# SQL JOINs – Simple Practice Questions with Answers

This document contains 10 simple and beginner-friendly SQL JOIN questions. These examples use two tables: employees and departments.

## Sample Tables & Data

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

CREATE TABLE employees (
   emp_id INT PRIMARY KEY,
   emp_name VARCHAR(50),
   dept_id INT,
   salary INT
);

INSERT INTO departments VALUES
(1, 'IT'),
(2, 'HR'),
(3, 'Finance');

INSERT INTO employees VALUES
(101, 'Amit', 1, 80000),
(102, 'Neha', 2, 60000),
(103, 'Rahul', 3, 90000),
(104, 'Sara', 1, 75000),
(105, 'John', NULL, 50000);
```

## JOIN Practice Questions

1. 1. What is an INNER JOIN?

Explanation: INNER JOIN returns only matching rows from both tables.
Answer:

```
SELECT e.emp_name, d.dept_name FROM employees e INNER JOIN departments d ON e.dept_id = d.dept_id;
```

2. 2. List employee names with their department names.

Explanation: Match employees with departments using dept_id.
Answer:

```
SELECT e.emp_name, d.dept_name FROM employees e JOIN departments d ON e.dept_id =
d.dept_id;
```

3. 3. Display all employees even if they don’t belong to a department.

Explanation: LEFT JOIN keeps all rows from employees table.
Answer:

```
SELECT e.emp_name, d.dept_name FROM employees e LEFT JOIN departments d ON
e.dept_id = d.dept_id;
```

4. 4. Display all departments even if they have no employees.

Explanation: RIGHT JOIN keeps all rows from departments table.
Answer:

```
SELECT e.emp_name, d.dept_name FROM employees e RIGHT JOIN departments d ON
e.dept_id = d.dept_id;
```

5. 5. Find employees who are not assigned to any department.

Explanation: Look for NULLs after LEFT JOIN.
Answer:

```
SELECT e.emp_name FROM employees e LEFT JOIN departments d ON e.dept_id = d.dept_id
WHERE d.dept_id IS NULL;
```

6. 6. Count number of employees in each department.

Explanation: JOIN then GROUP BY department.
Answer:

```
SELECT d.dept_name, COUNT(e.emp_id) FROM departments d LEFT JOIN employees e ON
d.dept_id = e.dept_id GROUP BY d.dept_name;
```

7. 7. Find average salary per department.

Explanation: JOIN employees with departments and apply AVG.
Answer:

```
SELECT d.dept_name, AVG(e.salary) FROM employees e JOIN departments d ON e.dept_id =
d.dept_id GROUP BY d.dept_name;
```

8. 8. Show only departments having more than one employee.

Explanation: Use HAVING after GROUP BY.
Answer:

```
SELECT d.dept_name, COUNT(e.emp_id) FROM departments d JOIN employees e ON
d.dept_id = e.dept_id GROUP BY d.dept_name HAVING COUNT(e.emp_id) > 1;
```

9. 9. Display employee name, department name, and salary.

Explanation: Simple INNER JOIN to fetch related data.
Answer:
SELECT e.emp_name, d.dept_name, e.salary FROM employees e JOIN departments d ON e.dept_id = d.dept_id;

10. 10. Difference between INNER JOIN and LEFT JOIN.

Explanation:
INNER JOIN → returns only matching rows.
LEFT JOIN → returns all rows from left table and matching rows from right.
Answer: Conceptual question (no SQL query).