# SQL JOINS

**By :**
**Dr. Rinkle Rani**
**Associate Professor, CSED**
**TIET, Patiala**

## What is SQL Join?

An SQL JOIN clause combines rows from two or more tables based on a common field between them. It creates a set of rows in a temporary table.

A "Join" can be recognized in a SQL SELECT statement if it has more than one table after the FROM keyword.

For example:

```
SELECT "list-of-columns"

FROM table1, table2
```

# List of SQL JOINS

1. INNER JOIN
2. LEFT JOIN OR LEFT OUTER JOIN
3. RIGHT JOIN OR RIGHT OUTER JOIN
4. FULL OUTER JOIN
5. NATURAL JOIN
6. CROSS JOIN
7. SELF JOIN

## Equi Join or Inner Join

The Inner Join or Equi Join returns all rows from both tables where there is a match. Or in other words, if the query is relating two tables using an equality operator (=), it is an equality join, also known as an inner join or an Equi Join.

Syntax:

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

INNER JOIN table2

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

**OR**

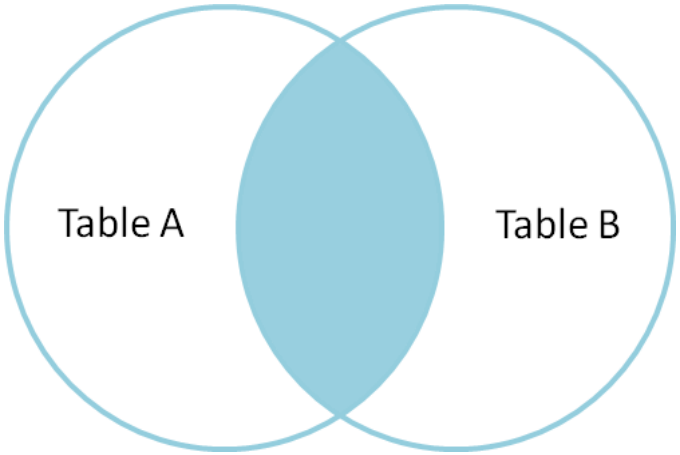
```
SELECT field1, field2, field3

FROM first_table , second_table

WHERE  first_table.keyfield = second_table.keyfield;
```

List the employee name with their department names.

```
SQL>SELECT ename, dname
        FROM Emp,dept
        where   emp.deptno=dept.deptno ;
```




The Output is:

ENAME DNAME

| ROLL_NO | NAME | ADDRESS | PHONE | Age |
|---|---|---|---|---|
| 1 | HARSH | DELHI | XXXXXXXXXX | 18 |
| 2 | PRATIK | BIHAR | XXXXXXXXXX | 19 |
| 3 | RIYANKA | SILIGURI | XXXXXXXXXX | 20 |
| 4 | DEEP | RAMNAGAR | XXXXXXXXXX | 18 |
| 5 | SAPTARHI | KOLKATA | XXXXXXXXXX | 19 |
| 6 | DHANRAJ | BARABAJAR | XXXXXXXXXX | 20 |
| 7 | ROHIT | BALURGHAT | XXXXXXXXXX | 18 |
| 8 | NIRAJ | ALIPUR | XXXXXXXXXX | 19 |

**STUDENT**

| COURSE_ID | ROLL_NO |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 2 | 3 |
| 3 | 4 |
| 1 | 5 |
| 4 | 9 |
| 5 | 10 |
| 4 | 11 |

**StudentCourse**

**Show the names and age of students enrolled in different courses.**

```
SELECT StudentCourse.COURSE_ID, Student.NAME, Student.AGE
FROM Student
INNER JOIN StudentCourse
ON Student.ROLL_NO = StudentCourse.ROLL_NO;
```

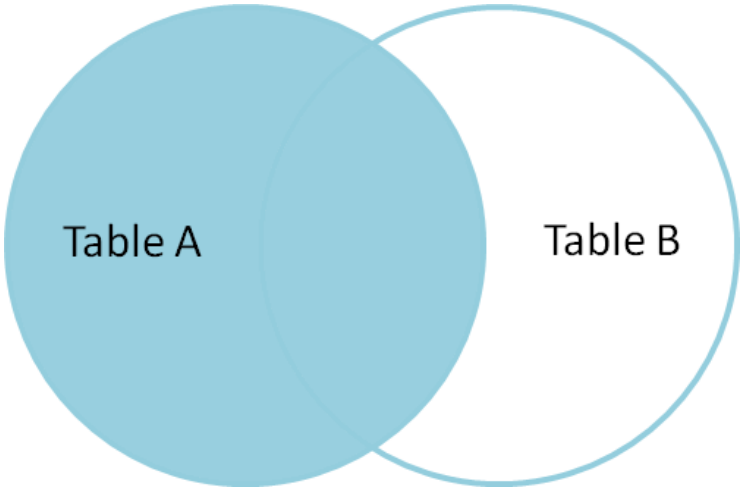
Output :

