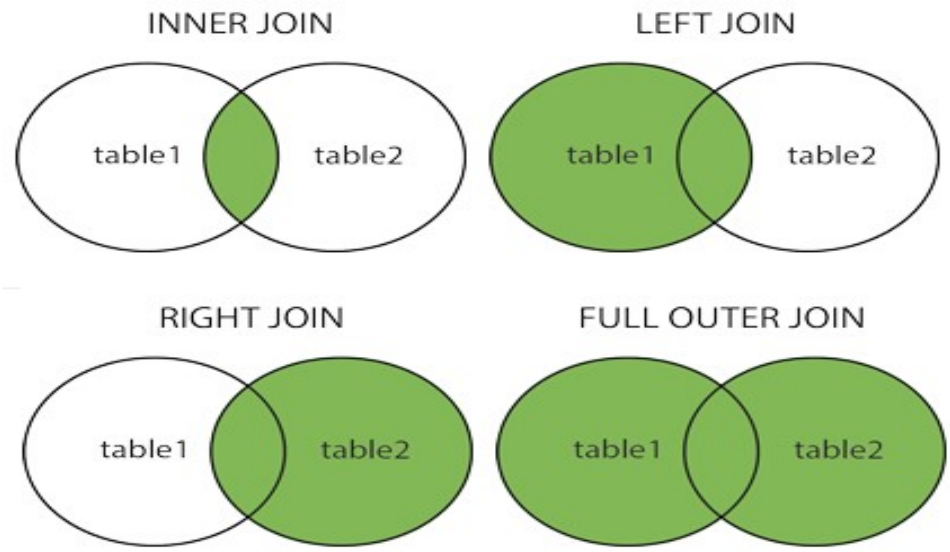
MYSQL JOINS

JOINS are used with SELECT statement. It is used to retrieve data from multiple tables. It is performed whenever you need to fetch records from two or more tables.

Here are the different types of the JOINs in SQL:

- ✓ **(INNER) JOIN**: Returns records that have matching values in both tables
- ✓ **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
- ✓ **RIGHT (OUTER) JOIN:** Returns all records from the right table, and the matched records from the left table
- ✓ **FULL (OUTER) JOIN:** Returns all records when there is a match in either left or right table



There are only three types of MySQL joins:

- MySQL **CROSS JOIN** (or sometimes called CARTESIAN join)
- MySQL **INNER JOIN** (or sometimes called simple join)
- MySQL **LEFT OUTER JOIN** (or sometimes called LEFT JOIN)
- MySQL **RIGHT OUTER JOIN** (or sometimes called RIGHT JOIN)
- MySQL **FULL JOIN** (union of LEFT JOIN and RIGHT JOIN)
- MySQL **NATURAL JOIN**

Let us take two exmple tables called **STUDENT** and **TEACHER**. STUDENT table has columns **roll_no**, **fname**, **lname**, **sex** and **state**. **TEACHER** table has columns **tid**, **t_fname**, **t_lname**, **sex** and **address**.

Let us create the **STUDENT** table by using the following statement:

```
CREATE TABLE student(
    roll_no int NOT NULL,
    fname varchar(45) NOT NULL,
    lname varchar(45) NOT NULL,
    sex varchar(10) NOT NULL,
    state varchar(20) NOT NULL );
```

Now, let us insert following records into this table EMPLOYEE by using following statements:

```
INSERT INTO student VALUES
(1,'Robin','Das','Male', 'Assam'),  (2,'Dwipen','Laskar', 'Male','Assam'),
(3,'Asangla','Sema', 'Female','Nagaland'), (4,'Praneeta','Rabha','Female', 'Manipur'),
(5,'Rosy','Kalita','Female', 'Tripura'),  (6,'Naznin','Akhtara','Female', 'Assam'),
(7,'Karabi','Bora', 'Female','Manipur'), (8,'Anil','Gogoi', 'Male','Manipur');
```

Now, we can see the recods of the table STUDENT by using SELECT * statement as below:

