MySQL Employee & Department SQL Queries Documentation

1. Table Creation & Inserts

```sql
-- Department table
CREATE TABLE Department (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(50) NOT NULL,
    Location VARCHAR(50)
);

INSERT INTO Department (DepartmentID, DepartmentName, Location) VALUES
(10, 'IT', 'New York'),
(20, 'HR', 'Chicago'),
(30, 'Finance', 'San Francisco');

-- Manager table
CREATE TABLE Manager (
    ManagerID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    DepartmentID INT,
    HireDate DATE,
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
);

INSERT INTO Manager (ManagerID, FirstName, LastName, DepartmentID, HireDate)
VALUES
(1, 'Alice', 'Johnson', 10, '2018-05-10'),
(2, 'Bob', 'Smith', 20, '2017-03-21'),
(3, 'Carol', 'White', 30, '2019-09-14');

-- Employee table
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Position VARCHAR(50),
    Salary DECIMAL(10,2),
    ManagerID INT,
    DepartmentID INT,
    HireDate DATE,
    FOREIGN KEY (ManagerID) REFERENCES Employee(EmployeeID),
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
);
```

```sql
INSERT INTO Employee (EmployeeID, FirstName, LastName, Position, Salary,
ManagerID, DepartmentID, HireDate) VALUES
(101, 'David', 'Brown', 'Software Engineer', 75000.00, 1, 10, '2021-06-15'),
(102, 'Eva', 'Green', 'System Analyst', 68000.00, 1, 10, '2022-03-12'),
(103, 'Frank', 'Black', 'Recruiter', 52000.00, 2, 20, '2021-11-05'),
(104, 'Grace', 'Blue', 'Accountant', 60000.00, 3, 30, '2023-01-20'),
(105, 'Henry', 'Gold', 'IT Support', 45000.00, 1, 10, '2022-08-10');
```

---

2. Basic Queries

```sql
-- List all employees
SELECT * FROM Employee;

-- Show only first name, last name, and salary
SELECT FirstName, LastName, Salary FROM Employee;

-- Get the names of all departments
SELECT DepartmentName FROM Department;

-- Find employees whose salary is greater than 70,000
SELECT FirstName, LastName, Salary FROM Employee WHERE Salary > 70000;

-- List employees hired after 2020-01-01
SELECT FirstName, LastName, HireDate FROM Employee WHERE HireDate >
'2020-01-01';

-- Show employees working in the "IT" department
SELECT e.FirstName, e.LastName, d.DepartmentName
FROM Employee e
JOIN Department d ON e.DepartmentID = d.DepartmentID
WHERE d.DepartmentName = 'IT';
```

---

3. Intermediate Queries

```sql
-- Employees with department name
SELECT e.FirstName, e.LastName, d.DepartmentName
FROM Employee e
JOIN Department d ON e.DepartmentID = d.DepartmentID;

-- Employees with their manager's full name
SELECT e.FirstName AS EmployeeFirst, e.LastName AS EmployeeLast,
       m.FirstName AS ManagerFirst, m.LastName AS ManagerLast
```

```sql
FROM Employee e
LEFT JOIN Employee m ON e.ManagerID = m.EmployeeID;

-- Number of employees per department
SELECT d.DepartmentName, COUNT(e.EmployeeID) AS EmployeeCount
FROM Department d
LEFT JOIN Employee e ON d.DepartmentID = e.DepartmentID
GROUP BY d.DepartmentName;

-- Average salary per department
SELECT d.DepartmentName, AVG(e.Salary) AS AvgSalary
FROM Department d
JOIN Employee e ON d.DepartmentID = e.DepartmentID
GROUP BY d.DepartmentName;

-- Employees in Finance earning > 65,000
SELECT e.FirstName, e.LastName, e.Salary
FROM Employee e
JOIN Department d ON e.DepartmentID = d.DepartmentID
WHERE d.DepartmentName = 'Finance' AND e.Salary > 65000;

-- Employees without a manager
SELECT FirstName, LastName FROM Employee WHERE ManagerID IS NULL;

-- Departments with no employees
SELECT d.DepartmentName
FROM Department d
LEFT JOIN Employee e ON d.DepartmentID = e.DepartmentID
WHERE e.EmployeeID IS NULL;

-- Highest salary per department
SELECT d.DepartmentName, MAX(e.Salary) AS MaxSalary
FROM Department d
JOIN Employee e ON d.DepartmentID = e.DepartmentID
GROUP BY d.DepartmentName;
```

---

4. Advanced Queries

```sql
-- Top 3 highest-paid employees per department
SELECT *
FROM (
    SELECT e.EmployeeID, e.FirstName, e.LastName, e.Salary, d.DepartmentName,
           RANK() OVER (PARTITION BY e.DepartmentID ORDER BY e.Salary DESC) AS
SalaryRank
    FROM Employee e
```

```sql
    JOIN Department d ON e.DepartmentID = d.DepartmentID
) ranked
WHERE SalaryRank <= 3;

-- Employees earning more than their manager
SELECT e.FirstName, e.LastName, e.Salary,
       m.FirstName AS ManagerFirst, m.Salary AS ManagerSalary
FROM Employee e
JOIN Employee m ON e.ManagerID = m.EmployeeID
WHERE e.Salary > m.Salary;

-- Recursive hierarchy
WITH RECURSIVE EmployeeHierarchy AS (
    SELECT EmployeeID, FirstName, LastName, ManagerID, 0 AS Level
    FROM Employee
    WHERE ManagerID IS NULL
    UNION ALL
    SELECT e.EmployeeID, e.FirstName, e.LastName, e.ManagerID, eh.Level + 1
    FROM Employee e
    INNER JOIN EmployeeHierarchy eh ON e.ManagerID = eh.EmployeeID
)
SELECT * FROM EmployeeHierarchy
ORDER BY Level, ManagerID;

-- Department with highest average salary
SELECT d.DepartmentName, AVG(e.Salary) AS AvgSalary
FROM Department d
JOIN Employee e ON d.DepartmentID = e.DepartmentID
GROUP BY d.DepartmentName
ORDER BY AvgSalary DESC
LIMIT 1;

-- Rank employees by salary within department
SELECT e.FirstName, e.LastName, d.DepartmentName,
       RANK() OVER (PARTITION BY e.DepartmentID ORDER BY e.Salary DESC) AS
SalaryRank
FROM Employee e
JOIN Department d ON e.DepartmentID = d.DepartmentID;

-- Salary percentage of department total
SELECT e.FirstName, e.LastName, d.DepartmentName, e.Salary,
       ROUND( (e.Salary / SUM(e.Salary) OVER (PARTITION BY e.DepartmentID)) *
100, 2 ) AS SalaryPercentage
FROM Employee e
JOIN Department d ON e.DepartmentID = d.DepartmentID;

-- Employees sorted by department and salary
SELECT e.FirstName, e.LastName, d.DepartmentName, e.Salary
```

```sql
FROM Employee e
JOIN Department d ON e.DepartmentID = d.DepartmentID
ORDER BY d.DepartmentName, e.Salary DESC;

-- Employees with same salary in different departments
SELECT DISTINCT e1.FirstName, e1.LastName, e1.Salary, d1.DepartmentName
FROM Employee e1
JOIN Department d1 ON e1.DepartmentID = d1.DepartmentID
JOIN Employee e2 ON e1.Salary = e2.Salary AND e1.DepartmentID <> e2.DepartmentID
ORDER BY e1.Salary;
```