Joining Tables

- **What are JOINS in SQL?**

JOIN is an operation that exists in SQL that helps us to combine rows from two or more tables based on a related column between them.

- **SQL Aliases –**

Alias is a concept that is used to give temporary names to tables or particular column in a table. They exist till the duration of the query. Alias name can be give through “AS” keyword or without it.

Syntax-1 –

```
SELECT t.col_name
FROM <table-name> t;
```

Syntax-2 –

```
SELECT t.col_name
FROM <table-name> AS t;
```

- **Types of JOIN**:-

1. INNER JOIN - This returns a resulting table that has matching values from both the table or all the tables.

2. OUTER JOIN – We have 3 types of OUTER JOINS -

- **LEFT OUTER JOIN** - This returns a resulting table that all the data from left table and the matched data from the right table
- **RIGHT OUTER JOIN** - This returns a resulting table that all the data from right table and the matched data from the left table
- **FULL OUTER JOIN** - This returns a resulting table that contains all data when there is a match on left or right table data. It can be implemented using UNION operation.

3. CROSS JOIN – This returns all the cartesian product of the data present in both the tables. Hence, all possible variations are reflected in the output.

4. SELF JOIN – It is used to get the output from a particular table, when the same table is joined to itself.

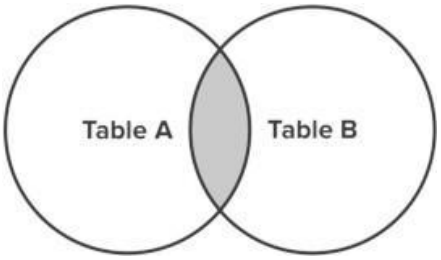
NOTE-1 : It is not compulsory to use "OUTER" keyword in the joins. We can specify the join as LEFT JOIN or LEFT OUTER JOIN, there is no difference or error in the output.

NOTE-2 : Query - ***Table_Name.****

Output – All the data from particular table. This kind of query is used in case of multiple tables.

NOTE-3: The JOINS can be used interchangeably (LEFT OUTER & RIGHT OUTER etc), and a query can be amended to change the particular join into another alternate join, but the output will be the same. There will be no error in doing so.

- **INNER JOIN -**



General form -

```
SELECT *
FROM TableA
INNER JOIN TableB
ON TableA.column1 = TableB.column1;
```

Example:

Table 1:- NINJA table is shown below -

Ninja_ID	Name	CITY
1	Ojasv Ninja	Jaipur
2	Tejas Ninja	Trichy
3	Rejas Ninja	Manipal

Table 2:- ORDERS

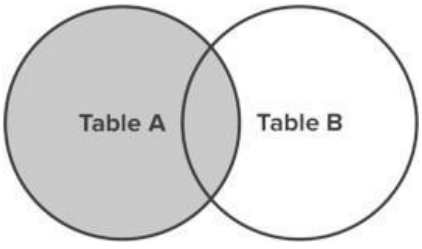
Order_ID	Product_name	Ninja_ID
1	SQL	1
2	WEBDEV	2
3	CP	3
4	ALGO	4

```
SELECT Ninja_ID, Name, City, Product_name
FROM NINJA
INNER JOIN ORDERS
ON NINJA.Ninja_ID = ORDERS.Order_ID;
```

Output:-

Ninja_ID	Name	City	Product_name
1	Ojasv Ninja	Jaipur	SQL
2	Tejas Ninja	Trichy	WEBDEV
3	Rejas Ninja	Manipal	CP

- **LEFT JOIN -**



General form -

```
SELECT *
FROM TableA
LEFT JOIN TableB
ON TableA.column1 = TableB.column1;
```

Example:
Table 1: NINJA table is shown below -

Ninja_ID	Name	CITY
1	Ojasv Ninja	Jaipur
2	Tejas Ninja	Trichy
3	Rejas Ninja	Manipal
5	Kejas Ninja	Lucknow

Table 2: ORDERS

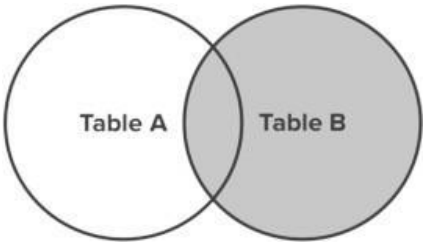
Order_ID	Product_name	Ninja_ID
1	SQL	1
2	WEBDEV	2
3	CP	3
4	ALGO	4

```
SELECT Ninja_ID, Name, City, Product_name
FROM NINJA
INNER JOIN ORDERS
ON NINJA.Ninja_ID = ORDERS.Order_ID;
```

Output:-

Ninja_ID	Name	City	Product_name
1	Ojasv Ninja	Jaipur	SQL
2	Tejas Ninja	Trichy	WEBDEV
3	Rejas Ninja	Manipal	CP
5	Kejas Ninja	Lucknow	NULL

- **RIGHT JOIN -**



General form -

```
SELECT *
FROM TableA
RIGHT JOIN TableB
ON TableA.column1 = TableB.column1;
```

Example:

