How to Perform an INNER JOIN with a WHERE Clause in MySQL

Authored by

mohammed looti

January 5, 2026

INNER JOIN is a method of combining data from two or more tables in a database based on a common column. Using a WHERE clause in MySQL allows for further filtering of the results, only returning data that meets a particular condition. This means that the INNER JOIN will only return rows where the values in the specified column match between the two tables, and the WHERE clause will further narrow down the results based on the specified condition. This allows for more precise and targeted data retrieval from multiple tables in a database.

You can use the following syntax in MySQL to perform an **INNER JOIN** with a **WHERE** clause:

```
SELECT *
FROM athletes1
INNER JOIN athletes2
ON athletes1.id = athletes2.id
WHERE athletes1.position = 'Guard';
```

This particular example performs an inner join based on matching values in the **id** columns of the **athletes1** and **athletes2** tables and only returns rows where the value in the **position** column of **athletes1** is Guard.

The following example shows how to use this syntax in practice.

Example: How to Do an INNER JOIN with WHERE Clause in MySQL

Suppose we create the following table named **athletes1** that contains information about various basketball players:

```
-- create table
CREATE TABLE athletes1 (
id INT NOT NULL,
position TEXT NOT NULL,
points INT NOT NULL
);
```

```
-- insert rows into table
INSERT INTO athletes1 VALUES (1, 'Guard', 13);
INSERT INTO athletes1 VALUES (2, 'Forward', 25);
INSERT INTO athletes1 VALUES (3, 'Center', 10);
INSERT INTO athletes1 VALUES (4, 'Guard', 28);
INSERT INTO athletes1 VALUES (5, 'Forward', 16);
INSERT INTO athletes1 VALUES (6, 'Center', 20);
```

```
-- view all rows in table
SELECT * FROM athletes1;
```

Output:

```
+----+----------+--------+
| id | position | points |
+----+----------+--------+
| 1 | Guard | 13 |
| 2 | Forward | 25 |
| 3 | Center | 10 |
| 4 | Guard | 28 |
| 5 | Forward | 16 |
| 6 | Center | 20 |
+----+----------+--------+
```

Then suppose we create another table named **athletes2** that contains more information about various basketball players:

```
-- create table
CREATE TABLE athletes2 (
id INT NOT NULL,
team_id INT NOT NULL,
assists INT NOT NULL
);

-- insert rows into table
INSERT INTO athletes2 VALUES (2, 011, 4);
INSERT INTO athletes2 VALUES (5, 012, 2);
INSERT INTO athletes2 VALUES (1, 013, 10);
INSERT INTO athletes2 VALUES (4, 014, 9);
INSERT INTO athletes2 VALUES (6, 015, 13);
INSERT INTO athletes2 VALUES (3, 016, 7);

-- view all rows in table
SELECT * FROM athletes2;
```

Output:

```
+----+---------+---------+
```

| id | team_id | assists |
+----+---------+---------+
| 2 | 11 | 4 |
| 5 | 12 | 2 |
| 1 | 13 | 10 |
| 4 | 14 | 9 |
| 6 | 15 | 13 |
| 3 | 16 | 7 |
+----+---------+---------+

Suppose that we would like to perform an **INNER JOIN** between these two tables based on matching values in the **id** columns but only return rows **WHERE** the value in the **position** column of the first table is Guard.

We can use the following syntax to do so:

```
SELECT athletes1.id, athletes1.position, athletes1.points, athletes2.team_id
FROM athletes1
INNER JOIN athletes2
ON athletes1.id = athletes2.id
WHERE athletes1.position = 'Guard';
```

Output:

+----+----------+--------+---------+
| id | position | points | team_id |
+----+----------+--------+---------+
| 1 | Guard | 13 | 13 |
| 4 | Guard | 28 | 14 |
+----+----------+--------+---------+

Notice that we're able to successfully perform an **INNER JOIN** between these two tables while using the **WHERE** clause.

Also note that you can use the **AND** or **OR** statements to specify multiple conditions in the **WHERE** clause.

For example, we could use the following syntax to perform an inner join but only return rows where the value in the **position** column is Guard *or* the value in the **points** column is greater than 20:

```
SELECT athletes1.id, athletes1.position, athletes1.points, athletes2.team_id
FROM athletes1
INNER JOIN athletes2
ON athletes1.id = athletes2.id
WHERE athletes1.position = 'Guard' OR athletes1.points > 20;
```

Output:

```
+----+----------+--------+---------+
| id | position | points | team_id |
+----+----------+--------+---------+
| 2 | Forward | 25 | 11 |
| 1 | Guard | 13 | 13 |
| 4 | Guard | 28 | 14 |
+----+----------+--------+---------+
```

Notice that we're able to successfully specify two conditions in the **WHERE** clause.

The following tutorials explain how to perform other common tasks in MySQL: