**1.)Exercise 1: Ranking and Window Functions**

**Code:**

/\* creating the table \*/

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10, 2)

);

INSERT INTO Products (ProductID, ProductName, Category, Price) VALUES

(1, 'Laptop A', 'Electronics', 1200.00),

(2, 'Laptop B', 'Electronics', 1000.00),

(3, 'Laptop C', 'Electronics', 1000.00),

(4, 'Phone A', 'Electronics', 800.00),

(5, 'Shirt A', 'Clothing', 50.00),

(6, 'Shirt B', 'Clothing', 70.00),

(7, 'Shirt C', 'Clothing', 70.00),

(8, 'Jacket A', 'Clothing', 100.00),

(9, 'Blender A', 'Appliances', 300.00),

(10, 'Blender B', 'Appliances', 200.00),

(11, 'Blender C', 'Appliances', 300.00);

/\*Applying ranking functions\*/

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;

/\*Fetching 3 most expensive product in each category\*/

WITH RankedProducts AS (

SELECT

ProductID,

ProductName,

Category,

Price,

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

FROM Products

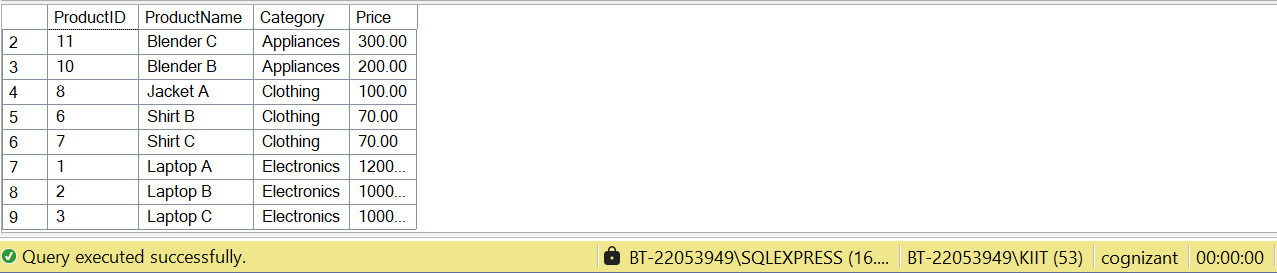
)

SELECT ProductID, ProductName, Category, Price

FROM RankedProducts

WHERE RowNum <= 3;

**Output:**



**2.) Exercise 1: Create a Stored Procedure**

**Code:**

/\* creating the table \*/

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT FOREIGN KEY REFERENCES Departments(DepartmentID),

Salary DECIMAL(10,2),

JoinDate DATE

);

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

INSERT INTO Employees (EmployeeID, FirstName, LastName, DepartmentID, Salary,

JoinDate) VALUES

(1, 'John', 'Doe', 1, 5000.00, '2020-01-15'),

(2, 'Jane', 'Smith', 2, 6000.00, '2019-03-22'),

(3, 'Michael', 'Johnson', 3, 7000.00, '2018-07-30'),

(4, 'Emily', 'Davis', 4, 5500.00, '2021-11-05');

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

**Code:**

CREATE PROCEDURE sp\_GetEmployeesByDepartment

@DepartmentID INT

AS

BEGIN

SELECT

E.EmployeeID,

E.FirstName,

E.LastName,

E.Salary,

E.JoinDate,

D.DepartmentName

FROM Employees E

INNER JOIN Departments D ON E.DepartmentID = D.DepartmentID

WHERE E.DepartmentID = @DepartmentID;

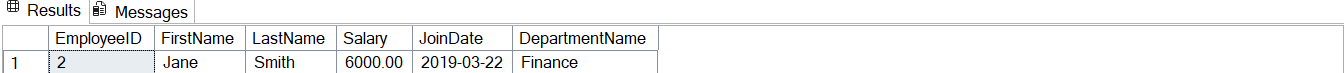
END;

**2.Write the SQL query to select employee details based on the DepartmentID.**

**Code:**

EXEC sp\_GetEmployeesByDepartment @DepartmentID = 2;

**Output:**



**3. Create a stored procedure named `sp\_InsertEmployee` with the following code:**

**Code:**

CREATE PROCEDURE sp\_InsertEmployee

@FirstName VARCHAR(50),

@LastName VARCHAR(50),

@DepartmentID INT,

@Salary DECIMAL(10,2),

@JoinDate DATE

AS

BEGIN

INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, JoinDate)

