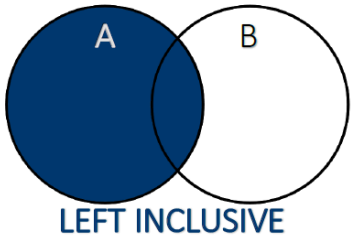
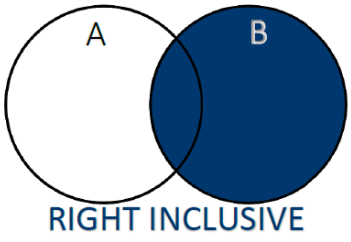
JOINS



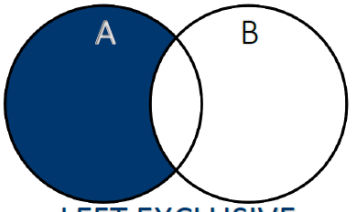
LEFT INCLUSIVE

SQL JOINS

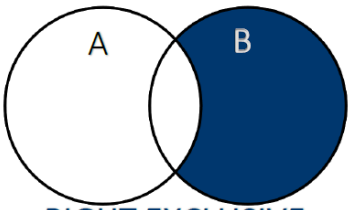
LEFT INCLUSIVE	RIGHT INCLUSIVE
SELECT *[Select List]* FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key	SELECT *[Select List]* FROM TableA A RIGHT OUTER JOIN TableB B ON A.Key= B.Key
LEFT EXCLUSIVE	**RIGHT EXCLUSIVE**
SELECT *[Select List]* FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key WHERE B.Key IS NULL	SELECT *[Select List]* FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key WHERE A.Key IS NULL
FULL OUTER INCLUSIVE	**FULL OUTER EXCLUSIVE**
SELECT *[Select List]* FROM TableA A FULL OUTER JOIN TableB B ON A.Key = B.Key	SELECT *[Select List]* FROM TableA A FULL OUTER JOIN TableB B ON A.Key = B.Key WHERE A.Key IS NULL OR B.Key IS NULL
INNER JOIN	
SELECT *[Select List]* FROM TableA A INNER JOIN TableB B ON A.Key = B.Key	



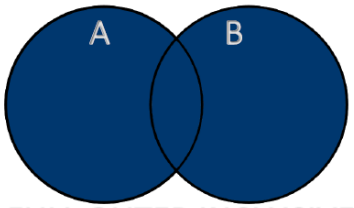
RIGHT INCLUSIVE



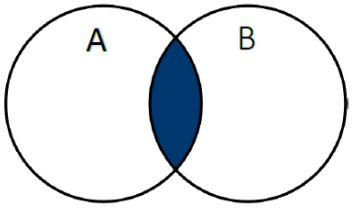
LEFT EXCLUSIVE



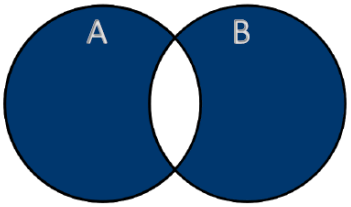
RIGHT EXCLUSIVE



FULL OUTER INCLUSIVE



INNER JOIN



FULL OUTER EXCLUSIVE

What is a JOIN?

A **JOIN** is used to retrieve data from **two or more tables** based on a related column between them. Typically, this related column is a **foreign key**.

JOINs help you answer questions like:

- Which customers placed which orders?
- What is each employee's department?
- Who are the students that didn't submit assignments?

Example Tables

customers

customer_id	name	city
1	Isha	Mumbai
2	Aditi	Delhi
3	Janhavi	Bangalore

orders

order_id	customer_id	product	amount
101	1	Laptop	55000
102	1	Headphones	3000
103	2	Smartphone	20000

Types of JOINs

---

1. INNER JOIN

Definition: Returns only rows with matching values in both tables.

Use when: You want records that exist in both tables.

Syntax:

```
SELECT customers.name, orders.product
FROM customers
INNER JOIN orders
ON customers.customer_id = orders.customer_id;
```

---

2. LEFT JOIN (or LEFT OUTER JOIN)

Definition: Returns all rows from the left table and matched rows from the right. If there is no match, NULL is returned for the right table.

Use when: You want all customers, even those with no orders.

Syntax:

```
SELECT customers.name, orders.product
FROM customers
LEFT JOIN orders
ON customers.customer_id = orders.customer_id;
```

---

3. RIGHT JOIN (or RIGHT OUTER JOIN)

Definition: Returns all rows from the right table and matched rows from the left.

Use when: You want all orders, even if some customers are missing.

Syntax:

```
SELECT customers.name, orders.product
FROM customers
RIGHT JOIN orders
ON customers.customer_id = orders.customer_id;
```

---

4. FULL OUTER JOIN

Definition: Returns all rows from both tables. Unmatched rows get NULLs.

Use when: You want everything from both tables, matched or not.

Syntax:

```
SELECT customers.name, orders.product
FROM customers
FULL OUTER JOIN orders
ON customers.customer_id = orders.customer_id;
```

---

5. SELF JOIN

Definition: A table joined with itself.

Use when: You want to compare rows within the same table, such as employees and their managers.

Example:

```
SELECT A.name AS employee, B.name AS manager
FROM employees A
JOIN employees B
ON A.manager_id = B.emp_id;
```

---

6. CROSS JOIN

Definition: Returns all possible combinations of rows from both tables (Cartesian product).

Use carefully: Results grow rapidly if tables are large.

Syntax:

```
SELECT customers.name, orders.product
FROM customers
CROSS JOIN orders;
```

JOIN Variations and Tips

- You can join more than two tables in a single query.
- You can use `JOIN ... USING(column_name)` if both tables have a column with the same name.
- Always specify meaningful `ON` conditions to avoid incorrect results.
- Use aliases (`AS`) to shorten table names and improve readability.

Real-Life Scenarios

Scenario	JOIN Type
List customers with orders	INNER JOIN
List all customers with or without orders	LEFT JOIN
List all orders even if customer info is missing	RIGHT JOIN
Show all customers and all orders	FULL OUTER JOIN
Find employee-manager relationship	SELF JOIN
Generate all possible filter combinations	CROSS JOIN

Common Questions

1. What is the difference between INNER and LEFT JOIN?

- INNER JOIN returns only matched rows.
- LEFT JOIN returns all rows from the left table, with NULLs if there's no match.

2. Can we JOIN more than two tables?
Yes.

Example:

FROM A
JOIN B ON ...
JOIN C ON ...

3. When would you use a FULL OUTER JOIN?
When you want to get everything from both tables — matched or unmatched.

4. How do JOINs affect performance?
JOINs can be heavy on large datasets. Use indexing on join columns for better performance.

5. What is a Cartesian product?
It's the result of a CROSS JOIN — all possible combinations from both tables.

Quick Reference: JOIN Summary

JOIN Type	Description
INNER JOIN	Only matched rows from both tables
LEFT JOIN	All left table rows + matched right table rows
RIGHT JOIN	All right table rows + matched left table rows
FULL OUTER JOIN	All rows from both tables, matched and unmatched
SELF JOIN	A table joined with itself
CROSS JOIN	All combinations of both tables

Practice Query Example

```
-- List all customers and their total spending
SELECT customers.name, SUM(orders.amount) AS total_spent
FROM customers
LEFT JOIN orders ON customers.customer_id = orders.customer_id
GROUP BY customers.name
ORDER BY total_spent DESC;
```