# SQL JOIN

The JOIN clause in SQL is used to **combine rows from several tables based on a related column between these tables**.

# SQL's 4 JOIN Types

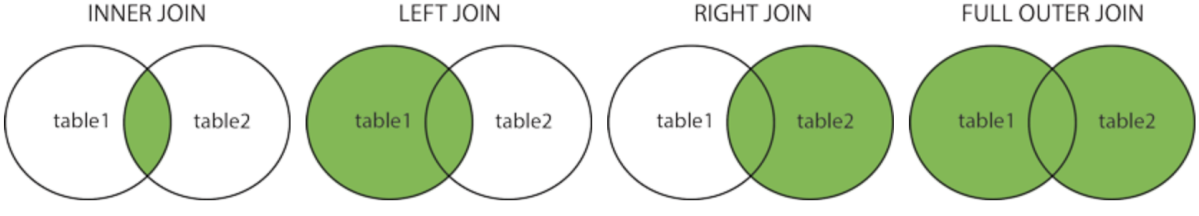
SQL JOIN types include:

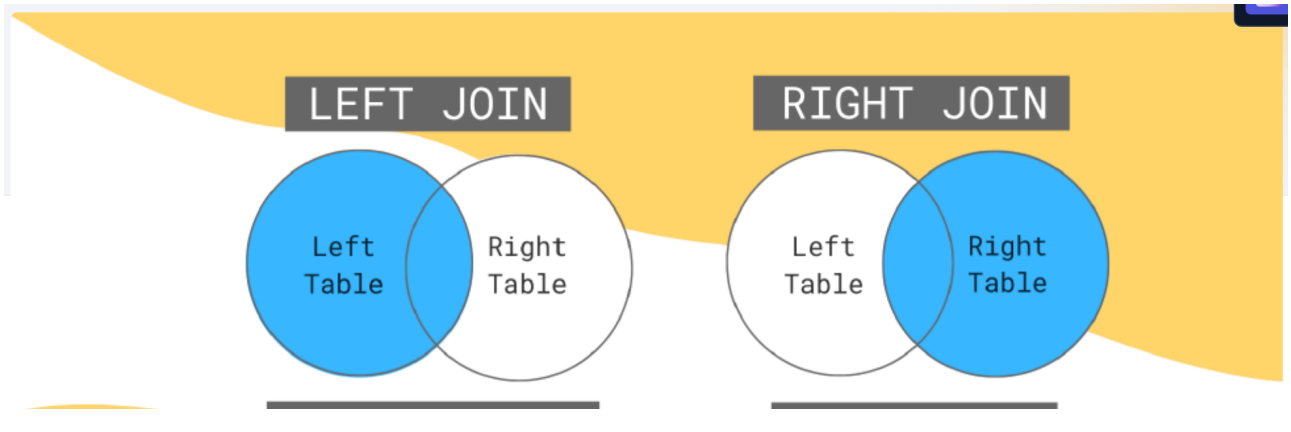
- INNER JOIN (also known as a 'simple' JOIN). This is the most common type of JOIN.
- LEFT JOIN (or LEFT OUTER JOIN)
- RIGHT JOIN (or RIGHT OUTER JOIN)
- FULL JOIN (or FULL OUTER JOIN)

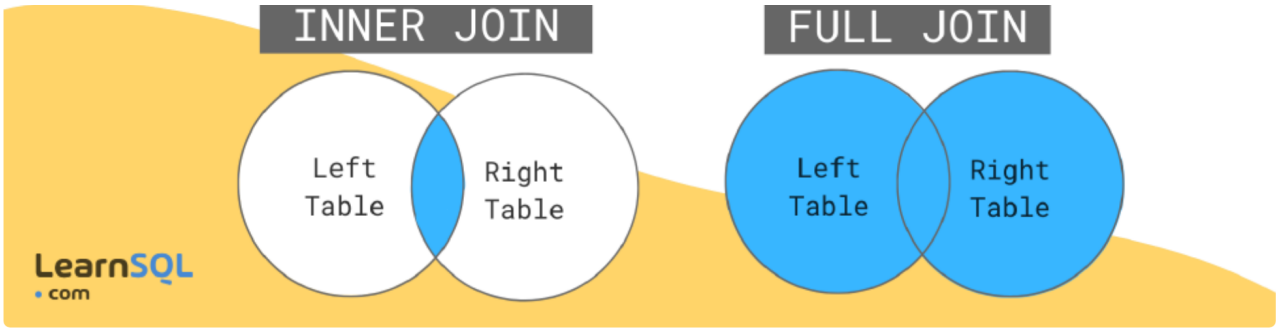
## Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

