```
--Create Database----

CREATE DATABASE CompanyManagementSystem




---Create Tables-----


--* Department Table:-

Create table Department(

 Department_Id INT PRIMARY KEY ,

 Department_Name Varchar(100)

);



--* Employee Table:-

CREATE TABLE Employee(

    Employee_Id INT PRIMARY KEY,

    First_Name VARCHAR(50),

    Last_Name VARCHAR(50),

    Email VARCHAR(100),

    Phone_Number VARCHAR(15),

    Hire_Date DATE,

    Job_Title VARCHAR(50),

    Salary DECIMAL(10, 2),

    Department_Id INT,

    FOREIGN KEY (Department_Id) REFERENCES Department(Department_Id)

);
```

```sql
--* Project Table:-
CREATE TABLE Project (
    Project_Id INT PRIMARY KEY,
    Project_Name VARCHAR(100),
    Start_Date DATE,
    End_Date DATE,
    Budget DECIMAL(12, 2)
);



--* Employee Table:-
CREATE TABLE EmployeeProject (
    EmployeeProject_Id INT PRIMARY KEY,
    Employee_Id INT,
    Project_Id INT,
    FOREIGN KEY (Employee_Id) REFERENCES Employee(Employee_Id),
    FOREIGN KEY (Project_Id) REFERENCES Project(Project_Id)
);



----Stored Procedures:-

--1.Write a stored procedure to insert a new employee into the `Employee` table. Include
--parameters for first name, last name, email, phone number, hire date, job title, salary, and
--department ID.
```

```
Alter PROCEDURE [dbo].[InsertEmployee]
 @EmployeeId INT,
    @FirstName VARCHAR(50),
    @LastName VARCHAR(50),
    @Email VARCHAR(100),
    @PhoneNumber VARCHAR(15),
    @HireDate DATE,
    @JobTitle VARCHAR(50),
    @Salary DECIMAL(10, 2),
    @DepartmentId INT
AS
BEGIN
     INSERT INTO [dbo].[Employee] (Employee_Id,First_Name, Last_Name, Email, Phone_Number,
Hire_Date, Job_Title, Salary, Department_Id)
       VALUES ( @EmployeeId,@FirstName, @LastName, @Email, @PhoneNumber, @HireDate,
@JobTitle, @Salary, @DepartmentId);

    PRINT 'Data Inserted in Employee Table';
END;



--2. Implement a stored procedure to retrieve an employee's details from the `Employee` table
--based on the employee ID provided as a parameter.
Alter PROCEDURE [dbo].[GetEmployeeDetails]
    @EmployeeId INT
AS
BEGIN
```

```
    SELECT
        [dbo].[Employee].Employee_Id,
        [dbo].[Employee].First_Name,
        [dbo].[Employee].Last_Name,
        [dbo].[Employee].Email,
        [dbo].[Employee].Phone_Number,
        [dbo].[Employee].Hire_Date,
        [dbo].[Employee].Job_Title,
        [dbo].[Employee].Salary,
        [dbo].[Employee].Department_Id
    FROM [dbo].[Employee]
    WHERE [dbo].[Employee].Employee_Id = @EmployeeId;
END;

--3.Develop a stored procedure that updates an existing employee's information in the
--`Employee` table. Include parameters for employee ID, first name, last name, email, phone
--number, hire date, job title, salary, and department ID.
CREATE PROCEDURE [dbo].[UpdateEmployee]
    @EmployeeId INT,
    @FirstName NVARCHAR(50),
    @LastName NVARCHAR(50),
    @Email NVARCHAR(100),
    @PhoneNumber NVARCHAR(20),
    @HireDate DATE,
    @JobTitle NVARCHAR(100),
    @Salary DECIMAL(10, 2),
    @DepartmentId INT
```

```
AS
BEGIN
    UPDATE [dbo].[Employee]
    SET
        First_Name = @FirstName,
        Last_Name = @LastName,
        Email = @Email,
        Phone_Number = @PhoneNumber,
        Hire_Date = @HireDate,
        Job_Title = @JobTitle,
        Salary = @Salary,
        Department_Id = @DepartmentId
    WHERE Employee_Id = @EmployeeId;
END;



--4. Create a stored procedure to delete an employee from the `Employee` table based on the
--employee ID provided as a parameter.
