

JOINS

SQL joins are used to combine rows from two or more tables.

SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them.

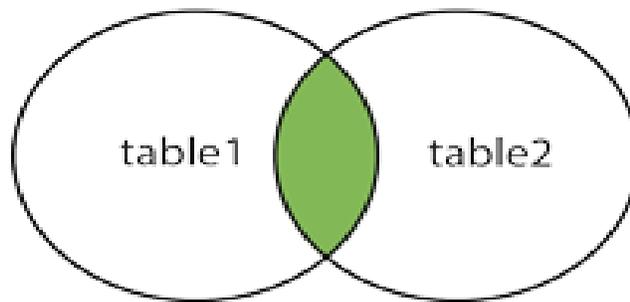
The most common type of join is: **SQL INNER JOIN (simple join)**. An SQL INNER JOIN returns all rows from multiple tables where the join condition is met.

Different SQL JOINS

Before we continue with examples, we will list the types of the different SQL JOINS you can use:

- **INNER JOIN:** Returns all rows when there is at least one match in BOTH tables

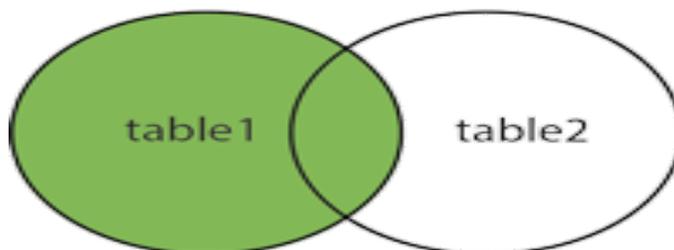
INNER JOIN



Note :- It is the common portion i.e. $A \cap B$ (Intersection)

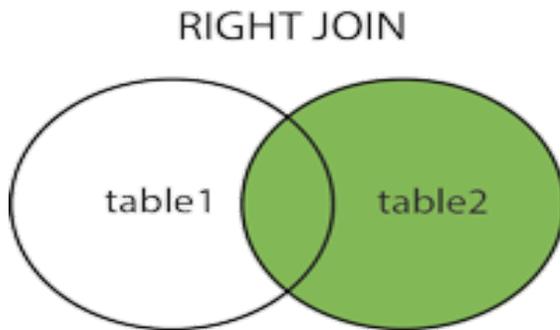
- **LEFT JOIN:** Return all rows from the left table, and the matched rows from the right table

LEFT JOIN



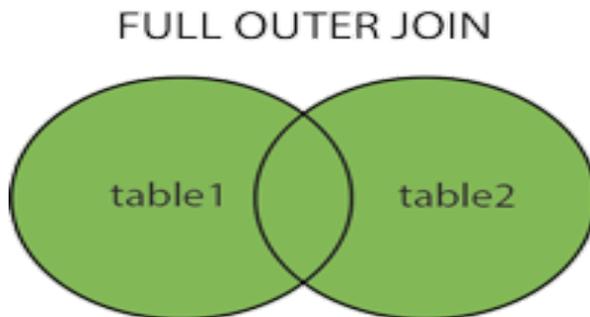
Note :- It is the difference i.e A - B

- **RIGHT JOIN:** Return all rows from the right table, and the matched rows from the left table.



Note :- It is the difference i.e B-A

- **FULL JOIN:** Return all rows when there is a match in ONE of the tables.



Note :- It is the Union of A and B.

Create tables below :-

```
SQL> select * from SQL;
```

SUBJECT	NAME	STU_ID
Math	Rohit	2
Science	Mohit	7

English	Ritu	3
Hindi	Rohit	5

SQL> select * from oracle;

NAME	STU_ID	GENDER	ADDRESS
Ritu	2	F	Mohali
Sonica	3	F	Chandigarh
Rohit	4	M	Mohali
Amit	5	M	Panchkula
Rohan	6	M	Chandigarh

INNER JOIN

The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns in both tables.

Syntax :-

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

SQL> select oracle.name, sql.subject from oracle inner join sql on oracle.stu_id = sql.stu_id;

LEFT OUTER JOIN

The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.

Syntax :-

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

SQL > select oracle.name,sql.subject from oracle left join sql on oracle.stu_id = sql.stu_id;

RIGHT OUTER JOIN

The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match.

Syntax :-

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

SQL> select sql.subject,oracle.name from sql right join oracle on sql.stu_id=oracle.stu_id;

FULL JOIN

The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2).

The FULL OUTER JOIN keyword combines the result of both LEFT and RIGHT joins.

Syntax :-

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
```

ON *table1.column_name=table2.column_name*;

**SQL>select sql.name,oracle.address,oracle.gender from sql full outer
join oracle on oracle.stu_id=sql.stu_id;**

CROSS or CARTESIAN JOIN

CROSS JOIN returns the [Cartesian product](#) of rows from tables in the join. In other words, it will produce rows which combine each row from the first table with each row from the second table

Syntax :-

The basic syntax of **CARTESIAN JOIN** or **CROSS JOIN** is as follows:

**SELECT table1.column1, table2.column2...
FROM table1, table2 [, table3]**

SQL>select oracle.name ,sql.stu_id from oracle,sql;

EQUI JOIN

An **equi-join** is a specific type of comparator-based join, that uses only [equality](#) comparisons in the join-predicate. Using other comparison operators (such as <) disqualifies a join as an equi-join.

Syntax :-

SQL>select * from oracle,sql where oracle.stu_id=sql.stu_id;