Statement: **SELECT * FROM student;**

Output:

```
mysql> SELECT * FROM student;
+---------+----------+---------+--------+----------+
| roll_no | fname    | lname   | sex    | state    |
+---------+----------+---------+--------+----------+
|       1 | Robin    | Das     | Male   | Assam    |
|       2 | Dwipen   | Laskar  | Male   | Assam    |
|       3 | Asangla  | Sema    | Female | Nagaland |
|       4 | Praneeta | Rabha   | Female | Manipur  |
|       5 | Rosy     | Kalita  | Female | Tripura  |
|       6 | Naznin   | Akhtara | Female | Assam    |
|       7 | Karabi   | Bora    | Female | Manipur  |
|       8 | Anil     | Gogoi   | Male   | Manipur  |
+---------+----------+---------+--------+----------+
8 rows in set (0.00 sec)
```

Let us create the **TEACHER** table by using the following statement:

```
CREATE TABLE teacher(
    tid int NOT NULL,
    t_fname varchar(45) NOT NULL,
    t_lname varchar(45) NOT NULL,
    sex varchar(10) NOT NULL,
    address varchar(20) NOT NULL );
```

Now, let us insert following records into this table TEACHER by using following statements:

INSERT INTO teacher VALUES

(1,'Monoj','Pathak','Male', 'Assam'), (2,'K.V.','Kanimozhi', 'Female','Kerala'),

(3,'Atul','Agnihotri', 'Male','Maharastra'), (4,'Praneeta','Bodo','Female', 'Assam'),

(5,'Priyanka','Bsumatari','Male', 'Mizoram'), (6,'Bikash','Punchal', 'Male','West Bangal');

Now, we can see the recods of the table TEACHER by using SELECT * statement as below:

Statement: **SELECT * FROM teacher;**

Output:

```
mysql> select * from teacher;
+-----+----------+-----------+--------+-------------+
| tid | t_fname  | t_lname   | sex    | address     |
+-----+----------+-----------+--------+-------------+
|   1 | Monoj    | Pathak    | Male   | Assam       |
|   2 | K.V.     | Kanimozhi | Female | Kerala      |
|   3 | Atul     | Agnihotri | Male   | Maharastra  |
|   4 | Praneeta | Bodo      | Female | Assam       |
|   5 | Priyanka | Bsumatari | Male   | Mizoram     |
|   6 | Bikash   | Punchal   | Male   | West Bangal |
+-----+----------+-----------+--------+-------------+
6 rows in set (0.00 sec)
```

1) The CROSS JOIN

Cross Join is essentially used to generate all possible combinations of the rows from two different (mostly unrelated) tables. In other words, it joins every row of first table with every row of second table thereby producing a Cartesian product. However, note that in practice it is not widely used.

Example: if we apply following statement then an CROSS join will be performed between STUDENT and TEACHER;

Statement:

SELECT * FROM student CROSS JOIN teacher;

OR

SELECT * FROM student,teacher;

Output:

It will return 48 rows i.e 8x6 rows from table STUDENT and TEACHER. Some of the records are:

```
mysql> SELECT * FROM student CROSS JOIN teacher;
+---------+----------+---------+--------+----------+-----+----------+-----------+--------+-------------+
| roll_no | fname    | lname   | sex    | state    | tid | t_fname  | t_lname   | sex    | address     |
+---------+----------+---------+--------+----------+-----+----------+-----------+--------+-------------+
|       1 | Robin    | Das     | Male   | Assam    |   6 | Bikash   | Punchal   | Male   | West Bangal |
|       1 | Robin    | Das     | Male   | Assam    |   5 | Priyanka | Bsumatari | Male   | Mizoram     |
|       1 | Robin    | Das     | Male   | Assam    |   4 | Praneeta | Bodo      | Female | Assam       |
|       1 | Robin    | Das     | Male   | Assam    |   3 | Atul     | Agnihotri | Male   | Maharastra  |
|       1 | Robin    | Das     | Male   | Assam    |   2 | K.V.     | Kanimozhi | Female | Kerala      |
|       1 | Robin    | Das     | Male   | Assam    |   1 | Monoj    | Pathak    | Male   | Assam       |
|       2 | Dwipen   | Laskar  | Male   | Assam    |   6 | Bikash   | Punchal   | Male   | West Bangal |
|       2 | Dwipen   | Laskar  | Male   | Assam    |   5 | Priyanka | Bsumatari | Male   | Mizoram     |
|       2 | Dwipen   | Laskar  | Male   | Assam    |   4 | Praneeta | Bodo      | Female | Assam       |
|       2 | Dwipen   | Laskar  | Male   | Assam    |   3 | Atul     | Agnihotri | Male   | Maharastra  |
|       2 | Dwipen   | Laskar  | Male   | Assam    |   2 | K.V.     | Kanimozhi | Female | Kerala      |
|       2 | Dwipen   | Laskar  | Male   | Assam    |   1 | Monoj    | Pathak    | Male   | Assam       |
|       3 | Asangla  | Sema    | Female | Nagaland |   6 | Bikash   | Punchal   | Male   | West Bangal |
|       3 | Asangla  | Sema    | Female | Nagaland |   5 | Priyanka | Bsumatari | Male   | Mizoram     |
```

2) The INNER JOIN

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. 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 the same.