| COURSE_ID | NAME | Age |
|---|---|---|
| 1 | HARSH | 18 |
| 2 | PRATIK | 19 |
| 2 | RIYANKA | 20 |
| 3 | DEEP | 18 |
| 1 | SAPTARHI | 19 |

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

**Syntax:**

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



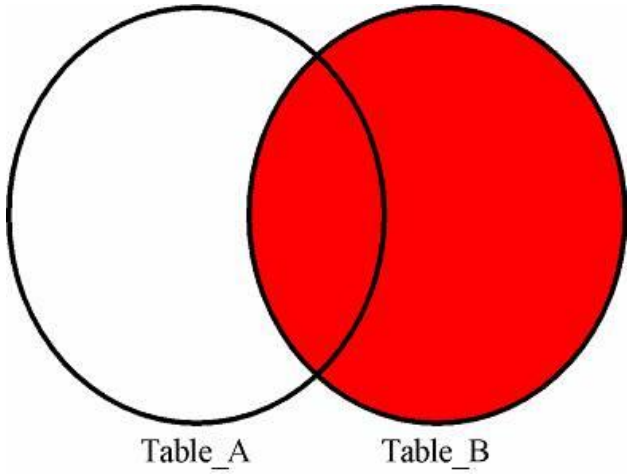
```
SELECT Student.NAME,StudentCourse.COURSE_ID
FROM Student LEFT JOIN StudentCourse ON
StudentCourse.ROLL_NO = Student.ROLL_NO;
```

**Output**:

| NAME | COURSE_ID |
|---|---|
| HARSH | 1 |
| PRATIK | 2 |
| RIYANKA | 2 |
| DEEP | 3 |
| SAPTARHI | 1 |
| DHANRAJ | *NULL* |
| ROHIT | *NULL* |
| NIRAJ | *NULL* |

**RIGHT JOIN**: 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 join.
The rows for which there is no matching row on 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;



```
SELECT Student.NAME,StudentCourse.COURSE_ID
FROM Student RIGHT JOIN StudentCourse
 ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

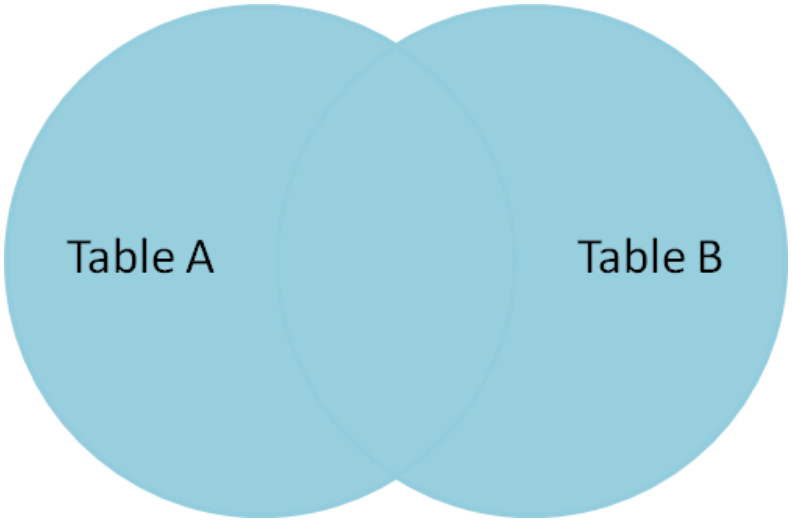
**Output:**

| NAME | COURSE_ID |
|---|---|
| HARSH | 1 |
| PRATIK | 2 |
| RIYANKA | 2 |
| DEEP | 3 |
| SAPTARHI | 1 |
| *NULL* | 4 |
| *NULL* | 5 |
| *NULL* | 4 |

**FULL JOIN:** FULL JOIN creates the result-set by combining result of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both the tables. 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;
```



```
SELECT Student.NAME,StudentCourse.COURSE_ID
FROM Student FULL JOIN StudentCourse
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

**Output:**

| NAME | COURSE_ID |
|---|---|
| HARSH | 1 |
| PRATIK | 2 |
| RIYANKA | 2 |
| DEEP | 3 |
| SAPTARHI | 1 |
| DHANRAJ | *NULL* |
| ROHIT | *NULL* |
| NIRAJ | *NULL* |
| *NULL* | 9 |
| *NULL* | 10 |
| *NULL* | 11 |

**CARTESIAN JOIN**: The CARTESIAN JOIN is also known as CROSS JOIN. In a CARTESIAN JOIN there is a join for each row of one table to every row of another table.

This usually happens when the matching column or WHERE condition is not specified.In the absence of a WHERE condition the CARTESIAN JOIN will behave like a CARTESIAN PRODUCT . i.e., the number of rows in the result-set is the product of the number of rows of the two tables.

In the presence of WHERE condition this JOIN will function like a INNER JOIN.

Generally speaking, Cross join is similar to an inner join where the join-condition will always evaluate to True

**Syntax:**

SELECT table1.column1 , table1.column2, table2.column1...
FROM table1 CROSS JOIN table2;

In query we will select NAME and Age from Student table and COURSE_ID from StudentCourse table.
In the output each row of the table Student is joined with every row of the table StudentCourse.
The total rows in the result-set = 4 * 4 = 16.

