**Employee Management System SQL Exercises**

**Database Schema**

The following schema defines the structure for an Employee Management System:

**Departments Table**

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

**Employees Table**

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

);

**Sample Data**

The following sample data can be used for testing:

**Departments Sample Data**

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

**Employees Sample Data**

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');

**Exercises**

**Exercise 1: Create a Stored Procedure**

Goal: Create a stored procedure to retrieve employee details by department.

Steps:

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

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

3. Create a stored procedure named `sp\_InsertEmployee` with the following 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;

**CODE:**

-- Drop the existing Employees table if it exists

DROP TABLE IF EXISTS Employees;

-- Create Departments Table

CREATE TABLE Departments (

    DepartmentID INT PRIMARY KEY,

    DepartmentName VARCHAR(100)

);

-- Create Employees Table with EmployeeID as an Identity Column

CREATE TABLE Employees (

    EmployeeID INT IDENTITY(1,1) PRIMARY KEY,

    FirstName VARCHAR(50),

    LastName VARCHAR(50),

    DepartmentID INT,

    Salary DECIMAL(10,2),

    JoinDate DATE,

    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

-- Insert Sample Data into Departments

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

-- Insert Sample Data into Employees

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');

-- Create Stored Procedure to Retrieve Employee Details by Department

GO

CREATE PROCEDURE sp\_GetEmployeesByDepartment

    @DepartmentID INT

AS

BEGIN

    SELECT

        EmployeeID,

        FirstName,

        LastName,

        Salary,

        JoinDate

    FROM

        Employees

    WHERE

        DepartmentID = @DepartmentID;

END;

GO

-- Create Stored Procedure to Insert a New Employee

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

-- Example of how to execute the stored procedure to retrieve employees by department

EXEC sp\_GetEmployeesByDepartment @DepartmentID = 1;

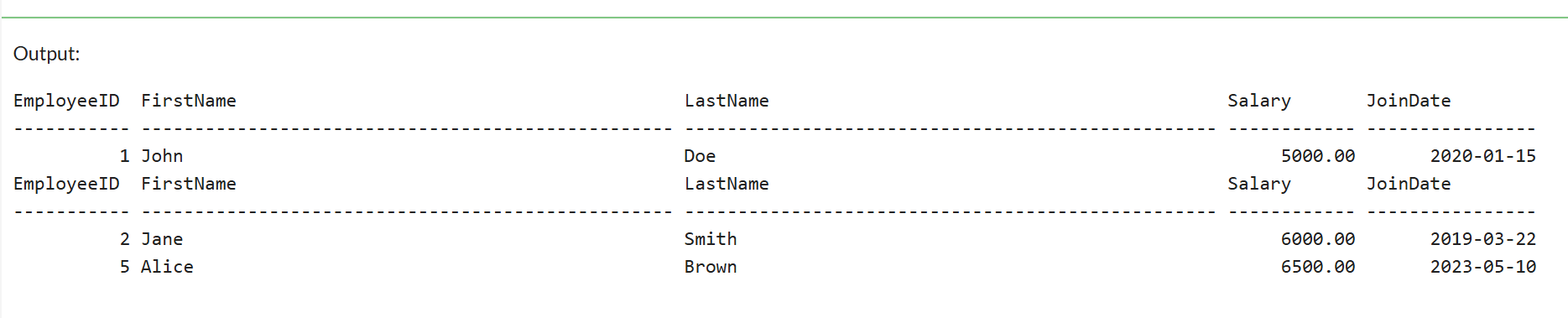
-- Example of how to execute the stored procedure to insert a new employee

EXEC sp\_InsertEmployee @FirstName = 'Alice', @LastName = 'Brown', @DepartmentID = 2, @Salary = 6500.00, @JoinDate = '2023-05-10';

-- After inserting a new employee, you can retrieve the employees again to see the updated list

EXEC sp\_GetEmployeesByDepartment @DepartmentID = 2;

**OUTPUT:**



**Exercise 4: Execute a Stored Procedure**

Goal: Execute the stored procedure to retrieve employee details for a specific department.

Steps:

1. Write the SQL command to execute the stored procedure with a DepartmentID

parameter.