Example: if we apply following statement then an inner join will be performed between STUDENT and TEACHER;

Statement:

SELECT * FROM student INNER JOIN teacher ON student.state = teacher.address ORDER;

Output:

```
mysql> SELECT * FROM student INNER JOIN teacher  ON student.state = teacher.address;
+---------+--------+---------+--------+-------+-----+----------+---------+--------+---------+
| roll_no | fname  | lname   | sex    | state | tid | t_fname  | t_lname | sex    | address |
+---------+--------+---------+--------+-------+-----+----------+---------+--------+---------+
|       1 | Robin  | Das     | Male   | Assam |   4 | Praneeta | Bodo    | Female | Assam   |
|       1 | Robin  | Das     | Male   | Assam |   1 | Monoj    | Pathak  | Male   | Assam   |
|       2 | Dwipen | Laskar  | Male   | Assam |   4 | Praneeta | Bodo    | Female | Assam   |
|       2 | Dwipen | Laskar  | Male   | Assam |   1 | Monoj    | Pathak  | Male   | Assam   |
|       6 | Naznin | Akhtara | Female | Assam |   4 | Praneeta | Bodo    | Female | Assam   |
|       6 | Naznin | Akhtara | Female | Assam |   1 | Monoj    | Pathak  | Male   | Assam   |
+---------+--------+---------+--------+-------+-----+----------+---------+--------+---------+
6 rows in set (0.00 sec)
```

3) The LEFT JOIN

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

Example: if we apply following statement then an LFFT JOIN will be performed between STUDENT and TEACHER;

Statement:

SELECT * FROM student LEFT JOIN teacher ON student.state = teacher.address;

Or

SELECT * FROM student LEFT OUTER JOIN teacher ON student.state = teacher.address;

Output:

```
mysql> SELECT * FROM student LEFT JOIN teacher  ON student.state = teacher.address;
+---------+----------+---------+--------+----------+------+----------+---------+--------+---------+
| roll_no | fname    | lname   | sex    | state    | tid  | t_fname  | t_lname | sex    | address |
+---------+----------+---------+--------+----------+------+----------+---------+--------+---------+
|       1 | Robin    | Das     | Male   | Assam    |    4 | Praneeta | Bodo    | Female | Assam   |
|       1 | Robin    | Das     | Male   | Assam    |    1 | Monoj    | Pathak  | Male   | Assam   |
|       2 | Dwipen   | Laskar  | Male   | Assam    |    4 | Praneeta | Bodo    | Female | Assam   |
|       2 | Dwipen   | Laskar  | Male   | Assam    |    1 | Monoj    | Pathak  | Male   | Assam   |
|       3 | Asangla  | Sema    | Female | Nagaland | NULL | NULL     | NULL    | NULL   | NULL    |
|       4 | Praneeta | Rabha   | Female | Manipur  | NULL | NULL     | NULL    | NULL   | NULL    |
|       5 | Rosy     | Kalita  | Female | Tripura  | NULL | NULL     | NULL    | NULL   | NULL    |
|       6 | Naznin   | Akhtara | Female | Assam    |    4 | Praneeta | Bodo    | Female | Assam   |
|       6 | Naznin   | Akhtara | Female | Assam    |    1 | Monoj    | Pathak  | Male   | Assam   |
|       7 | Karabi   | Bora    | Female | Manipur  | NULL | NULL     | NULL    | NULL   | NULL    |
|       8 | Anil     | Gogoi   | Male   | Manipur  | NULL | NULL     | NULL    | NULL   | NULL    |
+---------+----------+---------+--------+----------+------+----------+---------+--------+---------+
11 rows in set (0.04 sec)
```

4) The 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 the join. For the rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.