- `(INNER) JOIN`: Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN`: Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN`: Returns all records when there is a match in either left or right table






# INNER JOIN

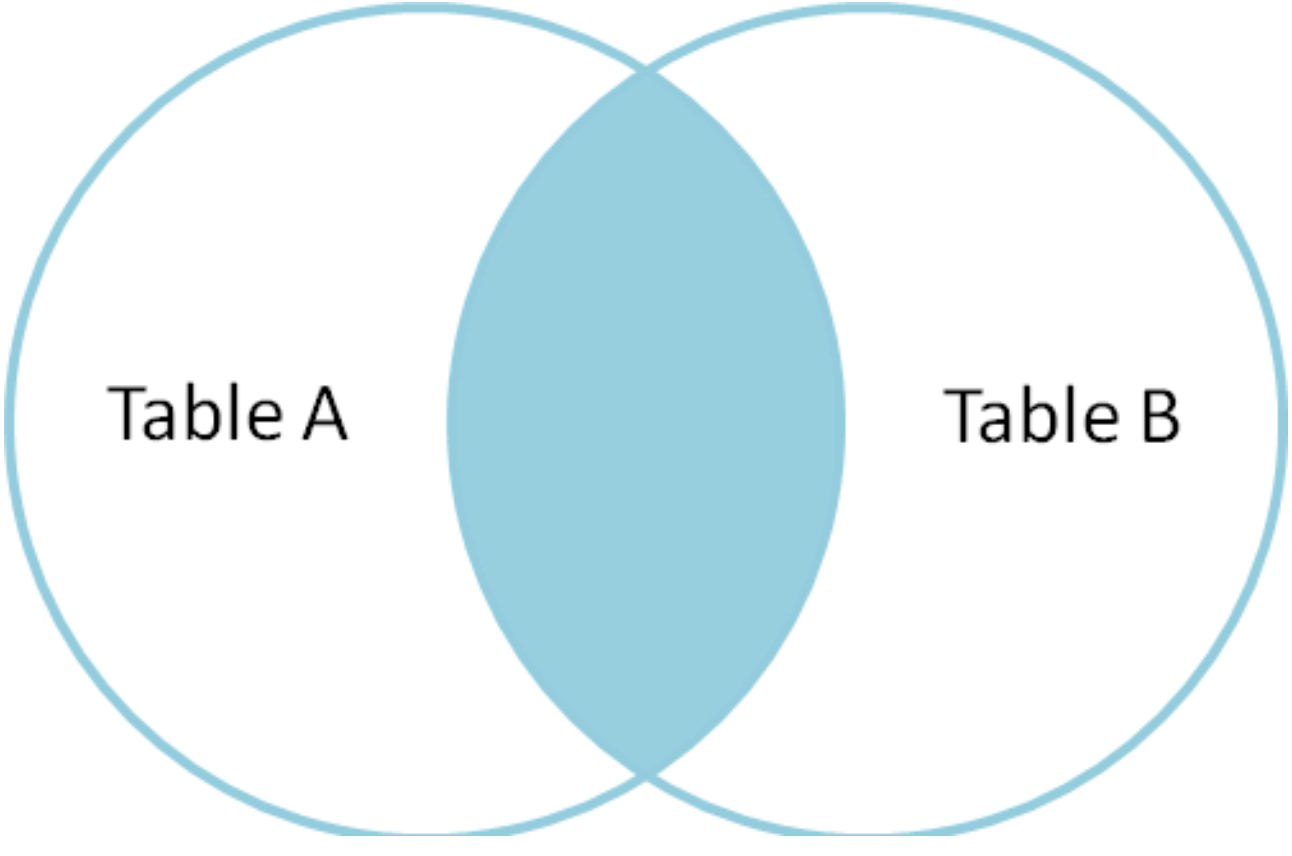
INNER JOIN is used to display matching records from both tables. This is also called a simple JOIN

➡️Returns only matching rows from both tables.🤝

➡️Excludes non-matching rows. ❗

➡️Useful for retrieving only common data from both tables.

