

```
-- Create Tables
CREATE TABLE Employees (
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50),
    ManagerID INT,
    DepartmentID INT
);

CREATE TABLE Departments (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(50)
);

CREATE TABLE Salaries (
    EmpID INT,
    Salary INT,
    FOREIGN KEY (EmpID) REFERENCES Employees(EmpID)
);

-- Insert Data into Employees
INSERT INTO Employees (EmpID, Name, ManagerID, DepartmentID) VALUES
(1, 'Alice', NULL, 10),
(2, 'Bob', 1, 10),
(3, 'Charlie', 1, 20),
(4, 'David', 2, 10),
(5, 'Eva', 2, 30),
(6, 'Frank', 3, 20);

-- Insert Data into Departments
INSERT INTO Departments (DepartmentID, DepartmentName) VALUES
(10, 'HR'),
(20, 'IT'),
(30, 'Finance');

-- Insert Data into Salaries
INSERT INTO Salaries (EmpID, Salary) VALUES
(1, 90000),
(2, 60000),
(3, 75000),
(4, 50000),
```

```

(5, 65000),
(6, 70000);

-- 1. Display each employee's name along with their manager's name
SELECT e.Name AS Employee, m.Name AS Manager
FROM Employees e
LEFT JOIN Employees m ON e.ManagerID = m.EmpID;

-- 2. Find employees who earn more than their managers
SELECT e.Name AS Employee, s1.Salary AS EmployeeSalary, m.Name AS Manager,
s2.Salary AS ManagerSalary
FROM Employees e
JOIN Salaries s1 ON e.EmpID = s1.EmpID
JOIN Employees m ON e.ManagerID = m.EmpID
JOIN Salaries s2 ON m.EmpID = s2.EmpID
WHERE s1.Salary > s2.Salary;

-- 3. List the names of employees working in the 'IT' department
SELECT e.Name
FROM Employees e
JOIN Departments d ON e.DepartmentID = d.DepartmentID
WHERE d.DepartmentName = 'IT';

-- 4. Display the department name along with the highest salary in that
department
SELECT d.DepartmentName, MAX(s.Salary) AS HighestSalary
FROM Employees e
JOIN Salaries s ON e.EmpID = s.EmpID
JOIN Departments d ON e.DepartmentID = d.DepartmentID
GROUP BY d.DepartmentName;

-- 5. Find employees who have the same salary as at least one other
employee
SELECT e.Name, s.Salary
FROM Employees e
JOIN Salaries s ON e.EmpID = s.EmpID
WHERE s.Salary IN (
    SELECT Salary
    FROM Salaries

```

```

        GROUP BY Salary
        HAVING COUNT(*) > 1
    );

-- 6. Display employees who do not manage anyone
SELECT e.Name
FROM Employees e
WHERE e.EmpID NOT IN (
    SELECT DISTINCT ManagerID
    FROM Employees
    WHERE ManagerID IS NOT NULL
);

-- 7. Display the names of employees whose salary is above the average
salary
SELECT e.Name, s.Salary
FROM Employees e
JOIN Salaries s ON e.EmpID = s.EmpID
WHERE s.Salary > (SELECT AVG(Salary) FROM Salaries);

-- 8. Find employees working in departments where the manager's salary is
greater than 80,000
SELECT e.Name, d.DepartmentName
FROM Employees e
JOIN Departments d ON e.DepartmentID = d.DepartmentID
JOIN Employees m ON e.ManagerID = m.EmpID
JOIN Salaries s ON m.EmpID = s.EmpID
WHERE s.Salary > 80000;

-- 9. Find the names of employees who earn more than the average salary of
their department
SELECT e.Name, s.Salary, d.DepartmentName
FROM Employees e
JOIN Salaries s ON e.EmpID = s.EmpID
JOIN Departments d ON e.DepartmentID = d.DepartmentID
WHERE s.Salary > (
    SELECT AVG(s2.Salary)
    FROM Employees e2
    JOIN Salaries s2 ON e2.EmpID = s2.EmpID
    WHERE e2.DepartmentID = e.DepartmentID

```

```
);

-- 10. List employees whose salary is higher than at least one employee in
the same department
SELECT e.Name, s.Salary, d.DepartmentName
FROM Employees e
JOIN Salaries s ON e.EmpID = s.EmpID
JOIN Departments d ON e.DepartmentID = d.DepartmentID
WHERE s.Salary > ANY (
    SELECT s2.Salary
    FROM Employees e2
    JOIN Salaries s2 ON e2.EmpID = s2.EmpID
    WHERE e2.DepartmentID = e.DepartmentID
    AND e2.EmpID <> e.EmpID
);
```