Example: if we apply following statement then an LFFT JOIN will be performed between STUDENT and TEACHER;

Statement:

SELECT * FROM student RIGHT JOIN teacher ON student.state = teacher.address;

Or

SELECT * FROM student RIGHT OUTER JOIN teacher ON student.state = teacher.address;

Output:

```
mysql> SELECT * FROM student RIGHT OUTER JOIN teacher  ON student.state = teacher.address;
+---------+--------+---------+--------+-------+-----+----------+-----------+--------+-------------+
| roll_no | fname  | lname   | sex    | state | tid | t_fname  | t_lname   | sex    | address     |
+---------+--------+---------+--------+-------+-----+----------+-----------+--------+-------------+
|       6 | Naznin | Akhtara | Female | Assam |   1 | Monoj    | Pathak    | Male   | Assam       |
|       2 | Dwipen | Laskar  | Male   | Assam |   1 | Monoj    | Pathak    | Male   | Assam       |
|       1 | Robin  | Das     | Male   | Assam |   1 | Monoj    | Pathak    | Male   | Assam       |
|    NULL | NULL   | NULL    | NULL   | NULL  |   2 | K.V.     | Kanimozhi | Female | Kerala      |
|    NULL | NULL   | NULL    | NULL   | NULL  |   3 | Atul     | Agnihotri | Male   | Maharastra  |
|       6 | Naznin | Akhtara | Female | Assam |   4 | Praneeta | Bodo      | Female | Assam       |
|       2 | Dwipen | Laskar  | Male   | Assam |   4 | Praneeta | Bodo      | Female | Assam       |
|       1 | Robin  | Das     | Male   | Assam |   4 | Praneeta | Bodo      | Female | Assam       |
|    NULL | NULL   | NULL    | NULL   | NULL  |   5 | Priyanka | Bsumatari | Male   | Mizoram     |
|    NULL | NULL   | NULL    | NULL   | NULL  |   6 | Bikash   | Punchal   | Male   | West Bangal |
+---------+--------+---------+--------+-------+-----+----------+-----------+--------+-------------+
10 rows in set (0.00 sec)
```

5) The FULL JOIN

In SQL, FULL JOIN is the result of a combination of both left and right outer join. Join tables have all the records from both tables. It puts NULL on the place of matches not found. In MySQL we cannot perform FULL join. We can perfrom it indirectly by taking the UNION of LEFT JOIN and RIGHT JOIN

Example: if we apply following statement then an FULL JOIN will be performed between STUDENT and TEACHER;

Statement:

```
(SELECT * FROM student LEFT JOIN teacher  ON teacher.address=student.state)
UNION
(SELECT * FROM student RIGHT JOIN teacher  ON teacher.address=student.state);
```

Output:

```
mysql> (SELECT * FROM student LEFT JOIN teacher  ON teacher.address=student.state)
    -> UNION
    -> (SELECT * FROM student RIGHT JOIN teacher  ON teacher.address=student.state);
+---------+----------+---------+--------+----------+------+----------+-----------+--------+-------------+
| roll_no | fname    | lname   | sex    | state    | tid  | t_fname  | t_lname   | sex    | address     |
+---------+----------+---------+--------+----------+------+----------+-----------+--------+-------------+
|       1 | Robin    | Das     | Male   | Assam    |    4 | Praneeta | Bodo      | Female | Assam       |
|       1 | Robin    | Das     | Male   | Assam    |    1 | Monoj    | Pathak    | Male   | Assam       |
|       2 | Dwipen   | Laskar  | Male   | Assam    |    4 | Praneeta | Bodo      | Female | Assam       |
|       2 | Dwipen   | Laskar  | Male   | Assam    |    1 | Monoj    | Pathak    | Male   | Assam       |
|       3 | Asangla  | Sema    | Female | Nagaland | NULL | NULL     | NULL      | NULL   | NULL        |
|       4 | Praneeta | Rabha   | Female | Manipur  | NULL | NULL     | NULL      | NULL   | NULL        |
|       5 | Rosy     | Kalita  | Female | Tripura  | NULL | NULL     | NULL      | NULL   | NULL        |
|       6 | Naznin   | Akhtara | Female | Assam    |    4 | Praneeta | Bodo      | Female | Assam       |
|       6 | Naznin   | Akhtara | Female | Assam    |    1 | Monoj    | Pathak    | Male   | Assam       |
|       7 | Karabi   | Bora    | Female | Manipur  | NULL | NULL     | NULL      | NULL   | NULL        |
|       8 | Anil     | Gogoi   | Male   | Manipur  | NULL | NULL     | NULL      | NULL   | NULL        |
|    NULL | NULL     | NULL    | NULL   | NULL     |    2 | K.V.     | Kanimozhi | Female | Kerala      |
|    NULL | NULL     | NULL    | NULL   | NULL     |    3 | Atul     | Agnihotri | Male   | Maharastra  |
|    NULL | NULL     | NULL    | NULL   | NULL     |    5 | Priyanka | Bsumatari | Male   | Mizoram     |
|    NULL | NULL     | NULL    | NULL   | NULL     |    6 | Bikash   | Punchal   | Male   | West Bangal |
+---------+----------+---------+--------+----------+------+----------+-----------+--------+-------------+
15 rows in set (0.06 sec)
```