Table 1: NINJA table is shown below -

Ninja_ID	Name	CITY
1	Ojasv Ninja	Jaipur
2	Tejas Ninja	Trichy
3	Rejas Ninja	Manipal
5	Kejas Ninja	Lucknow

Table 2: ORDERS

Order_ID	Product_name	Ninja_ID
1	SQL	1
2	WEBDEV	2
3	CP	3
4	ALGO	4

Query:

```
SELECT Ninja_ID, Name, City, Product_name
FROM NINJA
RIGHT JOIN ORDERS
ON NINJA.Ninja_ID = ORDERS.Order_ID;
```

Output:

Ninja_ID	Name	City	Product_name
1	Ojasv Ninja	Jaipur	SQL
2	Tejas Ninja	Trichy	WEBDEV
3	Rejas Ninja	Manipal	CP
NULL	NULL	NULL	ALGO

- **FULL JOIN -**

General form -

```
SELECT * FROM TableA
LEFT JOIN TableB ON TableA.column1 = TableB.column2
UNION
SELECT * FROM TableA
RIGHT JOIN TableB ON TableA.column1 = TableB.column2
```

Example:

Table 1: NINJA

Ninja_ID	Name	CITY
1	Ojasv Ninja	Jaipur
2	Tejas Ninja	Trichy
3	Rejas Ninja	Manipal
5	Kejas Ninja	Lucknow

Table 2: ORDERS

Order_ID	Product_name	Ninja_ID
1	SQL	1
2	WEBDEV	2
3	CP	3
4	ALGO	4

Query:

```
SELECT Ninja_ID, Name, City, Product_name
FROM NINJA
LEFT JOIN ORDERS  ON NINJA.Ninja_ID = ORDERS.Order_ID
UNION
SELECT Ninja_ID, Name, City, Product_name
FROM NINJA
RIGHT JOIN ORDERS  ON NINJA.Ninja_ID = ORDERS.Order_ID;
```

Output:

Ninja_ID	Name	City	Product_name
1	Ojasv Ninja	Jaipur	SQL
2	Tejas Ninja	Trichy	WEBDEV
3	Rejas Ninja	Manipal	CP
3	Kejas Ninja	Lucknow	NULL
NULL	NULL	NULL	ALGO

- **CROSS JOIN & FULL JOIN -**

The CROSS join produces the cartesian product of all the data present in both tables. Hence, all possible combinations or variations will be reflected, also it doesn't have ON clause, as we don't require any condition as we are joining everything to everything. A FULL JOIN is a combination of LEFT JOIN and RIGHT JOIN and gives the value output as per the WHERE clause condition.

- **CROSS JOIN –**

General form -

```
SELECT *
FROM TableA
CROSS JOIN TableA;
```

Example:
Table 1: NINJA

Ninja_ID	Name	CITY
1	Ojasv Ninja	Jaipur
2	Tejas Ninja	Trichy
3	Rejas Ninja	Manipal
5	Kejas Ninja	Lucknow

Table 2: ORDERS

Order_ID	Product_name	Ninja_ID
1	SQL	1
2	WEBDEV	2
3	CP	3
4	ALGO	4

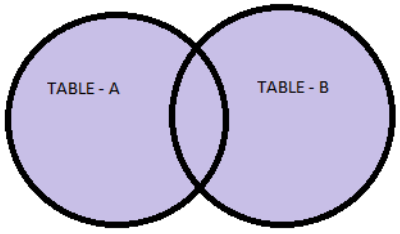
Query:

```
SELECT NINJA.Ninja_ID, NINJA.Name, NINJA.City, ORDERS.Product_name
FROM NINJA
CROSS JOIN ORDERS;
```

Output:

Ninja_ID	Name	City	Product_name
1	Ojasv Ninja	Jaipur	SQL
2	Tejas Ninja	Trichy	SQL
3	Rejas Ninja	Manipal	SQL
5	Kejas Ninja	Lucknow	SQL
1	Ojasv Ninja	Jaipur	WEBDEV
2	Tejas Ninja	Trichy	WEBDEV
3	Rejas Ninja	Manipal	WEBDEV
5	Kejas Ninja	Lucknow	WEBDEV
1	Ojasv Ninja	Jaipur	CP
2	Tejas Ninja	Trichy	CP
3	Rejas Ninja	Manipal	CP
5	Kejas Ninja	Lucknow	CP
1	Ojasv Ninja	Jaipur	ALGO
2	Tejas Ninja	Trichy	ALGO
3	Rejas Ninja	Manipal	ALGO
5	Kejas Ninja	Lucknow	ALGO

- **SELF JOIN -**



General form -

```
SELECT A.Col_1, B.Col_2
FROM TableA  A, TABLEA  B
WHERE A.COL_NAME = B.COL_NAME
AND <condition>;
```

Example:

Table 1: NINJA table is shown below -

Ninja_ID	Name	CITY
1	Ojasv Ninja	Jaipur
2	Tejas Ninja	Trichy
3	Rejas Ninja	Manipal

```
SELECT a.Ninja_ID, b.Name
FROM NINJA a, NINJA b
WHERE a.Ninja_ID = b.Ninja_ID
AND a.City = “Jaipur”
```

Output:-

Ninja_ID	Name
1	Ojasv Ninja