Day – 28 : Joins in SQL

Types of SQL Joins

1. **INNER JOIN**: Retrieves records with matching values in both tables.
2. **LEFT JOIN (LEFT OUTER JOIN)**: Retrieves all records from the left table and matching records from the right table. Non-matching records in the right table result in NULL.
3. **RIGHT JOIN (RIGHT OUTER JOIN)**: Retrieves all records from the right table and matching records from the left table. Non-matching records in the left table result in NULL.
4. **FULL JOIN (FULL OUTER JOIN)**: Retrieves all records from both tables, with NULL for non-matching rows in either table.
5. **CROSS JOIN**: Combines all rows from both tables, resulting in a Cartesian product.
6. **SELF JOIN**: Joins a table to itself.

To explain joins, we’ll create two tables: **Employees3** and **Departments**.

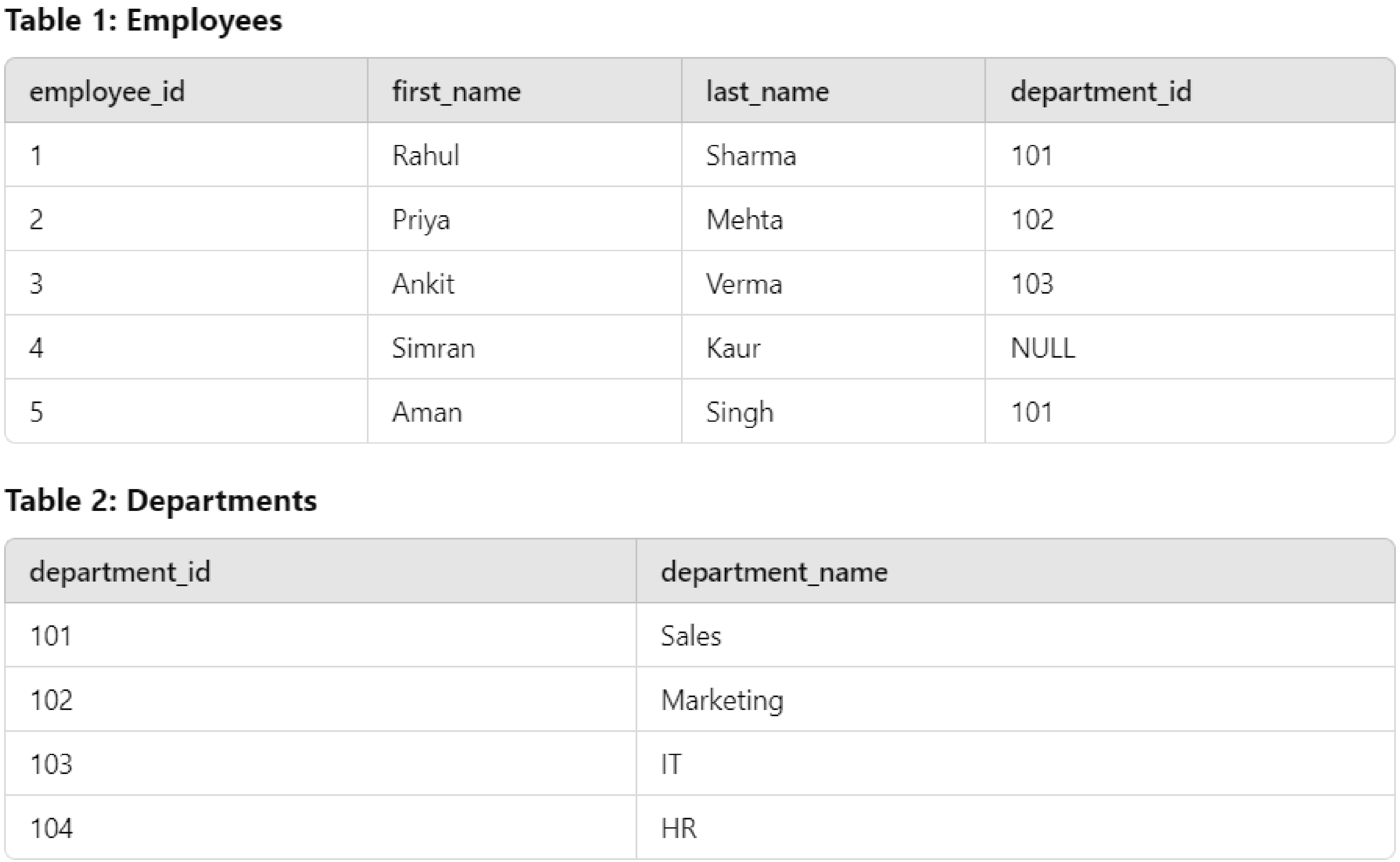


Table 1: Employees

employee_id	first_name	last_name	department_id
1	Rahul	Sharma	101
2	Priya	Mehta	102
3	Ankit	Verma	103
4	Simran	Kaur	NULL
5	Aman	Singh	101