6) The NATURAL JOIN

Natural Join does not utilize any of the comparison operators. In this type of join, the attributes should have the same name and domain. In Natural Join, there should be at least one common attribute between two relations. It performs selection forming equality on those attributes which appear in both relations and eliminates the duplicate attributes.

Example: if we apply following statement then an NATURAL JOIN will be performed between STUDENT and TEACHER;

Statement:

```
SELECT * FROM student NATURAL JOIN teacher;
```

Output:

```
mysql> SELECT * FROM student NATURAL JOIN teacher;
+--------+---------+----------+---------+----------+-----+----------+-----------+-------------+
| sex    | roll_no | fname    | lname   | state    | tid | t_fname  | t_lname   | address     |
+--------+---------+----------+---------+----------+-----+----------+-----------+-------------+
| Male   |       1 | Robin    | Das     | Assam    |   6 | Bikash   | Punchal   | West Bangal |
| Male   |       1 | Robin    | Das     | Assam    |   5 | Priyanka | Bsumatari | Mizoram     |
| Male   |       1 | Robin    | Das     | Assam    |   3 | Atul     | Agnihotri | Maharastra  |
| Male   |       1 | Robin    | Das     | Assam    |   1 | Monoj    | Pathak    | Assam       |
| Male   |       2 | Dwipen   | Laskar  | Assam    |   6 | Bikash   | Punchal   | West Bangal |
| Male   |       2 | Dwipen   | Laskar  | Assam    |   5 | Priyanka | Bsumatari | Mizoram     |
| Male   |       2 | Dwipen   | Laskar  | Assam    |   3 | Atul     | Agnihotri | Maharastra  |
| Male   |       2 | Dwipen   | Laskar  | Assam    |   1 | Monoj    | Pathak    | Assam       |
| Female |       3 | Asangla  | Sema    | Nagaland |   4 | Praneeta | Bodo      | Assam       |
| Female |       3 | Asangla  | Sema    | Nagaland |   2 | K.V.     | Kanimozhi | Kerala      |
| Female |       4 | Praneeta | Rabha   | Manipur  |   4 | Praneeta | Bodo      | Assam       |
| Female |       4 | Praneeta | Rabha   | Manipur  |   2 | K.V.     | Kanimozhi | Kerala      |
| Female |       5 | Rosy     | Kalita  | Tripura  |   4 | Praneeta | Bodo      | Assam       |
| Female |       5 | Rosy     | Kalita  | Tripura  |   2 | K.V.     | Kanimozhi | Kerala      |
| Female |       6 | Naznin   | Akhtara | Assam    |   4 | Praneeta | Bodo      | Assam       |
| Female |       6 | Naznin   | Akhtara | Assam    |   2 | K.V.     | Kanimozhi | Kerala      |
| Female |       7 | Karabi   | Bora    | Manipur  |   4 | Praneeta | Bodo      | Assam       |
| Female |       7 | Karabi   | Bora    | Manipur  |   2 | K.V.     | Kanimozhi | Kerala      |
| Male   |       8 | Anil     | Gogoi   | Manipur  |   6 | Bikash   | Punchal   | West Bangal |
| Male   |       8 | Anil     | Gogoi   | Manipur  |   5 | Priyanka | Bsumatari | Mizoram     |
| Male   |       8 | Anil     | Gogoi   | Manipur  |   3 | Atul     | Agnihotri | Maharastra  |
| Male   |       8 | Anil     | Gogoi   | Manipur  |   1 | Monoj    | Pathak    | Assam       |
+--------+---------+----------+---------+----------+-----+----------+-----------+-------------+
22 rows in set (0.04 sec)
```