➡️Typically results in a smaller result set compared to other joins.



The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.

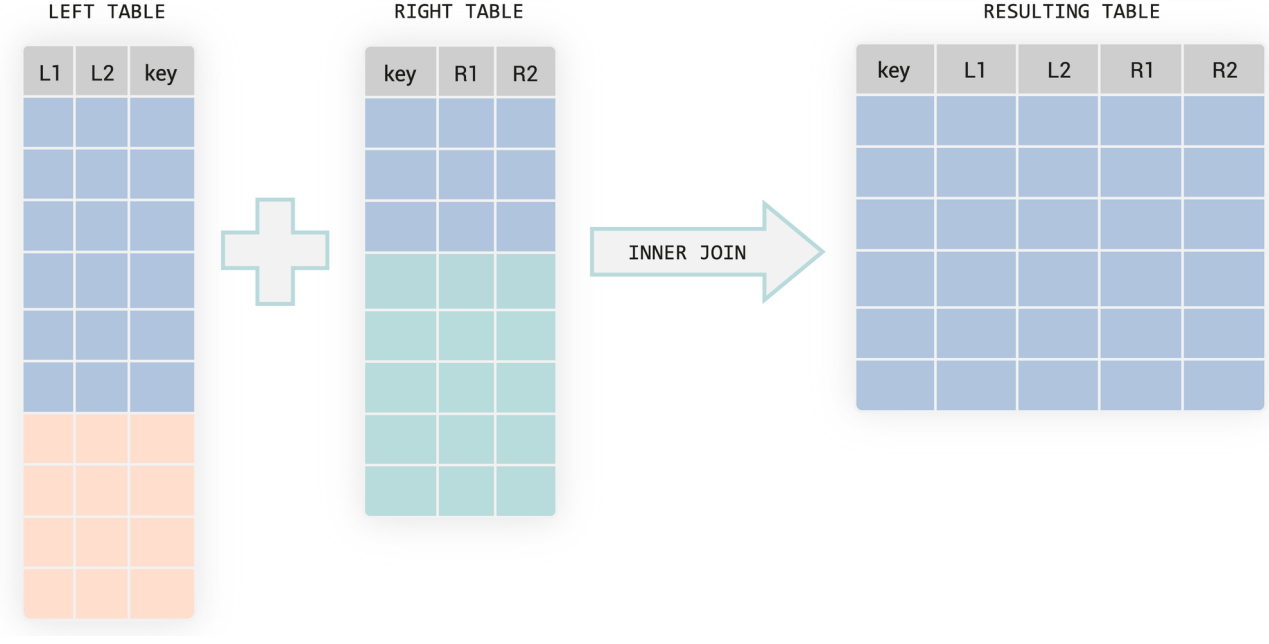
**Syntax**:

```
SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

INNER JOIN table2

ON table1.matching_column = table2.matching_column;
```