CREATE PROCEDURE [dbo].[DeleteEmployee]
    @EmployeeId INT
AS
BEGIN
    DELETE FROM [dbo].[Employee]
    WHERE Employee_Id = @EmployeeId;
END;
```

```sql
--5.Write a stored procedure to fetch all employees belonging to a specific department from
--the `Employee` table. Use the department ID as a parameter.
CREATE PROCEDURE [dbo].[GetEmployeesByDepartment]
    @DepartmentId INT
AS
BEGIN
    SELECT
        Employee_Id,
        First_Name,
        Last_Name,
        Email,
        Phone_Number,
        Hire_Date,
        Job_Title,
        Salary,
        Department_Id
    FROM [dbo].[Employee]
    WHERE Department_Id = @DepartmentId;
END;


--6.Develop a stored procedure to retrieve all projects assigned to a specific employee from
--the `Project` table. Use the employee ID as a parameter.
CREATE PROCEDURE [dbo].[GetProjectsByEmployee]
    @EmployeeId INT
AS
BEGIN
```

```sql
    SELECT
        p.Project_Id,
        p.Project_Name,
        p.Start_Date,
        p.End_Date,
        p.Budget
    FROM
        [dbo].[Project] p
    INNER JOIN
        [dbo].[EmployeeProject] ep ON p.Project_Id = ep.Project_Id
    WHERE
        ep.Employee_Id = @EmployeeId;
END;

--7.Implement a stored procedure to fetch all employees who are assigned to a particular
--project from the `Employee` table. Use the project ID as a parameter.
CREATE PROCEDURE [dbo].[GetEmployeesByProject]
    @ProjectId INT
AS
BEGIN
    SELECT
        e.Employee_Id,
        e.First_Name,
        e.Last_Name,
        e.Email,
        e.Phone_Number,
        e.Hire_Date,
        e.Job_Title,
        e.Salary,
        e.Department_Id
    FROM
        [dbo].[Employee] e
    INNER JOIN
        [dbo].[EmployeeProject] ep ON e.Employee_Id = ep.Employee_Id
    WHERE
        ep.Project_Id = @ProjectId;
END;
```

--8.Create a stored procedure to fetch details of a specific department from the `Department` --table along with the count of employees in that department. Use the department ID as a parameter.

```
CREATE PROCEDURE [dbo].[GetDepartmentDetails]
    @DepartmentId INT
AS
BEGIN
    SELECT
        d.Department_Id,
        d.Department_Name,
        COUNT(e.Employee_Id) AS Employee_Count
    FROM
        [dbo].[Department] d
    LEFT JOIN
        [dbo].[Employee] e ON d.Department_Id = e.Department_Id
    WHERE
```

```
        d.Department_Id = @DepartmentId
    GROUP BY
        d.Department_Id, d.Department_Name;
END;


--9.Write a stored procedure to calculate and return the total budget allocated to projects
--within a specific department from the `Project` table. Use the department ID as a parameter.
CREATE PROCEDURE [dbo].[GetTotalBudgetByDepartment]
    @DepartmentId INT
AS
BEGIN
    SELECT
        SUM(p.Budget) AS Total_Budget
    FROM
        [dbo].[Project] p
    INNER JOIN
        [dbo].[EmployeeProject] ep ON p.Project_Id = ep.Project_Id
    INNER JOIN
        [dbo].[Employee] e ON ep.Employee_Id = e.Employee_Id
    WHERE
        e.Department_Id = @DepartmentId;
END;


--10.Write a stored procedure to retrieve all employees with their Department and Project
information.
CREATE PROCEDURE [dbo].[GetEmployeeDepartmentProjectInfo]
AS
```

```sql
BEGIN

    SELECT

        e.Employee_Id,

        e.First_Name,

        e.Last_Name,

        e.Email,

        e.Phone_Number,

        e.Hire_Date,

        e.Job_Title,

        e.Salary,

        d.Department_Id,

        d.Department_Name,

        p.Project_Id,

        p.Project_Name,

        p.Start_Date,

        p.End_Date,

        p.Budget

    FROM

        [dbo].[Employee] e

    LEFT JOIN

        [dbo].[Department] d ON e.Department_Id = d.Department_Id

    LEFT JOIN

        [dbo].[EmployeeProject] ep ON e.Employee_Id = ep.Employee_Id

    LEFT JOIN

        [dbo].[Project] p ON ep.Project_Id = p.Project_Id;

END;
```