```
SELECT Student.NAME,
Student.AGE,
StudentCourse.COURSE_ID
FROM Student CROSS JOIN
StudentCourse;
```

| NAME | AGE | COURSE_ID |
|---|---|---|
| Ram | 18 | 1 |
| Ram | 18 | 2 |
| Ram | 18 | 2 |
| Ram | 18 | 3 |
| RAMESH | 18 | 1 |
| RAMESH | 18 | 2 |
| RAMESH | 18 | 2 |
| RAMESH | 18 | 3 |
| SUJIT | 20 | 1 |
| SUJIT | 20 | 2 |
| SUJIT | 20 | 2 |
| SUJIT | 20 | 3 |
| SURESH | 18 | 1 |
| SURESH | 18 | 2 |
| SURESH | 18 | 2 |
| SURESH | 18 | 3 |

**SELF JOIN**: As the name signifies, in SELF JOIN a table is joined to itself. That is, each row of the table is joined with itself and all other rows depending on some conditions. In other words we can say that it is a join between two copies of the same table.

**Syntax:**

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

```
SELECT a.ROLL_NO , b.NAME
FROM Student a, Student b
WHERE a.ROLL_NO < b.ROLL_NO;
```

**Output:**

| ROLL_NO | NAME |
|---|---|
| 1 | RAMESH |
| 1 | SUJIT |
| 2 | SUJIT |
| 1 | SURESH |
| 2 | SURESH |
| 3 | SURESH |

**Natural Join :**

Natural Join joins two tables based on same attribute name and datatypes. The resulting table will contain all the attributes of both the table but keep only one copy of each common column.

**Example:**

Consider the two tables given below:

Student

| Roll_No | Name |
|---|---|
| 1 | A |
| 2 | B |
| 3 | C |

Marks

| Roll_No | Marks |
|---|---|
| 2 | 70 |
| 3 | 50 |
| 4 | 85 |

SELECT * FROM Student S NATURAL JOIN Marks M;

| Roll_No | Name | Marks |
|---|---|---|
| 2 | B | 70 |
| 3 | C | 50 |

## Difference between Natural join and Inner Join in SQL

SELECT * FROM student S INNER JOIN Marks M ON S.Roll_No = M.Roll_No;

| Roll_No | Name | Roll_No | Marks |
|---|---|---|---|
| 2 | B | 2 | 70 |
| 3 | C | 3 | 50 |

# Difference between Natural JOIN and INNER JOIN in SQL :

| SR.NO. | NATURAL JOIN | INNER JOIN |
|---|---|---|
| 1. | Natural Join joins two tables based on same attribute name and datatypes. | Inner Join joins two table on the basis of the column which is explicitly specified in the ON clause. |
| 2. | In Natural Join, The resulting table will contain all the attributes of both the tables but keep only one copy of each common column | In Inner Join, The resulting table will contain all the attribute of both the tables including duplicate columns also |
| 3. | In Natural Join, If there is no condition specifies then it returns the rows based on the common column | In Inner Join, only those records will return which exists in both the tables |
| 4. | SYNTAX:<br>SELECT *<br>FROM table1 NATURAL JOIN table2; | SYNTAX:<br>SELECT *<br>FROM table1 INNER JOIN table2 ON table1.Column_Name = table2.Column_Name; |