**table1**: First table.

**table2**: Second table

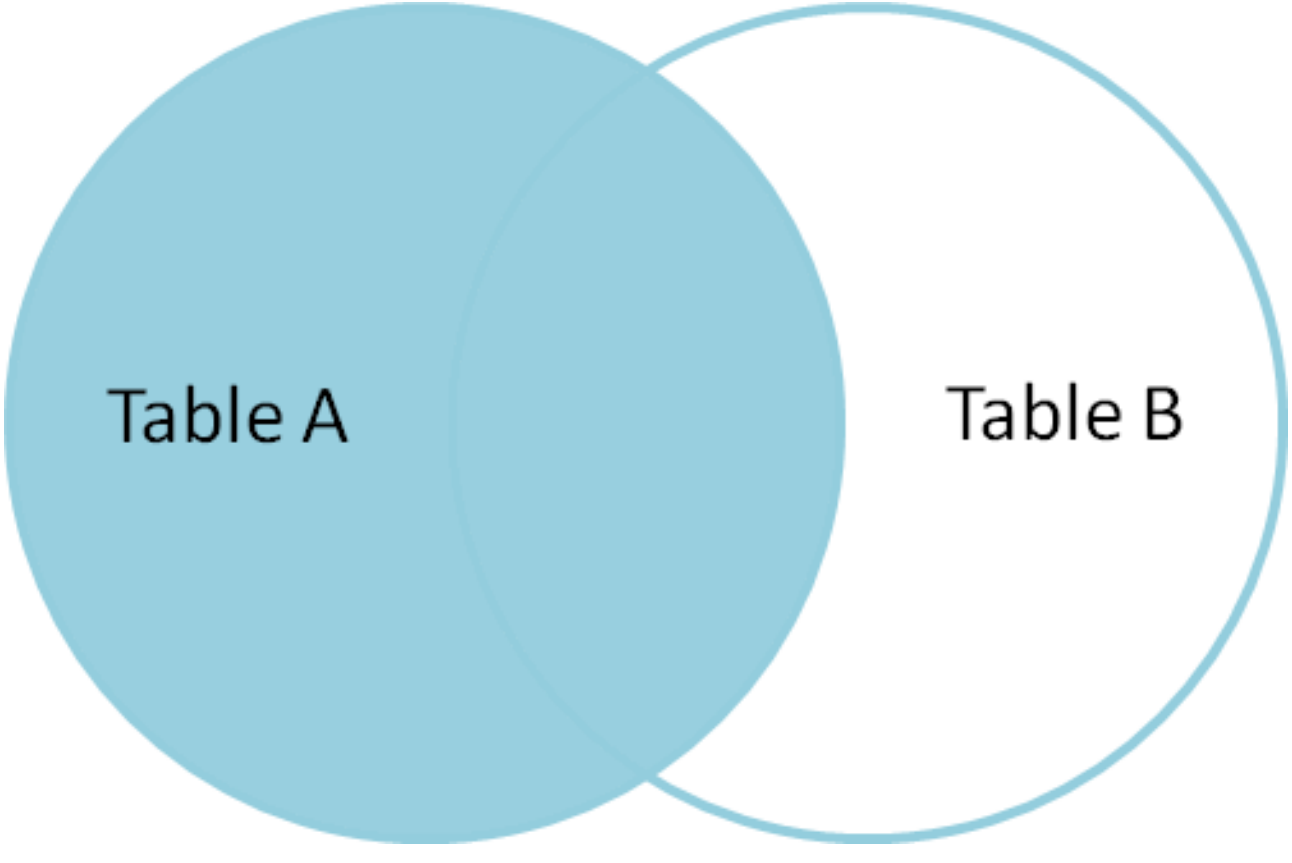
**matching_column**: Column common to both the tables.




| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2556889 | 12000 | 2 | RET | 4 | 1 | MARC | TESCO | M | M | Y |
| 1323598795 | 1550 | 1 | RET | 1 | 2 | ANNA | MARTIN | F | F | N |
| 2225546 | 5000 | 2 | RET | 5 | 3 | EMMA | JOHNSON | F | F | Y |
| 5516229 | 6000 | 5 | RET | 4 | 4 | DARIO | PENTAL | M | M | N |
| 5356222 | 7500 | 5 | RET | 5 | 5 | ELENA | SIMSON | F | F | N |
| 2221889 | 5400 | 2 | RET | 1 | 6 | TIM | ROBITH | M | M | N |
| 2455688 | 12500 | 2 | CORP | 50 | 7 | MILA | MORRIS | F | F | N |
| 1322488656 | 2500 | 1 | CORP | 51 | 8 | JENNY | DWARTH | F | F | Y |
| 1323598795 | 3100 | 1 | CORP | 52 | | | | | | |
| 1323111595 | 1220 | 1 | CORP | 53 | | | | | | |

# LEFT JOIN (or LEFT OUTER JOIN):

**This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN.**



➡️Returns all rows from the left table and matching rows from the right table.