2. Execute the command and review the results.

**CODE:**

-- Drop the existing Employees table if it exists

DROP TABLE IF EXISTS Employees;

-- Create Departments Table

CREATE TABLE Departments (

    DepartmentID INT PRIMARY KEY,

    DepartmentName VARCHAR(100)

);

-- Create Employees Table with EmployeeID as an Identity Column

CREATE TABLE Employees (

    EmployeeID INT IDENTITY(1,1) PRIMARY KEY,

    FirstName VARCHAR(50),

    LastName VARCHAR(50),

    DepartmentID INT,

    Salary DECIMAL(10,2),

    JoinDate DATE,

    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

-- Insert Sample Data into Departments

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

-- Insert Sample Data into Employees

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');

-- Create Stored Procedure to Retrieve Employee Details by Department

GO

CREATE PROCEDURE sp\_GetEmployeesByDepartment

    @DepartmentID INT

AS

BEGIN

    SELECT

        EmployeeID,

        FirstName,

        LastName,

        Salary,

        JoinDate

    FROM

        Employees

    WHERE

        DepartmentID = @DepartmentID;

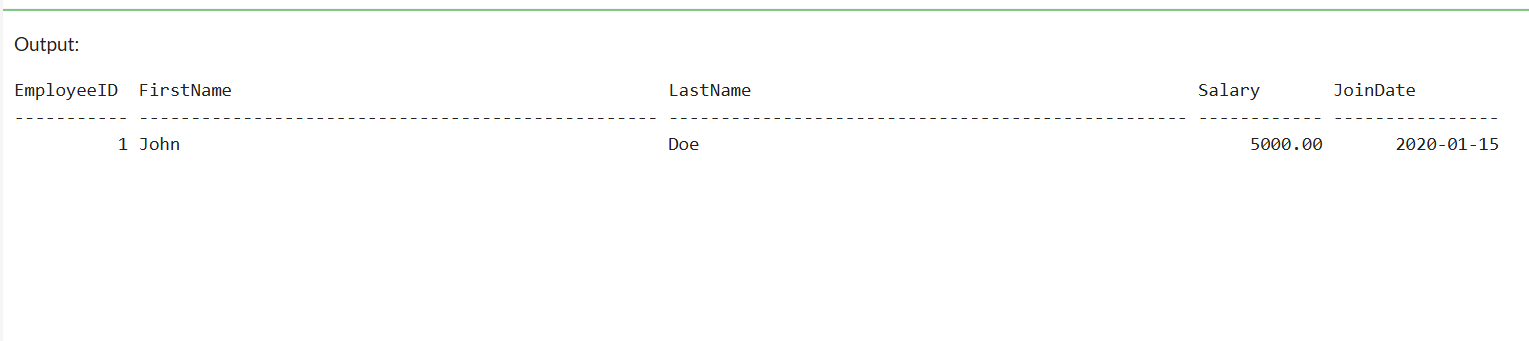
END;

GO

-- Execute the stored procedure to retrieve employees from the HR department (DepartmentID = 1)

EXEC sp\_GetEmployeesByDepartment @DepartmentID = 1;

**OUTPUT:**



**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.

**CODE:**

-- Drop the existing Employees table if it exists

DROP TABLE IF EXISTS Employees;

-- Create Departments Table

CREATE TABLE Departments (

    DepartmentID INT PRIMARY KEY,

    DepartmentName VARCHAR(100)

);

-- Create Employees Table with EmployeeID as an Identity Column

CREATE TABLE Employees (

    EmployeeID INT IDENTITY(1,1) PRIMARY KEY,

    FirstName VARCHAR(50),

    LastName VARCHAR(50),

    DepartmentID INT,

    Salary DECIMAL(10,2),

    JoinDate DATE,

    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

-- Insert Sample Data into Departments

INSERT INTO Departments (DepartmentID, DepartmentName) VALUES

(1, 'HR'),

(2, 'Finance'),

(3, 'IT'),

(4, 'Marketing');

-- Insert Sample Data into Employees

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');

-- Create Stored Procedure to Return Total Number of Employees in a Department

GO

CREATE PROCEDURE sp\_GetTotalEmployeesInDepartment

    @DepartmentID INT

AS

BEGIN

    SELECT COUNT(\*) AS TotalEmployees

    FROM Employees

    WHERE DepartmentID = @DepartmentID;

END;

GO

-- Example of how to execute the stored procedure to get the total number of employees in the HR department

EXEC sp\_GetTotalEmployeesInDepartment @DepartmentID = 1;

**OUTPUT:**

