A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are:

* INNER JOIN
* LEFT JOIN
* RIGHT JOIN
* FULL JOIN

Consider the two tables below:

Student table:

Create table student (sid int primary key ,name varchar(20),address varchar(20),age int);

insert into student values(1,'ram','hyd',20),(2,'ravi','chennai',20),(3,'kiran','pune',20),(4,'giri','hyd',20),(5,'sai','hyd',20);

select \* from student;

Course Table:

Create table course (cid int primary key,cname varchar(20),startdate date,sid int references student(sid));

insert into course values(101,'java','2021/07/05',1);

insert into course values(102,'mysql','2021/07/05',2);

insert into course values(103,'oracle','2021/07/05',1);

insert into course values(104,'spring','2021/07/05',3);

insert into course values(105,'aws','2021/07/05',null);

insert into course values(106,'docker','2021/07/05',null);

select \* from course;

# INNER JOIN

select s.sid,s.name,s.address,s.age,c.cid,c.cname,c.startdate from student s inner join course c on s.sid=c.sid;

#LEFT OUTER JOIN

select s.sid,s.name,s.address,s.age,c.cid,c.cname,c.startdate from student s left outer join course c on s.sid=c.sid;

#RIGHT OUTER JOIN

select s.sid,s.name,s.address,s.age,c.cid,c.cname,c.startdate from student s right outer join course c on s.sid=c.sid;

# FULL OUTER JOIN

select s.sid,s.name,s.address,s.age,c.cid,c.cname,c.startdate from student s left outer join course c on s.sid=c.sid

union

select s.sid,s.name,s.address,s.age,c.cid,c.cname,c.startdate from student s right outer join course c on s.sid=c.sid;

The simplest Join is INNER JOIN.

1. **INNER JOIN:** The INNER JOIN keyword selects all rows from both the tables as long as the condition satisfies. 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 same.  
   **Syntax**:
2. SELECT table1.column1,table1.column2,table2.column1,....
3. FROM table1
4. INNER JOIN table2
5. ON table1.matching\_column = table2.matching\_column;
6. **table1**: First table.
7. **table2**: Second table
8. **matching\_column**: Column common to both the tables.

**Note**: We can also write JOIN instead of INNER JOIN. JOIN is same as INNER JOIN.



**Example Queries(INNER JOIN)**

* + This query will 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**:  
[](https://media.geeksforgeeks.org/wp-content/uploads/table22.png)

1. **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:**
2. SELECT table1.column1,table1.column2,table2.column1,....
3. FROM table1
4. LEFT JOIN table2
5. ON table1.matching\_column = table2.matching\_column;
6. table1: First table.
7. table2: Second table
8. matching\_column: Column common to both the tables.

**Note**: We can also use LEFT OUTER JOIN instead of LEFT JOIN, both are same.  
[](https://i.stack.imgur.com/VkAT5.png)

**Example Queries(LEFT JOIN)**:

SELECT Student.NAME,StudentCourse.COURSE\_ID

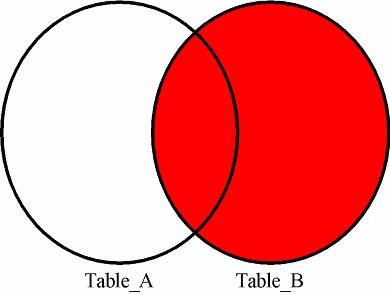
FROM Student

LEFT JOIN StudentCourse

ON StudentCourse.ROLL\_NO = Student.ROLL\_NO;

**Output**:  
[](https://media.geeksforgeeks.org/wp-content/uploads/table31.png)

1. **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:**
2. SELECT table1.column1,table1.column2,table2.column1,....
3. FROM table1
4. RIGHT JOIN table2
5. ON table1.matching\_column = table2.matching\_column;
6. table1: First table.
7. table2: Second table
8. matching\_column: Column common to both the tables.

**Note**: We can also use RIGHT OUTER JOIN instead of RIGHT JOIN, both are same.  


**Example Queries(RIGHT JOIN)**:

SELECT Student.NAME,StudentCourse.COURSE\_ID

FROM Student

RIGHT JOIN StudentCourse

ON StudentCourse.ROLL\_NO = Student.ROLL\_NO;

**Output:**  
[](https://media.geeksforgeeks.org/wp-content/uploads/table6.png)

1. **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:**
2. SELECT table1.column1,table1.column2,table2.column1,....
3. FROM table1
4. FULL JOIN table2
5. ON table1.matching\_column = table2.matching\_column;
6. table1: First table.
7. table2: Second table
8. matching\_column: Column common to both the tables.



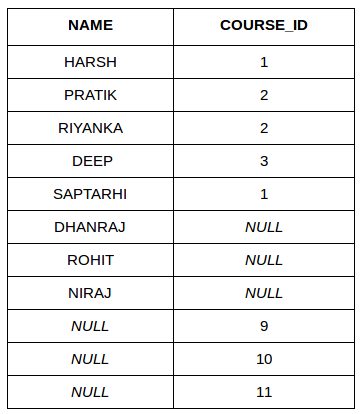
**Example Queries(FULL JOIN)**:

SELECT Student.NAME,StudentCourse.COURSE\_ID

FROM Student

FULL JOIN StudentCourse

ON StudentCourse.ROLL\_NO = Student.ROLL\_NO;

**Output:**  
[](https://media.geeksforgeeks.org/wp-content/uploads/table7.png)

SELF JOIN:

create table employee(eid int, ename varchar(20) , mid int);

insert into employee values(1,'ram',1),(2,'kiran',1),(3,'ravi',1),(4,'smitha',2),(5,'sir',3),(6,'ramesh',2);

select \* from employee;

select a.eid ManagerID , a.ename ManagerName ,b.ename Emplyee

from employee a

join employee b

on a.eid=b.mid;