➡️If there's no match in the right table, it fills with NULL values.

➡️Ideal for situations where you want all records from the left table and related data from the right table.

Syntax:

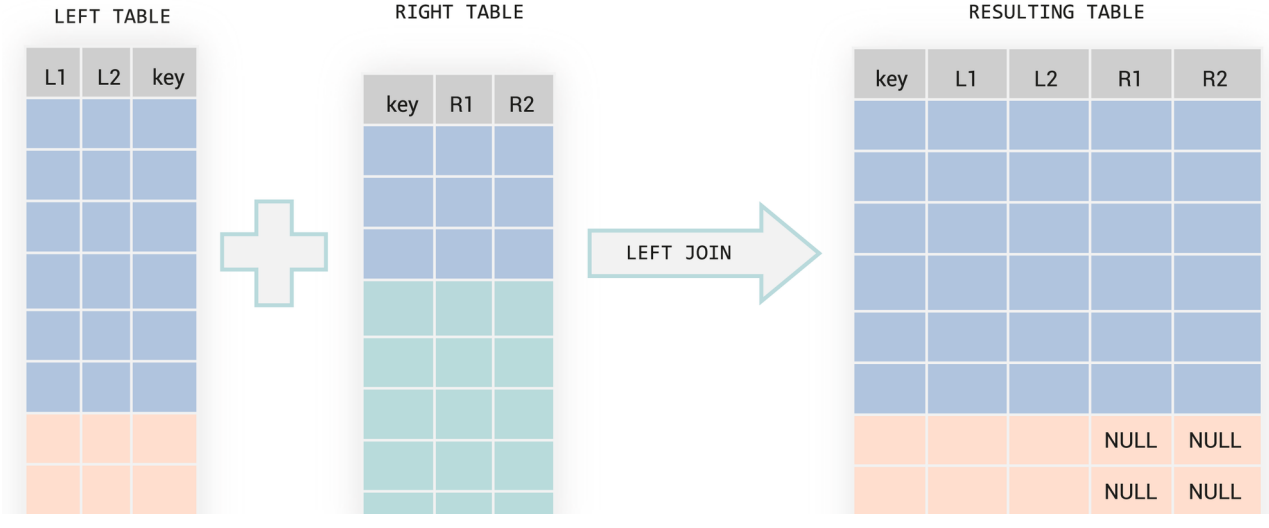
```
SELECT table1.column1,table1.column2,table2.column1,....
FROM table1
LEFT JOIN table2
ON table1.matching_column = table2.matching_column;
```

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

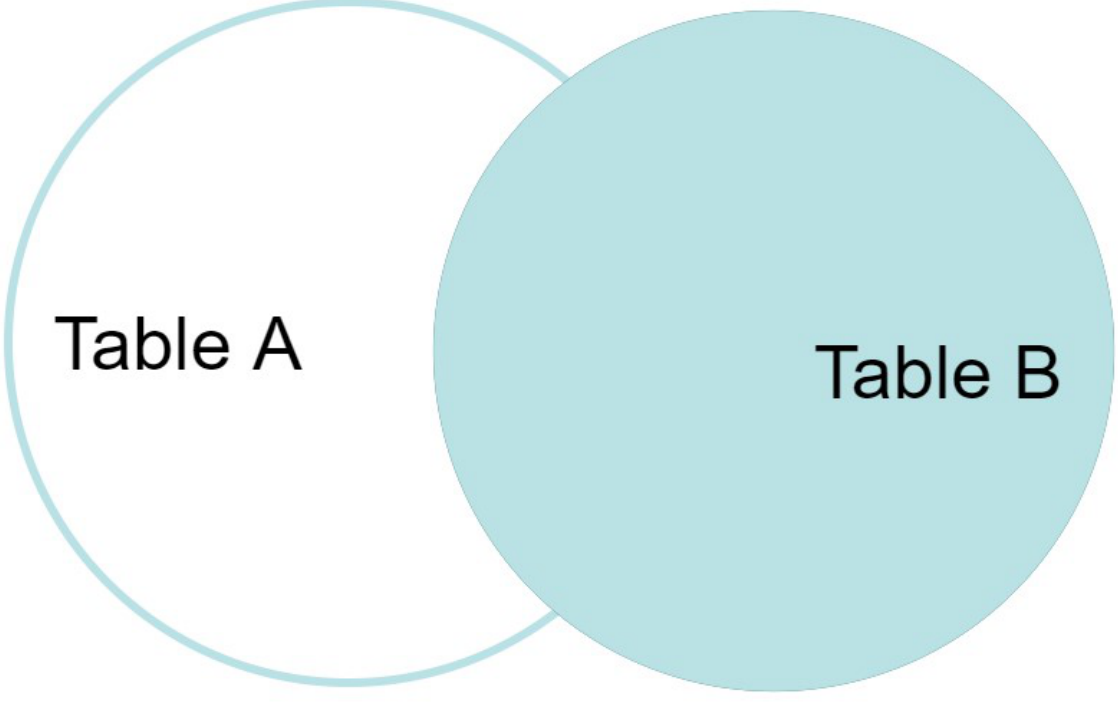
Note: We can also use LEFT OUTER JOIN instead of LEFT JOIN, both are the same.




