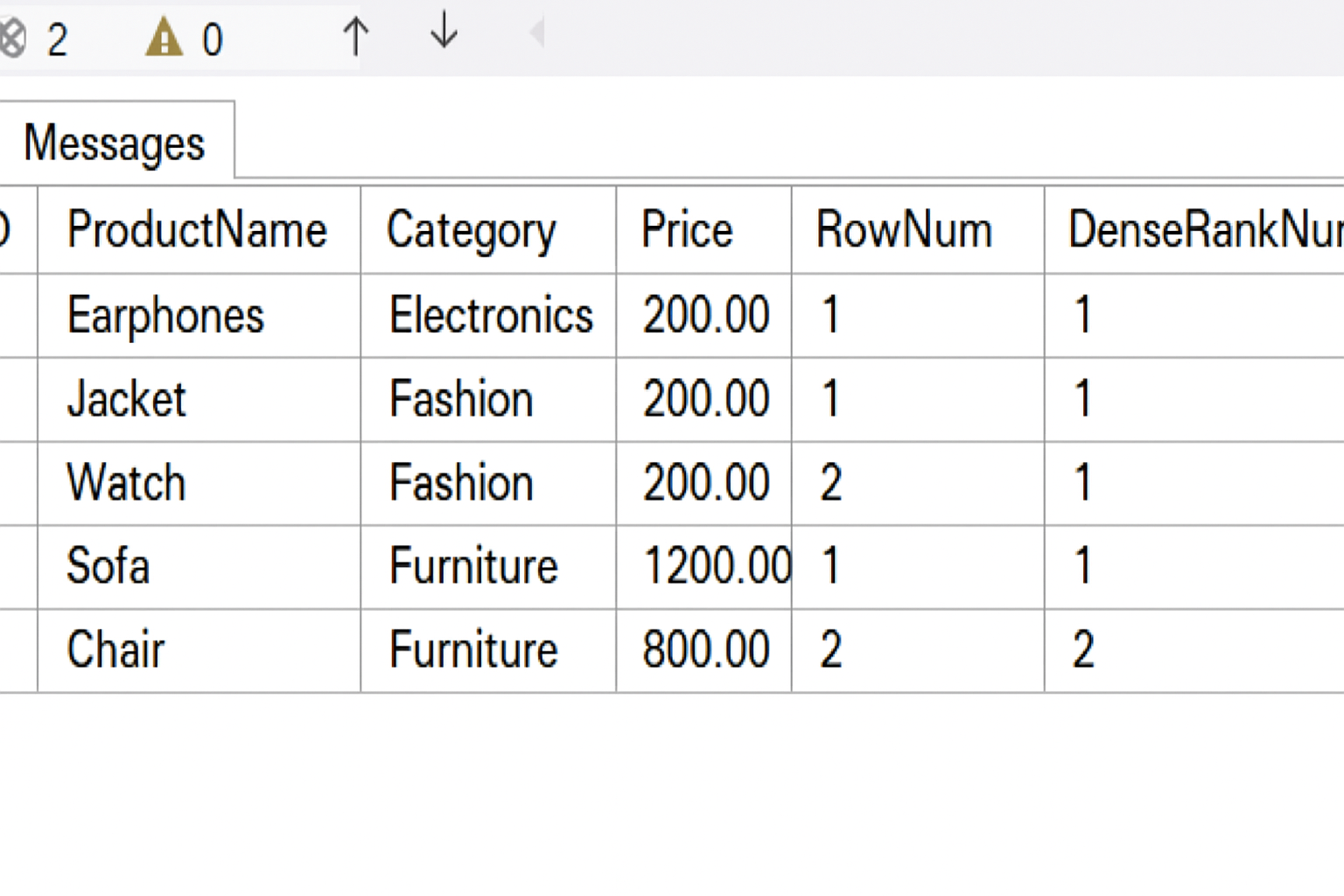
from

Products

)

select \* from RankedProducts where RowNum <= 3;