Table 2: Departments

department_id	department_name
101	Sales
102	Marketing
103	IT
104	HR

```sql
-- Create Employees3 Table
CREATE TABLE Employees3 (
    employee_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    department_id INT
);

-- Insert Data into Employees3
INSERT INTO Employees3 (first_name, last_name, department_id)
VALUES
('Rahul', 'Sharma', 101),
('Priya', 'Mehta', 102),
('Ankit', 'Verma', 103),
('Simran', 'Kaur', NULL),
('Aman', 'Singh', 101);
```

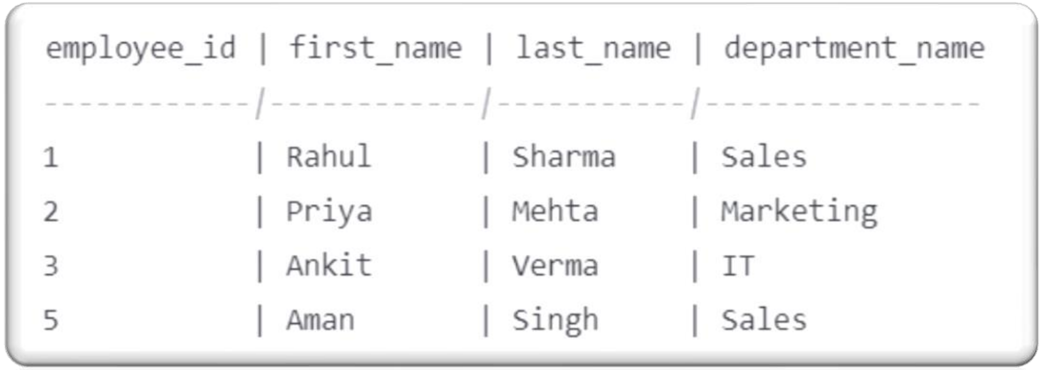
```
-- Create Departments Table
CREATE TABLE Departments (
   department_id INT PRIMARY KEY,
   department_name VARCHAR(50)
);

-- Insert Data into Departments
INSERT INTO Departments (department_id, department_name)
VALUES
(101, 'Sales'),
(102, 'Marketing'),
(103, 'IT'),
(104, 'HR');
```

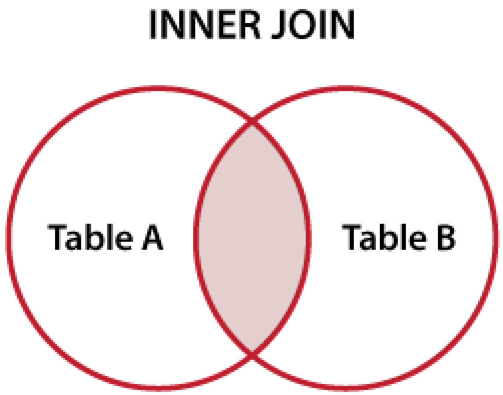
1. INNER JOIN - Retrieve Employees3 and their department names where a match exists.

```
SELECT
  e.employee_id,
  e.first_name,
  e.last_name,
  d.department_name
FROM
  Employees3 e
INNER JOIN
  Departments d
ON
  e.department_id = d.department_id;
```

OUTPUT →



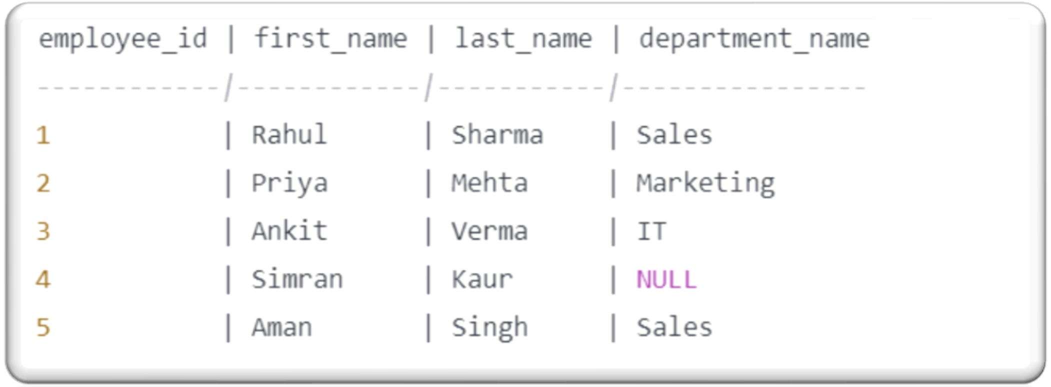
```
employee_id | first_name | last_name | department_name
------------|------------|-----------|----------------
1           | Rahul      | Sharma    | Sales
2           | Priya      | Mehta     | Marketing
3           | Ankit      | Verma     | IT
5           | Aman       | Singh     | Sales
```