# RIGHT JOIN

RIGHT JOIN **keeps all of the records from the right table, even if they cannot be matched to the left table**.

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.




Syntax:

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

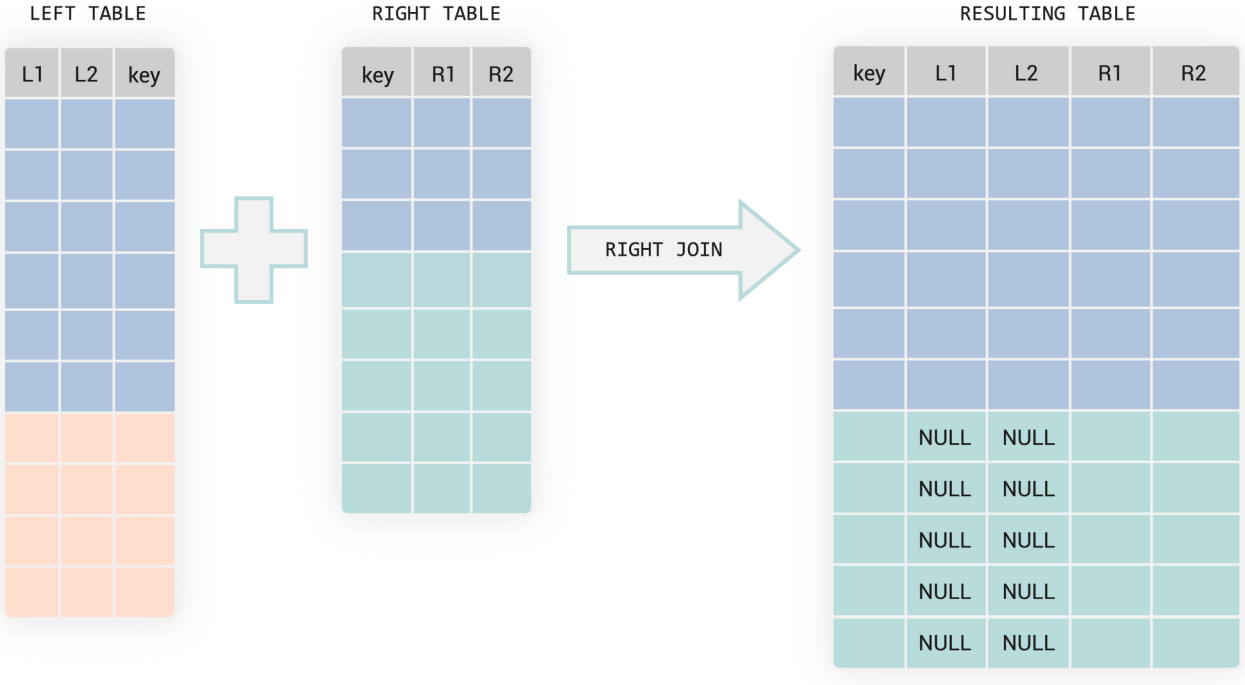
RIGHT JOIN table2

ON table1.matching_column = table2.matching_column;

table1: First table.

table2: Second table

matching_column: Column common to both the tables.




