

## **DATABASE SETUP**

### **Create Tables**

```
CREATE TABLE departments (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(50),
    location VARCHAR(50)
);
```

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(50),
    dept_id INT,
    salary INT,
    hire_date DATE,
    manager_id INT,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);
```

```
CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(50),
    city VARCHAR(50),
    signup_date DATE
);
```

```
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
```

```
customer_id INT,  
order_date DATE,  
order_amount INT,  
FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

### **Insert Sample Data**

```
INSERT INTO departments VALUES
```

```
(1, 'HR', 'Mumbai'),  
(2, 'IT', 'Bangalore'),  
(3, 'Finance', 'Delhi');
```

```
INSERT INTO employees VALUES
```

```
(101, 'Amit', 1, 40000, '2018-05-10', NULL),  
(102, 'Neha', 1, 35000, '2019-03-15', 101),  
(103, 'Raj', 2, 70000, '2017-07-01', NULL),  
(104, 'Sneha', 2, 65000, '2020-08-21', 103),  
(105, 'Karan', 3, 50000, '2016-01-11', NULL),  
(106, 'Pooja', 3, 45000, '2021-09-05', 105);
```

```
INSERT INTO customers VALUES
```

```
(1, 'Rohit', 'Mumbai', '2022-01-10'),  
(2, 'Anjali', 'Delhi', '2021-11-20'),  
(3, 'Suresh', 'Bangalore', '2023-02-15'),  
(4, 'Meena', 'Mumbai', '2022-06-18');
```

```
INSERT INTO orders VALUES
```

```
(201, 1, '2023-01-05', 12000),
```

(202, 1, '2023-03-12', 18000),  
(203, 2, '2022-12-25', 22000),  
(204, 3, '2023-04-10', 8000);

## PROBLEM QUESTIONS

### Scenario 1: Department Cost Evaluation

The finance team is reviewing **operational costs across departments**. They believe some departments may be **costing more per employee** than the company norm.

Instead of looking at individual salaries, management wants to **compare departments as a whole** with the **company's overall salary behavior**.

This insight will help decide whether certain departments require restructuring or optimization.

**Students must figure out:**

- What defines “company norm”
- Whether comparison is row-level or aggregate
- Which clause filters aggregated results

### Scenario 2: Long-Term Employee Risk Analysis

HR has noticed that some employees have stayed with the company for many years but **might be underpaid compared to peers**.

Such employees are considered **high attrition risk**, especially if they are earning **less than others in their own department**.

The HR team wants to proactively identify these cases.

**Students must extract:**

- Department-level salary comparison
- Experience calculation using dates
- Multiple conditions combined

### **Scenario 3: Leadership Effectiveness Review**

Top management believes that a manager's effectiveness reflects in the **financial strength of their team**.

They want to identify managers whose team members earn **more on average than the company's overall employee average**.

This analysis will be used during leadership performance reviews.

#### **Students must deduce:**

- Who qualifies as a manager
- Team-wise aggregation
- Comparison with a global metric

### **Scenario 4: High-Value Customer Identification**

The marketing department is planning a **premium loyalty program**.

Instead of rewarding customers based on single large orders, they want to focus on **consistent high spenders**.

A customer qualifies if their **total spending exceeds the average spending across all customers**.

#### **Students must realize:**

- Order data must be summarized per customer
- A benchmark must be calculated first
- Filtering happens after aggregation

### **Scenario 5: Customer Engagement Gap**

The CRM team wants to understand **where customer engagement is breaking down**.

They found that many users sign up but never actually place an order.

These customers represent **lost conversion opportunities**.

The company wants a list of such users to run targeted campaigns.

#### **Students must infer:**

- Existence vs absence of related records
- Proper join strategy

- NULL-based filtering