# WEEK 2 ADVANCED SQL (STORED PROCEDURE)

THE CODE:

DROP PROCEDURE IF EXISTS sp\_GetEmployeesByDepartment;

DROP PROCEDURE IF EXISTS sp\_InsertEmployee;

DROP PROCEDURE IF EXISTS sp\_GetEmployeeCountByDepartment;

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Departments;

GO

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

FirstName VARCHAR(50),

LastName VARCHAR(50),

DepartmentID INT FOREIGN KEY REFERENCES Departments(DepartmentID),

Salary DECIMAL(10,2),

JoinDate DATE

);

GO

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

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

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

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

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

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

GO

CREATE PROCEDURE sp\_GetEmployeesByDepartment

@DepartmentID INT

AS

BEGIN

SELECT

EmployeeID,

FirstName,

LastName,

DepartmentID,

Salary,

JoinDate

FROM Employees

WHERE DepartmentID = @DepartmentID;

END;

GO

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;

GO

CREATE PROCEDURE sp\_GetEmployeeCountByDepartment

@DepartmentID INT

AS

BEGIN

SELECT COUNT(\*) AS TotalEmployees

FROM Employees

WHERE DepartmentID = @DepartmentID;

END;

GO

DROP PROCEDURE IF EXISTS sp\_UpdateEmployeeSalary;

GO

CREATE PROCEDURE sp\_UpdateEmployeeSalary

@EmployeeID INT,

@NewSalary DECIMAL(10,2)

AS

BEGIN

UPDATE Employees

SET Salary = @NewSalary

WHERE EmployeeID = @EmployeeID;

END;

GO

EXEC sp\_UpdateEmployeeSalary 1, 5500.00;

DROP PROCEDURE IF EXISTS sp\_UpdateSalaryWithTransaction;

GO

CREATE PROCEDURE sp\_UpdateSalaryWithTransaction

@EmployeeID INT,

@NewSalary DECIMAL(10, 2)

AS

BEGIN

BEGIN TRY

BEGIN TRANSACTION;

UPDATE Employees

SET Salary = @NewSalary

WHERE EmployeeID = @EmployeeID;

IF @@ROWCOUNT = 0

BEGIN

THROW 50001, 'Employee not found.', 1;

END

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION;

THROW;

END CATCH

END;

GO

DROP PROCEDURE IF EXISTS sp\_GetEmployeesByDynamicFilter;

GO

CREATE PROCEDURE sp\_GetEmployeesByDynamicFilter

@FilterColumn NVARCHAR(50),

@FilterValue NVARCHAR(100)

AS

BEGIN

DECLARE @SQL NVARCHAR(MAX);

SET @SQL = N'

SELECT EmployeeID, FirstName, LastName, DepartmentID, Salary, JoinDate

FROM Employees

WHERE ' + QUOTENAME(@FilterColumn) + ' = @FilterValue';

EXEC sp\_executesql @SQL, N'@FilterValue NVARCHAR(100)', @FilterValue = @FilterValue;

END;

GO

DROP PROCEDURE IF EXISTS sp\_UpdateSalaryWithErrorHandling;

GO

CREATE PROCEDURE sp\_UpdateSalaryWithErrorHandling

@EmployeeID INT,

@NewSalary DECIMAL(10,2)

AS

BEGIN

BEGIN TRY

UPDATE Employees

SET Salary = @NewSalary

WHERE EmployeeID = @EmployeeID;

IF @@ROWCOUNT = 0

BEGIN

THROW 51000, 'Employee ID not found.', 1;

END

PRINT 'Salary updated successfully.';

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

SET @ErrorMessage = 'Error: ' + ERROR\_MESSAGE();

PRINT @ErrorMessage;

END CATCH

END;

GO