# 🟡 OUTER JOIN (or FULL OUTER JOIN):

**FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain *NULL* values.**

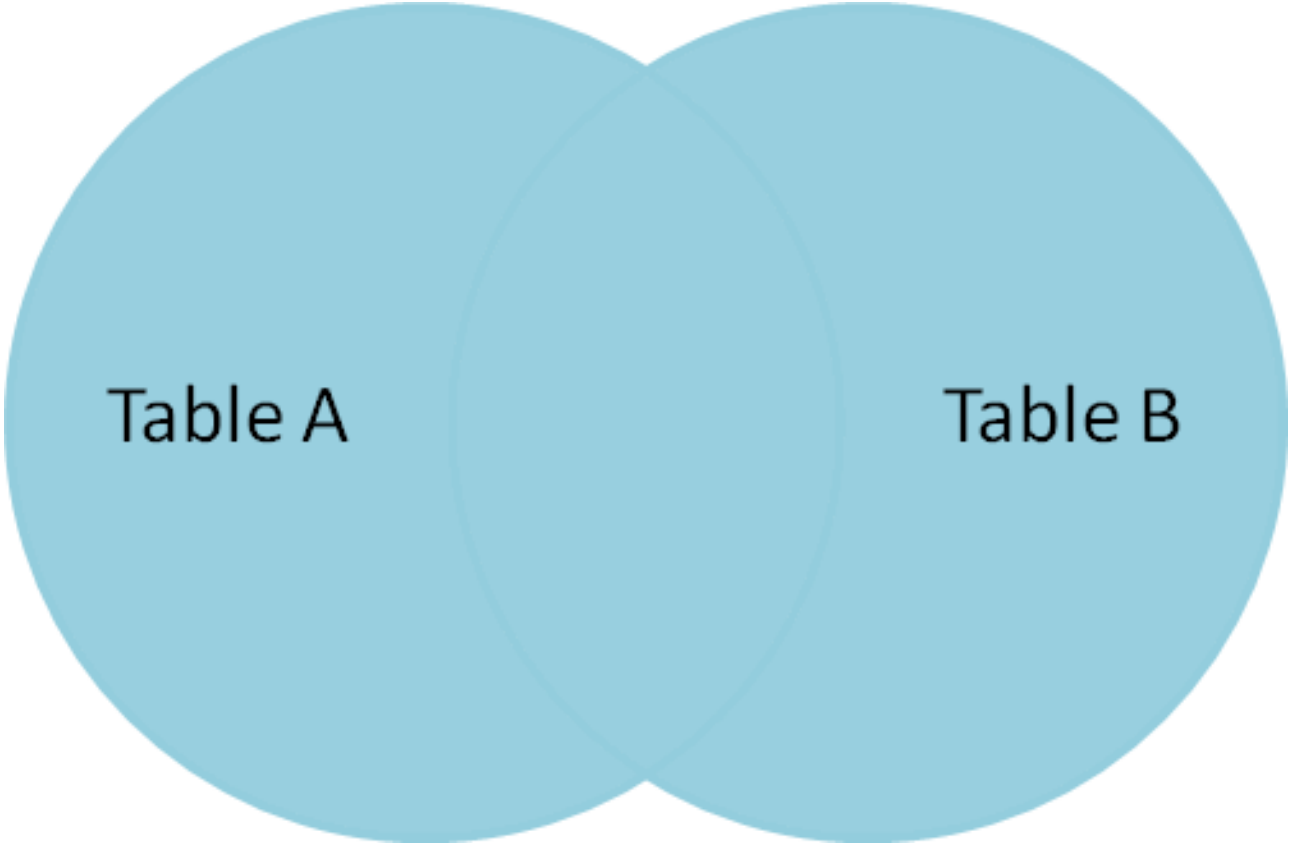
**Syntax:**

```
SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

FULL JOIN table2

ON table1.matching_column = table2.matching_column;
```

