

BIT 5th Sem (Professional Class)

QA Training – Day 8

Facilitator: Neha Rouniyar

SQL JOINS: Link Your Data

In today's apps, data is often split into many tables. SQL JOINS help you bring this data together. They show you the complete picture.

For QA testers, knowing how to use JOINS is very important for good testing. It helps you check that all data is correct and matches up.

Why Testers Need SQL JOINS



Check Data in Tables

Make sure linked data in different tables is correct and matches up.



Check Connections

See if data, like users and their orders, is properly connected. No missing pieces.



Find Errors

Spot if data is wrong or missing. This helps find bugs.

JOIN TYPES

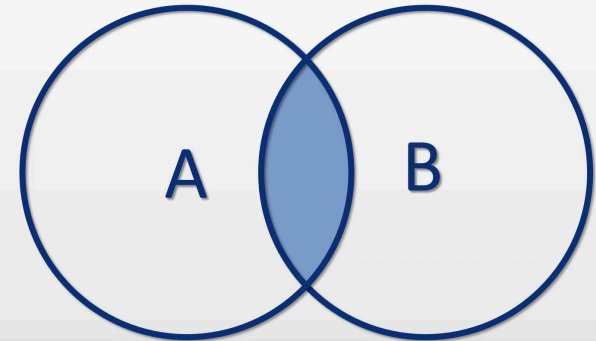
INNER JOIN: Find Matching Data

The **INNER JOIN** is a common type of join. It shows only the records that match in **both** tables. It finds what is shared between two sets of data.

As a QA, you use this to check data that must exist in both tables.

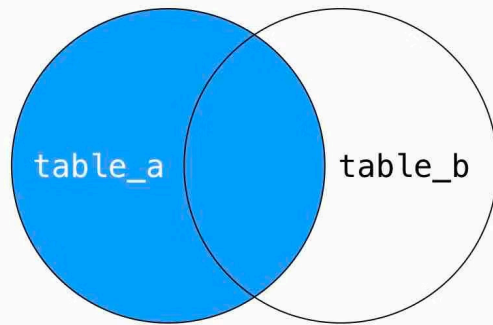
```
SELECT users.name, orders.order_id FROM users INNER JOIN orders  
ON users.user_id = orders.user_id;
```

INNER JOIN



LEFT JOIN: All from One List, Matches from Another

LEFT JOIN



A **LEFT JOIN** (also called LEFT OUTER JOIN) is used when you want to see everything from your main list. It gives you:

- All items from the **left list**.
- Any matching items from the right list.
- If there's no match for an item from the left list, it shows "nothing" (NULL) for the right list's details.

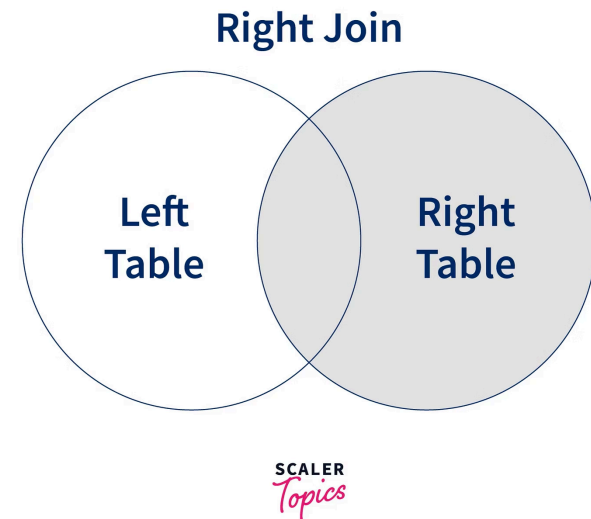
```
SELECT users.name, orders.order_id FROM users LEFT JOIN orders ON  
users.user_id = orders.user_id;
```

RIGHT JOIN: Focusing on the "Other" Side

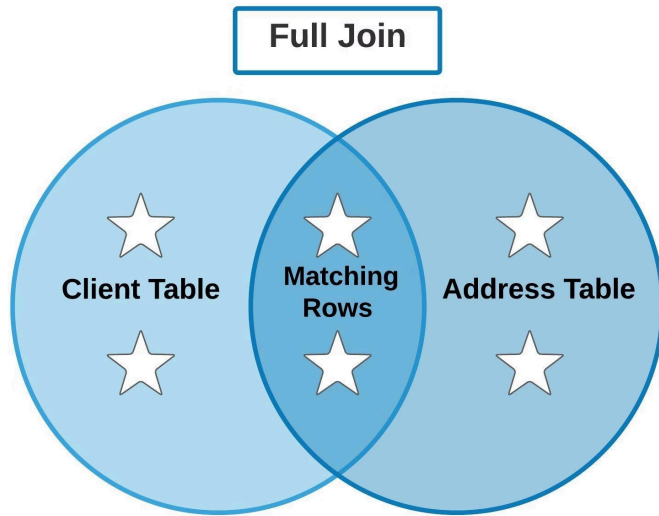
A **RIGHT JOIN** is like a LEFT JOIN but in reverse. It shows you:

