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

Goal: Create a stored procedure that returns the total number of employees in a

department.

Steps:

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

2. Write the SQL query to count the number of employees in the specified department.

3. Save the stored procedure by executing the Stored procedure content.

Sql:  
USE sp\_GetEmployeeCountByDepartment;

GO

IF OBJECT\_ID('sp\_GetEmployeeCountByDepartment', 'P') IS NOT NULL

DROP PROCEDURE sp\_GetEmployeeCountByDepartment;

GO

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

@DepartmentID INT

AS

BEGIN

SET NOCOUNT ON;

SELECT COUNT(\*) AS EmployeeCount

FROM Employees

WHERE DepartmentID = @DepartmentID;

END;

GO

C#:  
using System;

using System.Data;

using System.Data.SqlClient;

namespace EmployeeCountApp

{

class Program

{

static void Main(string[] args)

{

string connectionString = @"Data Source=localhost;Initial Catalog=sp\_GetEmployeeCountByDepartment;Integrated Security=True";

Console.Write("Enter Department ID to count employees: ");

if (!int.TryParse(Console.ReadLine(), out int departmentId))

{

Console.WriteLine("Invalid input. Department ID must be a number.");

return;

}

try

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

using (SqlCommand cmd = new SqlCommand("sp\_GetEmployeeCountByDepartment", conn))

{

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@DepartmentID", departmentId);

conn.Open();

object result = cmd.ExecuteScalar();

int count = result != null ? Convert.ToInt32(result) : 0;

Console.WriteLine($"\nTotal employees in Department {departmentId}: {count}");

}

}

}

catch (SqlException ex)

{

Console.WriteLine("SQL Error: " + ex.Message);

}

catch (Exception ex)

{

Console.WriteLine("Unexpected Error: " + ex.Message);

}

Console.WriteLine("\nPress any key to exit...");

Console.ReadKey();

}

}

}

Output:  
