Joins

JOIN combines data from two tables.It is used to combine rows from two or more than two tables based on any common record between them.

There are six types of the JOINs in SQL:

1. **INNER JOIN**: Returns records that have matching values in both tables
2. **LEFT JOIN**: Returns all records from the left table, and the matched records from the right table
3. **RIGHT JOIN**: Returns all records from the right table, and the matched records from the left table
4. **FULL JOIN:** Returns all records when there is a match in either left or right table
5. **CROSS JOIN:** Returns the cartesian product of the tables.
6. **SELF JOIN:** Returns records by joining the table with itself based on some condition.

Example - The two tables taken for understanding joins are **employee** which contains employee id as well as employee name. The second table **address** contains employee id and the city they live in.

employee

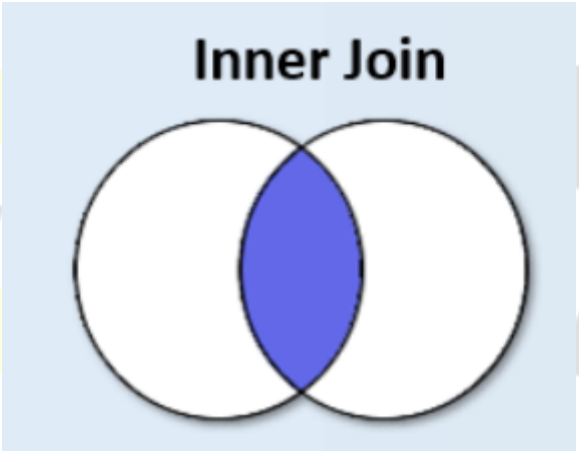
EMP_ID	EMP_NAME
1	MANAV
2	ROHAN
3	NIKHIL

address

EMP_ID	CITY
1	DELHI
2	MUMBAI
5	PUNE
6	CHENNAI

Types of Joins

1. **Inner Join** -
    - Inner Join gets all records that are common between both tables based on the supplied ON clause.
    - It selects records that have matching values in both tables.



Syntax -

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

Example -

```
SELECT EMP_ID, EMP_NAME, CITY
FROM employee
INNER JOIN address
ON employee.EMP_ID = address.EMP_ID
```

Result -

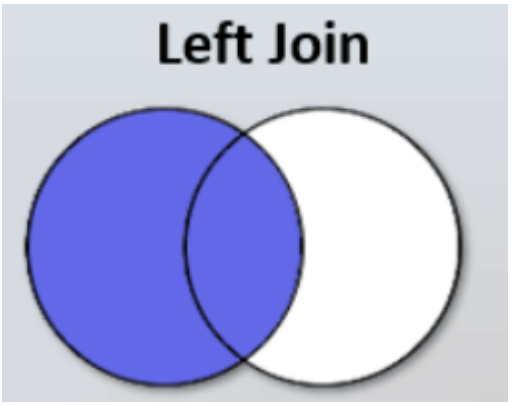
EMP_ID	EMP_NAME	CITY
1	MANAV	DELHI
2	ROHAN	MUMBAI

2. LEFT JOIN

- LEFT JOIN returns all rows from the left table with matching rows from the right table.
- Rows without a match are filled with NULL

Syntax -

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```



Example -

```
SELECT EMP_ID, EMP_NAME, CITY
FROM employee
LEFT JOIN address
ON employee.EMP_ID = address.EMP_ID
```

RESULT-

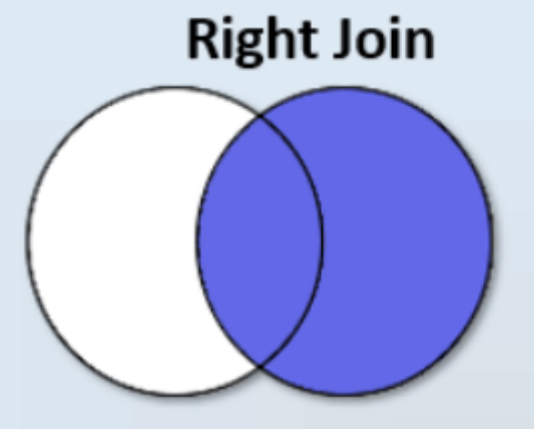
EMP_ID	EMP_NAME	CITY
1	MANAV	DELHI
2	ROHAN	MUMBAI
3	NIKHIL	NULL