- All records from the **right table**.
- Any matching records from the left table.
- Empty spots (NULL) in the left table's columns if there's no match for a right table record.

```
SELECT users.name, orders.order_id FROM users RIGHT JOIN  
orders ON users.user_id = orders.user_id;
```



FULL JOIN: All Data Together



A **FULL JOIN** (or FULL OUTER JOIN) brings together all data from both tables. It shows:

- All records from both the left and right tables.
- Where records match in both tables.
- Empty spots (NULL) if a record has no match in the other table.

📄 ⚠️ Note: SQLite usually doesn't have FULL JOIN. You can often do the same thing by using LEFT JOIN, RIGHT JOIN (or a LEFT JOIN with tables swapped), and combining them.

SQL JOINS: Quick Guide

INNER JOIN

Check matching data. It finds rows that are in both tables.

LEFT JOIN

Find missing data. It shows all rows from the first table, and only matching rows from the second.

RIGHT JOIN

Check data in the second table. It shows all rows from the second table, and only matching rows from the first. (Often done with LEFT JOIN.)

FULL JOIN

Check all data. It shows all rows from both tables, whether they match or not. (Often done with UNION.)



Let's Practice  

What is an API?

API = Messenger Between Apps

An API (Application Programming Interface) acts as a digital messenger, allowing different software applications to communicate and share information with each other.

Example: Weather App Gets Data

Think of a weather app on your phone. It uses an API to ask a weather server for the latest forecast, and the server sends the data back to your app.

Why Learn APIs?



APIs are everywhere

They power almost every app and service you use daily.



Test without full app

You can test parts of a system before the entire application is ready.



Helps with automation later

Understanding APIs is key to building robust automated tests and workflows.

What We Check

1

Status codes

Such as 200=OK, 404=Not found

2

Response time

How fast the system responds

3

Response data

Is the data returned correct?

4

Error handling

How the system reacts to wrong input

API Request Types

GET

(read data)

POST

(send data)

PUT

(update data)

DELETE

(remove data)

Why Test APIs?



Works Correctly

Ensure the core logic functions as expected.



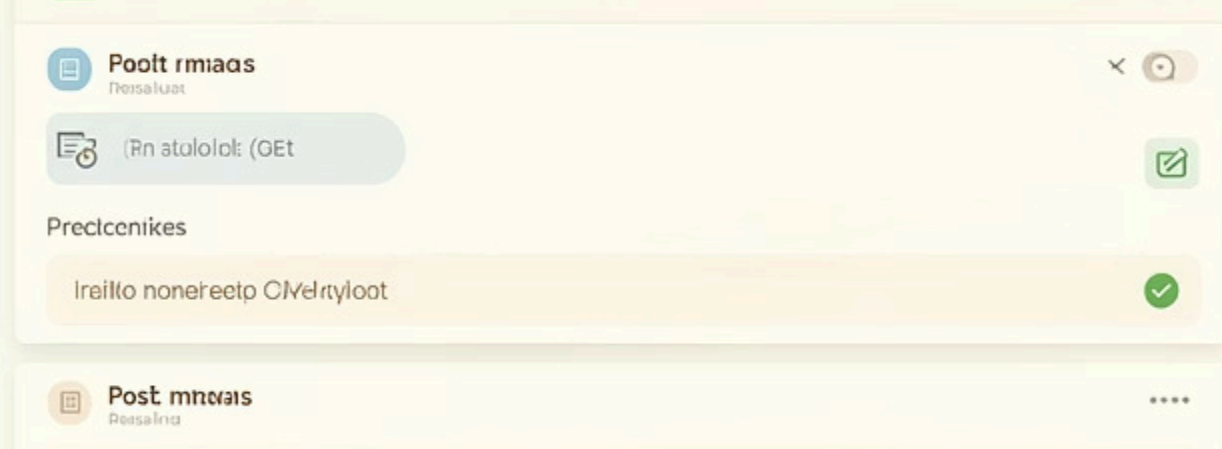
Handles Errors Gracefully

Verify how the API responds to invalid inputs or unexpected scenarios.



Enables Fast Testing

Test backend logic quickly without waiting for UI development or complex interactions.



Next Class Practice:

Sending GET and POST requests in Postman

Checking response status and data

Questions & Discussions ??