1. **DDL: Create the above tables with appropriate primary and foreign keys.**

CREATE TABLE Employees (

    EmployeeID INT PRIMARY KEY IDENTITY(100,1),

    FirstName VARCHAR(50) NOT NULL,

    LastName VARCHAR(50) NOT NULL,

    Email VARCHAR(100) UNIQUE,

    DepartmentID INT FOREIGN KEY REFERENCES Departments(DepartmentID),

    HireDate DATE,

    Salary DECIMAL(10,2)

);

1. **DML: Insert at least 5 employees into the Employees table.**

INSERT INTO Employees (FirstName, LastName, Email, DepartmentID, HireDate, Salary)

VALUES

('Ali', 'Khan', 'ali.khan@company.com', 1, '2020-02-10', 65000.00),

('Sara', 'Malik', 'sara.malik@company.com', 2, '2019-05-15', 85000.00),

('Usman', 'Raza', 'usman.raza@company.com', 2, '2021-03-12', 55000.00),

('Ayesha', 'Ahmed', 'ayesha.ahmed@company.com', 3, '2022-07-01', 60000.00),

('Hassan', 'Iqbal', 'hassan.iqbal@company.com', 1, '2018-11-25', 90000.00);

1. **Insert at least 3 departments into the Departments table.**

INSERT INTO Departments (DepartmentName)

VALUES

('Human Resources'),

('IT'),

('Finance');

1. **Insert at least 3 projects into the Projects table.**

INSERT INTO Projects (ProjectName, StartDate, EndDate)

VALUES

('Payroll System', '2024-01-10', '2024-06-30'),

('Website Redesign', '2024-03-01', '2024-09-30'),

('Mobile App Development', '2024-02-15', '2024-08-31');

1. Assign employees to projects by inserting records into EmployeeProjects (at least 7 records).

INSERT INTO EmployeeProjects (EmployeeID, ProjectID, AssignedDate)

VALUES

(100, 500, '2024-02-01'),

(101, 500, '2024-02-10'),

(102, 501, '2024-03-10'),

(103, 501, '2024-04-05'),

(104, 502, '2024-05-12'),

(100, 501, '2024-03-20'),

(102, 502, '2024-06-01');

1. Update the salary of an employee.

UPDATE Employees

SET Salary = 95000.00

WHERE EmployeeID = 105;

1. Delete a project from the Projects table (only if no employees are assigned).

DELETE FROM Projects

WHERE ProjectID NOT IN (SELECT DISTINCT ProjectID FROM EmployeeProjects);

1. Write a query to list all employees along with their department names (use INNER JOIN).

SELECT E.FirstName, E.LastName, D.DepartmentName

FROM Employees E

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

1. List all employees and the projects they are assigned to (use LEFT JOIN to include employees with no projects).

SELECT E.FirstName, E.LastName, P.ProjectName

FROM Employees E

LEFT JOIN EmployeeProjects EP ON E.EmployeeID = EP.EmployeeID

LEFT JOIN Projects P ON EP.ProjectID = P.ProjectID

ORDER BY E.EmployeeID;

1. Find the names of employees who are managers of any department.

SELECT DISTINCT E.FirstName, E.LastName

FROM Employees E

INNER JOIN Departments D ON E.EmployeeID = D.ManagerID;

1. Find employees who are not assigned to any project.

SELECT E.FirstName, E.LastName

FROM Employees E

LEFT JOIN EmployeeProjects EP ON E.EmployeeID = EP.EmployeeID

WHERE EP.ProjectID IS NULL;

1. Calculate the total salary paid in each department.

SELECT D.DepartmentName, SUM(E.Salary) AS TotalSalary

FROM Employees E

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

GROUP BY D.DepartmentName;

1. Find the average salary of all employees.

SELECT AVG(Salary) AS AverageSalary FROM Employees;

1. Find the maximum and minimum salaries in the company.

SELECT MAX(Salary) AS MaxSalary, MIN(Salary) AS MinSalary FROM Employees;

1. Count how many projects each employee is working on.

SELECT E.FirstName, E.LastName, COUNT(EP.ProjectID) AS ProjectCount

FROM Employees E

LEFT JOIN EmployeeProjects EP ON E.EmployeeID = EP.EmployeeID

GROUP BY E.FirstName, E.LastName;

1. Write a query to find the employee with the highest salary.

SELECT TOP 1 FirstName, LastName, Salary

FROM Employees

ORDER BY Salary DESC;

1. Write a query to find projects that have no employees assigned.

SELECT P.ProjectName

FROM Projects P

LEFT JOIN EmployeeProjects EP ON P.ProjectID = EP.ProjectID

WHERE EP.EmployeeID IS NULL;