3. **RIGHT Join**

   a. RIGHT JOIN returns all rows from the right table with matching rows from the left table.
   b. Rows without a match are filled with NULL

Syntax-

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```



Example -

```
SELECT EMP_ID, EMP_NAME, CITY
FROM employee
RIGHT JOIN address
ON employee.EMP_ID = address.EMP_ID
```

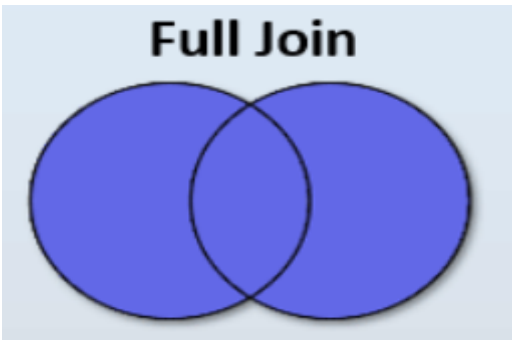
RESULT -

EMP_ID	EMP_NAME	CITY
1	MANAV	DELHI
2	ROHAN	MUMBAI
5	NULL	PUNE
6	NULL	CHENNAI

4. **OUTER JOIN**
   a. It returns all rows from the left table and all rows from the right table
   b. It fills the non-matching rows with NULL

Syntax -

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```



Example -

```
SELECT EMP_ID, EMP_NAME, CITY
FROM employee
OUTER JOIN address
ON employee.EMP_ID = address.EMP_ID
```

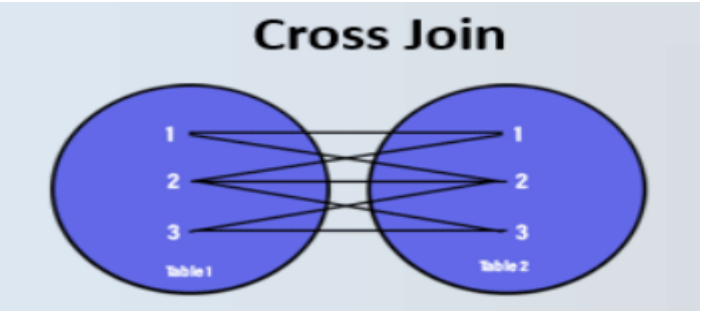
Result -

EMP_ID	EMP_NAME	CITY
1	MANAV	DELHI
2	ROHAN	MUMBAI
3	NIKHIL	NULL
5	NULL	PUNE
6	NULL	CHENNAI

5. **Cross Join**
    a. It returns all possible combinations of rows from left and right tables.
    b. The cross join joins each row from table A to every row available in table B. Therefore, the output is also known as a Cartesian product of both tables.

Syntax -

```
SELECT *
FROM table1
CROSS JOIN table2;
```



Example -

```
SELECT EMP_NAME, CITY
FROM employee
CROSS JOIN address ;
```

OUTPUT -

EMP_NAME	CITY
MANAV	DELHI
MANAV	MUMBAI
MANAV	PUNE
MANAV	CHENNAI
ROHAN	DELHI
ROHAN	MUMBAI
ROHAN	PUNE
ROHAN	CHENNAI

NIKHIL	DELHI
NIKHIL	MUMBAI
NIKHIL	PUNE
NIKHIL	CHENNAI

6. SELF JOIN

a. In this Join, SQL Server joins the table with itself. This means the table name appears twice in the from clause.
b. Each row of the table is joined with itself and all other rows depending on some conditions.

Consider the employee table, we will perform self-join on this.

employee

EMP_ID	EMP_NAME
1	MANAV
2	ROHAN
3	NIKHIL

Syntax:

```
SELECT a.coulmn1 , b.column2
FROM table_name a, table_name b
WHERE some_condition;
```

Example-

```
SELECT a.EMP_ID , b.EMP_NAME
FROM employee a, employee b
WHERE a.EMP_ID < b.EMP_ID;
```

Output -

EMP_ID	EMP_NAME
1	ROHAN
1	NIKHIL
2	NIKHIL