

How to Perform Case-Sensitive LIKE Searches in MySQL

Authored by
mohammed loot

January 5, 2026

RECOMMENDED CITATION

mohammed loot (2026). *How to Perform Case-Sensitive LIKE Searches in MySQL*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=124683>

MySQL can use a case-sensitive LIKE search by using the **BINARY** operator before the LIKE keyword. This instructs the database to compare the strings in a case-sensitive manner, taking into account uppercase and lowercase letters. This allows for more precise and specific searches, as the database will only return results that exactly match the specified case. Without the **BINARY** operator, LIKE searches are case-insensitive, meaning that uppercase and lowercase letters are treated the same. This feature can be useful for databases that store data in a case-sensitive manner, such as usernames or passwords.

By default, the **LIKE** operator in MySQL is case-insensitive.

However, you can add **BINARY** after the **LIKE** operator to perform a case-sensitive search instead:

```
SELECT * FROM athletes WHERE team LIKE BINARY '%avs';
```

This particular query returns all rows from the **athletes** table where the value in the **team** column ends in the string 'avs', in which the letters must all be lowercase.

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

Example: How to Use Case-Sensitive LIKE Search in MySQL

Suppose we have the following table named **athletes** that contains information about various basketball players:

```
-- create table
```

```
CREATE TABLE athletes (  
id INT PRIMARY KEY,  
team TEXT NOT NULL,  
position TEXT NOT NULL,  
points INT NOT NULL  
);
```

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

```
INSERT INTO athletes VALUES (0001, 'Mavs', 'Guard', 15);  
INSERT INTO athletes VALUES (0002, 'mavs', 'Guard', 22);  
INSERT INTO athletes VALUES (0003, 'MAVS', 'Forward', 36);  
INSERT INTO athletes VALUES (0004, 'Spurs', 'Guard', 18);  
INSERT INTO athletes VALUES (0005, 'spurs', 'Forward', 40);  
INSERT INTO athletes VALUES (0006, 'CAVS', 'Forward', 25);
```

-- view all rows in table

```
SELECT * FROM athletes;
```

Output:

```
+---+-----+-----+-----+
| id | team | position | points |
+---+-----+-----+-----+
| 1 | Mavs | Guard | 15 |
| 2 | mavs | Guard | 22 |
| 3 | MAVS | Forward | 36 |
| 4 | Spurs | Guard | 18 |
| 5 | spurs | Forward | 40 |
| 6 | CAVS | Forward | 25 |
+---+-----+-----+-----+
```

Suppose that we would like to return all rows where the value in the **team** column ends in 'avs' in which each of the letters are lowercase.

If we use a **LIKE** operator by itself, the query will return all rows where the value in the **team** column ends in 'avs', regardless of case:

```
SELECT * FROM athletes WHERE team LIKE '%avs';
```

Output:

```
+---+-----+-----+-----+
| id | team | position | points |
+---+-----+-----+-----+
| 1 | Mavs | Guard | 15 |
| 2 | mavs | Guard | 22 |
| 3 | MAVS | Forward | 36 |
| 6 | CAVS | Forward | 25 |
+---+-----+-----+-----+
```

If we would like to perform a case-sensitive search then we can use **LIKE BINARY** instead:

```
SELECT * FROM athletes WHERE team LIKE BINARY '%avs';
```

```
+----+-----+-----+-----+
| id | team | position | points |
+----+-----+-----+-----+
| 1 | Mavs | Guard | 15 |
| 2 | mavs | Guard | 22 |
+----+-----+-----+-----+
```

Notice that only the rows where the value in the **team** column ends in lowercase 'avs' are returned.

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