**Output:**

****

**SQL Exercise: Stored Procedure**

**Exercise 1: Create a Stored Procedure**

**1.Define the stored procedure with a parameter for DepartmentID.**

Create PROCEDURE sp\_GetEmployeesByDepartment

@DepartmentID INT

AS

BEGIN

select \* from Employees

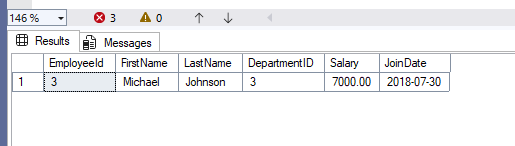
where DepartmentID = @DepartmentID;

END;

**2. Select employee based on DepartmentID**

SELECT \* FROM Employees WHERE DepartmentID = 3;

**Output:**



**3. Stored procedure named ‘sp\_InsertEmployee’**

EXEC sp\_InsertEmployee

@FirstName = 'Sachin',

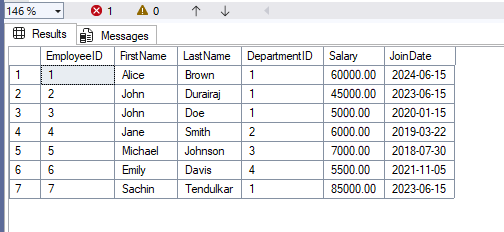
@LastName = 'Tendulkar',

@DepartmentID = 1,

@Salary = 85000.00,

@JoinDate = '2023-06-15';

**Output:**

****

**SQL Exercises: Functions**

**Exercise 7: Return data from Scalar function**

**1.Execute the `fn\_CalculateAnnualSalary` function for an employee with `EmployeeID = 1`.**

create FUNCTION dbo.fn\_CalculateAnnualSalary (@EmpID INT)

returns DECIMAL(12,2)

AS

BEGIN

DECLARE @Annual DECIMAL(12,2);

select @Annual = Salary \* 12

from Employees where EmployeeID = @EmpID;

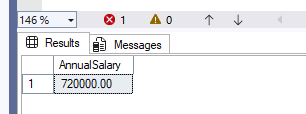
RETURN @Annual;

END;

GO

SELECT dbo.fn\_CalculateAnnualSalary(1) AS AnnualSalary;

**Output:**

****

**2. Verify the result**

SELECT e.EmployeeID,

e.FirstName,

e.LastName,

d.DepartmentName,

e.Salary AS MonthlySalary,

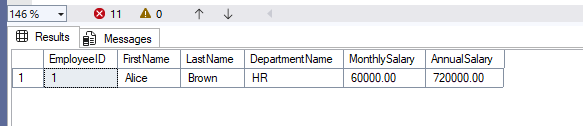
dbo.fn\_CalculateAnnualSalary(e.EmployeeID) AS AnnualSalary

FROM Employees e

JOIN Departments d ON d.DepartmentID = e.DepartmentID

WHERE e.EmployeeID = 1;

**Output:**



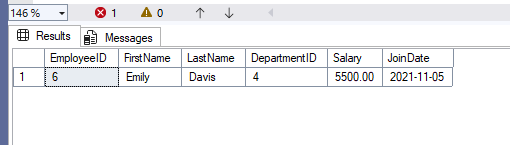
**SQL Exercise: Stored Procedure**

**Exercise 4: Execute a stored procedure**

**1.To Execute the stored procedure**

EXEC sp\_GetEmployeesByDepartment @DepartmentID = 4;

**Output:**

****

**SQL Exercise: Stored Procedure**

**Exercise 5: Return Data from a Stored Procedure**

**1.Stored Procedure with DepartmentdID**

create PROCEDURE sp\_GetEmployeeCountByDepartment

@DepartmentID INT

AS

BEGIN

select COUNT(\*) AS EmployeeCount from Employees

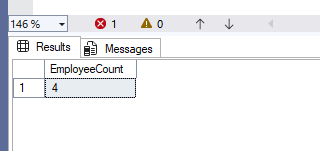
where DepartmentID = @DepartmentID;

END;

**2.Query for count the number of employees in the department**

EXEC sp\_GetEmployeeCountByDepartment @DepartmentID = 1;

**Output:**

****