VALUES (@FirstName, @LastName, @DepartmentID, @Salary, @JoinDate);

END;

**3.)Exercise 5: Return Data from a Stored Procedure**

**Code:**

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

@DepartmentID INT

AS

BEGIN

SELECT COUNT(\*) AS TotalEmployees

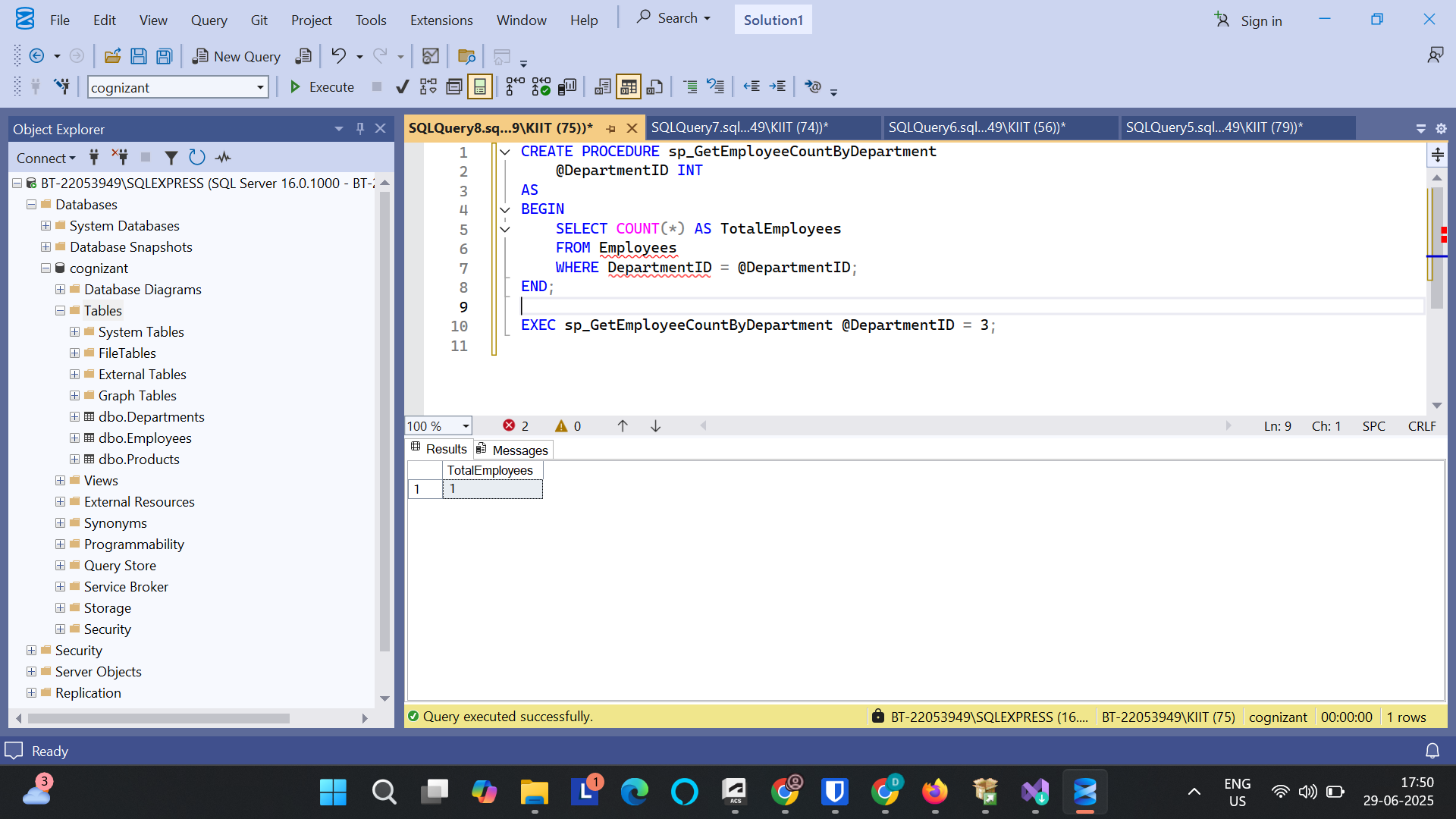
FROM Employees

WHERE DepartmentID = @DepartmentID;

END;

EXEC sp\_GetEmployeeCountByDepartment @DepartmentID = 2;

**Output:**

****