**table1: First table.**

**table2: Second table**

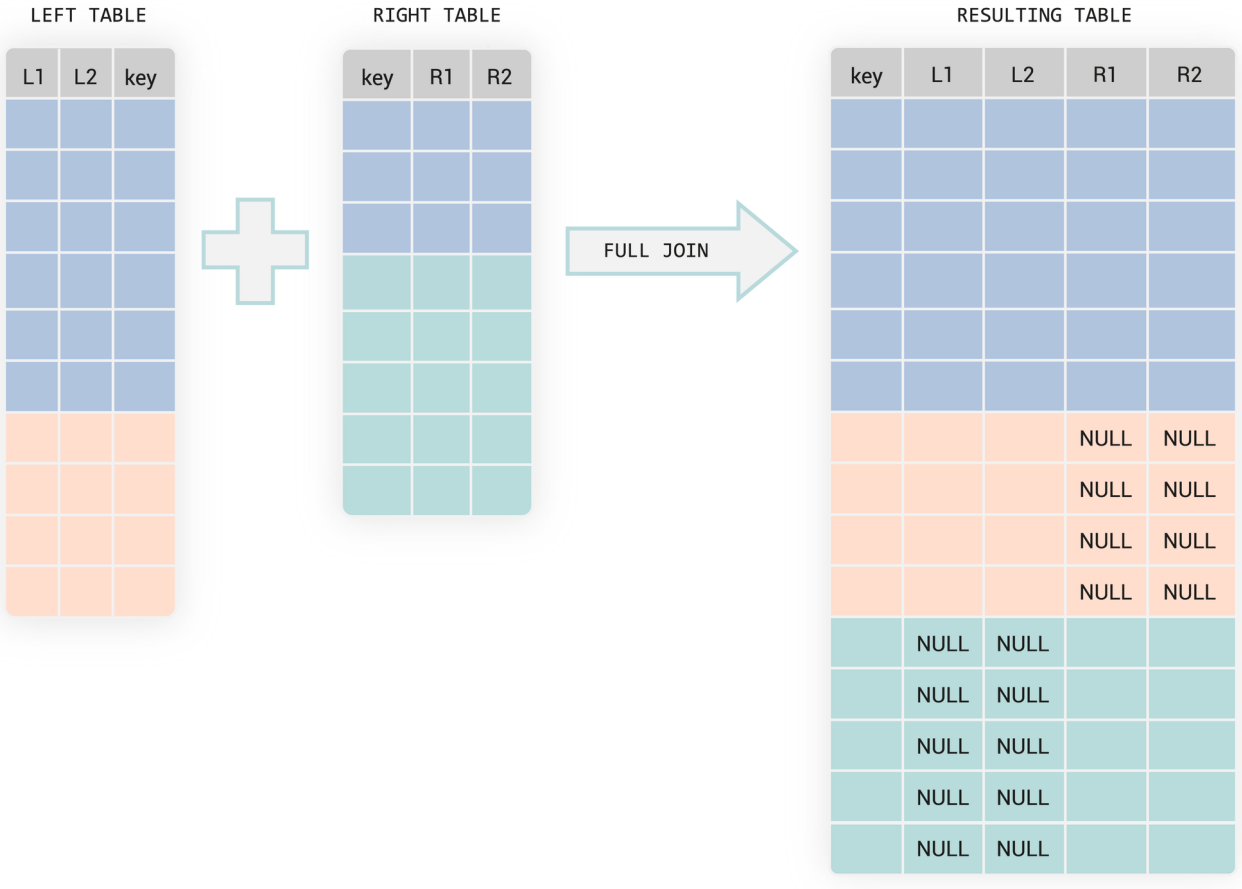
**matching_column: Column common to both the tables.**




➡️Returns all rows when there is a match in either the left or right table.

➡️Results in NULL values where there are no matches in either the left or right table.

➡️Useful for retrieving all available data from both tables, including unmatched records.




@pintukushwaha