2. LEFT JOIN - Retrieve all Employees3 and their department names, including those without a department.

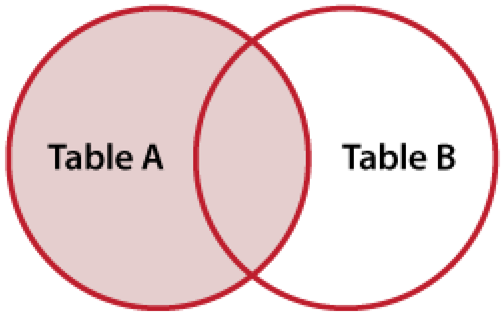
```
SELECT
   e.employee_id,
   e.first_name,
   e.last_name,
   d.department_name
FROM
   Employees3 e
LEFT JOIN
   Departments d
ON
   e.department_id = d.department_id;
```

OUTPUT →



```
employee_id | first_name | last_name | department_name
------------|------------|-----------|----------------
1           | Rahul      | Sharma    | Sales
2           | Priya      | Mehta     | Marketing
3           | Ankit      | Verma     | IT
4           | Simran     | Kaur      | NULL
5           | Aman       | Singh     | Sales
```

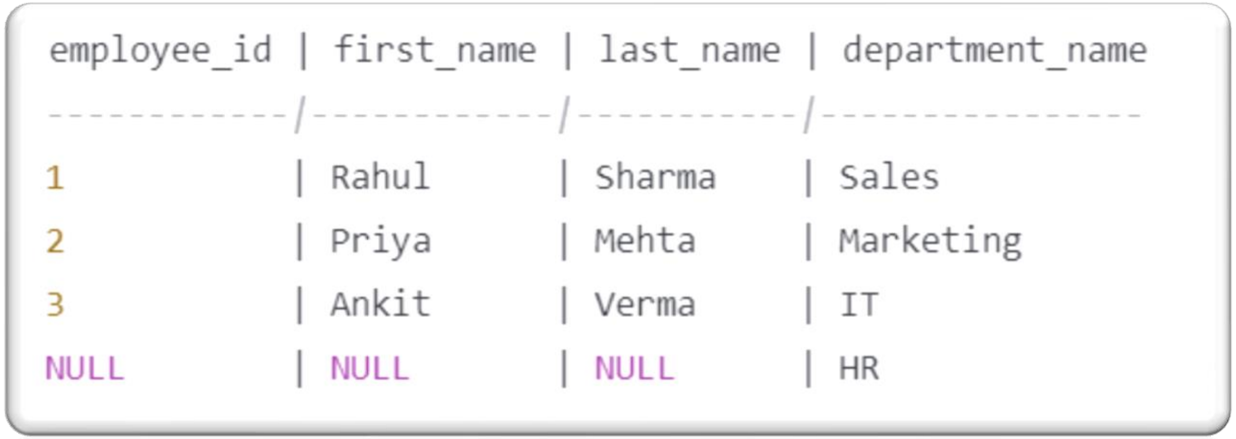
LEFT OUTER JOIN



3. RIGHT JOIN - Retrieve all departments and the Employees3 working in them, including departments without

```
SELECT
    e.employee_id,
    e.first_name,
    e.last_name,
    d.department_name
FROM
    Employees3 e
RIGHT JOIN
    Departments d
ON
    e.department_id = d.department_id;
```

OUTPUT →



```
employee_id | first_name | last_name | department_name
------------|------------|-----------|----------------
1           | Rahul      | Sharma    | Sales
2           | Priya      | Mehta     | Marketing
3           | Ankit      | Verma     | IT
NULL        | NULL       | NULL      | HR
```

RIGHT OUTER JOIN

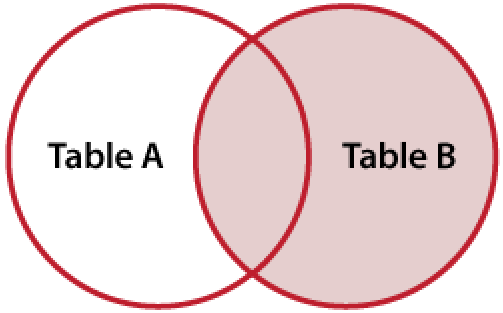


Table A
Table B