

How to Perform an Inner Join on Multiple Columns in MySQL

Authored by
mohammed loot

January 5, 2026

RECOMMENDED CITATION

mohammed loot (2026). *How to Perform an Inner Join on Multiple Columns in MySQL*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=124631>

Yes, an Inner Join can be performed on multiple columns in MySQL. This is achieved by specifying the columns to join on in the ON clause of the Inner Join statement. By joining on multiple columns, the query will only return rows where the values in all specified columns match in both tables. This allows for more specific and precise data retrieval from multiple related tables in a database.

You can use the following syntax in MySQL to perform an inner join between two tables based on multiple columns:

```
SELECT team, position, points, assists  
FROM athletes1  
INNER JOIN athletes2  
ON ((athletes1.team = athletes2.team_name)  
AND (athletes1.position = athletes2.position_name))
```

This particular example performs an inner join on the tables `athletes1` and `athletes2` based on matching values between the following columns:

team column of `athletes1` and **team_name** column of `athletes2`
position column of `athletes1` and **position_name** column of `athletes2`

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

Example: How to Do Inner Join on Multiple Columns in MySQL

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

```
-- create table  
CREATE TABLE athletes1 (  
team TEXT NOT NULL,  
position TEXT NOT NULL,  
points INT NOT NULL  
);  
  
-- insert rows into table  
INSERT INTO athletes1 VALUES ('Mavs', 'Guard', 13);  
INSERT INTO athletes1 VALUES ('Mavs', 'Forward', 25);  
INSERT INTO athletes1 VALUES ('Mavs', 'Center', 10);  
INSERT INTO athletes1 VALUES ('Spurs', 'Guard', 28);  
INSERT INTO athletes1 VALUES ('Spurs', 'Forward', 16);
```

```
INSERT INTO athletes1 VALUES ('Spurs', 'Center', 20);
```

```
-- view all rows in table
```

```
SELECT * FROM athletes1;
```

Output:

```
+-----+-----+-----+
| team | position | points |
+-----+-----+-----+
| Mavs | Guard | 13 |
| Mavs | Forward | 25 |
| Mavs | Center | 10 |
| Spurs | Guard | 28 |
| Spurs | Forward | 16 |
| Spurs | Center | 20 |
+-----+-----+-----+
```

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

```
-- create table
```

```
CREATE TABLE athletes2 (
  team_name TEXT NOT NULL,
  position_name TEXT NOT NULL,
  assists INT NOT NULL
);
```

```
-- insert rows into table
```

```
INSERT INTO athletes2 VALUES ('Mavs', 'Forward', 4);
INSERT INTO athletes2 VALUES ('Spurs', 'Forward', 2);
INSERT INTO athletes2 VALUES ('Mavs', 'Guard', 10);
INSERT INTO athletes2 VALUES ('Spurs', 'Guard', 9);
INSERT INTO athletes2 VALUES ('Mavs', 'Center', 13);
INSERT INTO athletes2 VALUES ('Spurs', 'Center', 7);
```

```
-- view all rows in table
```

```
SELECT * FROM athletes2;
```

Output:

```

+-----+-----+-----+
| team_name | position_name | assists |
+-----+-----+-----+
| Mavs | Forward | 4 |
| Spurs | Forward | 2 |
| Mavs | Guard | 10 |
| Spurs | Guard | 9 |
| Mavs | Center | 13 |
| Spurs | Center | 7 |
+-----+-----+-----+

```

Suppose that we would like to perform an **INNER JOIN** between these two tables by joining on both the **team** and **position** columns.

We can use the following syntax to do so:

```

SELECT team, position, points, assists
FROM athletes1
INNER JOIN athletes2
ON ((athletes1.team = athletes2.team_name)
AND (athletes1.position = athletes2.position_name))

```

```

+-----+-----+-----+
| team | position | points | assists |
+-----+-----+-----+
| Mavs | Forward | 25 | 4 |
| Spurs | Forward | 16 | 2 |
| Mavs | Guard | 13 | 10 |
| Spurs | Guard | 28 | 9 |
| Mavs | Center | 10 | 13 |
| Spurs | Center | 20 | 7 |
+-----+-----+-----+

```

Notice that we're able to successfully perform an inner join based on the values in the **team** and **position** columns.

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