
0000000 000000000 – Company Staffing System

A company wants a simple staffing database. There are **departments** (with an ID, name, location) and **staff** (with an ID, name, age, salary). Some staff may **not** belong to any department yet, and some departments may currently have **no** staff. Your task (as learners) after setup: write queries to explore the data, perform joins, and analyze edge cases where results contain **NULL** values.

000 – Create DB, Tables, and Insert Seed Data

Copy-paste into MySQL. This guarantees:

- At least two **departments without staff**
- At least two **staff without a department (NULL dept_id)**
- Multiple staff in the same department (for grouping/joins)

```
-- Create and use database
CREATE DATABASE IF NOT EXISTS company_db;
USE company_db;

-- Clean re-run
DROP TABLE IF EXISTS staff;
DROP TABLE IF EXISTS departments;

-- Departments master
CREATE TABLE departments (
    dept_id    INT PRIMARY KEY AUTO_INCREMENT,
    dept_name  VARCHAR(50) NOT NULL UNIQUE,
    location   VARCHAR(50) NOT NULL
);

-- Staff table (dept is intentionally optional)
CREATE TABLE staff (
    staff_id   INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50) NOT NULL,
    last_name  VARCHAR(50) NOT NULL,
    age        INT,
    salary     DECIMAL(10,2),
    dept_id    INT NULL,
    CONSTRAINT fk_staff_dept
        FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);

-- Seed Departments
-- (Finance and R&D will have NO staff to demonstrate RIGHT/FULL JOIN behavior)
INSERT INTO departments (dept_id, dept_name, location) VALUES
(1, 'IT',          'Bangalore'),
(2, 'HR',          'Hyderabad'),
(3, 'Finance',     'Mumbai'),      -- no staff
(4, 'Marketing',   'Delhi'),
(5, 'Operations',  'Chennai');
```

```

(6, 'R&D',      'Pune');      -- no staff

-- Seed Staff
-- (Some rows have NULL dept_id to demonstrate LEFT/FULL JOIN behavior)
INSERT INTO staff (staff_id, first_name, last_name, age, salary, dept_id) VALUES
(101, 'Amit',    'Verma',      28, 55000.00, 1),    -- IT
(102, 'Sneha',   'Reddy',      32, 60000.00, 2),    -- HR
(103, 'Ravi',    'Sharma',     26, 48000.00, NULL), -- no department
(104, 'Pooja',   'Iyer',       29, 52000.00, 4),    -- Marketing
(105, 'Arjun',   'Mehta',      35, 75000.00, 1),    -- IT
(106, 'Divya',   'Nair',       30, 50000.00, 5),    -- Operations
(107, 'Rahul',   'Kapoor',     41, 91000.00, 1),    -- IT
(108, 'Priya',   'Singh',      24, 42000.00, NULL), -- no department
(109, 'Vikram',  'Rao',       37, 68000.00, 4),    -- Marketing
(110, 'Neha',    'Kulkarni',   33, 58500.00, 2);    -- HR

```

▮ Exercises – Company Staffing System

Part A – Basic Staff Queries

1. Show all staff details.
 2. List first name, last name, and salary of staff earning more than 60,000.
 3. Find staff who do not belong to any department.
 4. Display all staff sorted by age (youngest first).
 5. Count the total number of staff.
-

Part B – Basic Department Queries

6. Show all department details.
 7. List all departments located in Bangalore or Chennai.
 8. Find department names starting with the letter "M".
 9. Count how many unique locations exist.
 10. List all departments sorted alphabetically.
-

Part C – INNER JOIN

11. List staff names with their department names.
 12. Show staff working in "IT" along with their salary.
-

Part D – LEFT JOIN

14. Show all staff with their department (if missing, display NULL).
 15. Find staff who do not belong to any department.
-

Part E – RIGHT JOIN

17. Show all departments with staff (if missing, display NULL).
 18. Find departments that do not have any staff.
 19. List all departments with their staff, but keep departments without staff visible.
-