**📘 MySQL – JOINS (Full Notes Book Format)**

**🧾 Step 1: Create Sample Tables**

CREATE TABLE students (

student\_id INT,

name VARCHAR(50),

course\_id INT

);

CREATE TABLE courses (

course\_id INT,

course\_name VARCHAR(50)

);

**🧾 Step 2: Insert Sample Data**

INSERT INTO students (student\_id, name, course\_id) VALUES

(1, 'Ankit', 101),

(2, 'Riya', 102),

(3, 'Aman', 101),

(4, 'Sneha', 103),

(5, 'Rohit', NULL);

INSERT INTO courses (course\_id, course\_name) VALUES

(101, 'Python'),

(102, 'Java'),

(103, 'SQL'),

(104, 'C++');

**📊 Data in students Table:**

| **student\_id** | **name** | **course\_id** |
| --- | --- | --- |
| 1 | Ankit | 101 |
| 2 | Riya | 102 |
| 3 | Aman | 101 |
| 4 | Sneha | 103 |
| 5 | Rohit | NULL |

**📊 Data in courses Table:**

| **course\_id** | **course\_name** |
| --- | --- |
| 101 | Python |
| 102 | Java |
| 103 | SQL |
| 104 | C++ |

----------------------------------------------->

**✅ 1. INNER JOIN**

SELECT students.name, courses.course\_name

FROM students

INNER JOIN courses ON students.course\_id = courses.course\_id;

👉 Sirf wo rows milengi jinke dono tables me matching course\_id ho.  
👉 Rohit (NULL) aur course\_id 104 (unused) exclude ho jaayenge.

📤 Output:

| **name** | **course\_name** |
| --- | --- |
| Ankit | Python |
| Riya | Java |
| Aman | Python |
| Sneha | SQL |

----------------------------------------------->

**✅ 2. LEFT JOIN**

SELECT students.name, courses.course\_name

FROM students

LEFT JOIN courses ON students.course\_id = courses.course\_id;

👉 Sabhi students milenge, chahe unka course match ho ya na ho.  
👉 Rohit bhi dikhai dega, jiska course\_id NULL hai.

📤 Output:

| **name** | **course\_name** |
| --- | --- |
| Ankit | Python |
| Riya | Java |
| Aman | Python |
| Sneha | SQL |
| Rohit | NULL |

----------------------------------------------->

**✅ 3. RIGHT JOIN**

SELECT students.name, courses.course\_name

FROM students

RIGHT JOIN courses ON students.course\_id = courses.course\_id;

👉 Sabhi courses dikhte hain, chahe student assigned ho ya nahi.  
👉 C++ bhi dikhai dega, jisme koi student nahi hai.

📤 Output:

| **name** | **course\_name** |
| --- | --- |
| Ankit | Python |
| Riya | Java |
| Aman | Python |
| Sneha | SQL |
| NULL | C++ |

----------------------------------------------->

**✅ 4. FULL OUTER JOIN (MySQL me directly nahi hota — workaround se hota hai)**

SELECT s.name, c.course\_name

FROM students s

LEFT JOIN courses c ON s.course\_id = c.course\_id

UNION

SELECT s.name, c.course\_name

FROM students s

RIGHT JOIN courses c ON s.course\_id = c.course\_id;

👉 Dono tables ke sabhi rows milte hain, chahe match ho ya na ho.  
👉 Rohit aur C++ dono included honge.

📤 Output:

| **name** | **course\_name** |
| --- | --- |
| Ankit | Python |
| Riya | Java |
| Aman | Python |
| Sneha | SQL |
| Rohit | NULL |
| NULL | C++ |

----------------------------------------------->

**✅ 5. SELF JOIN (Table joins with itself)**

SELECT A.name AS Student1, B.name AS Student2

FROM students A, students B

WHERE A.course\_id = B.course\_id AND A.student\_id <> B.student\_id;

👉 Ek hi table ke students compare ho rahe hain jo same course me hain.  
👉 Same course me multiple students ho to unko pair bana ke dikhata hai.

📤 Output:

| **Student1** | **Student2** |
| --- | --- |
| Ankit | Aman |
| Aman | Ankit |

----------------------------------------------->

**✅ Summary Table – Types of JOINS**

| **JOIN Type** | **Description** |
| --- | --- |
| INNER JOIN | Only matching rows from both tables |
| LEFT JOIN | All from left + matching from right |
| RIGHT JOIN | All from right + matching from left |
| FULL OUTER JOIN | All rows from both (via UNION in MySQL) |
| SELF JOIN | Table joins with itself |