# **MySQL IN**

Introduction to the MySQL IN operator

The IN operator allows you to determine if a value matches any value in a list of values. Here’s the syntax of the IN operator:

value IN (value1, value2, value3,...)

The IN operator returns 1 (true) if the value equals any value in the list (value1, value2, value3,…). Otherwise, it returns 0.

In this syntax:

* First, specify the value to test on the left side of the IN operator. The value can be a column or an expression.
* Second, specify a comma-separated list of values to match in the parentheses.

The IN operator is functionally equivalent to the combination of multiple [OR](https://www.mysqltutorial.org/mysql-or/) operators:

value = value1 OR value = value2 OR value = value3 OR ...

The following example returns 1 because 1 is in the list:

SELECT 1 IN (1,2,3);

+--------------+

| 1 IN (1,2,3) |

+--------------+

| 1 |

+--------------+

1 row in set (0.00 sec)

The following example returns 0 because 4 is not in the list:

SELECT 4 IN (1,2,3);

+--------------+

| 4 IN (1,2,3) |

+--------------+

| 0 |

+--------------+

1 row in set (0.00 sec)

In practice, you’ll use the IN operator to form conditions in a WHERE clause of the SELECT, DELETE, and UPDATE statements. Also, you’ll use the IN operator in a query that contains a subquery.

MySQL IN operator and NULL

Generally, the IN operator returns NULL in two cases:

* The value on the left side of the operator is NULL.
* The value doesn’t equal any value in the list and one of values in the list is NULL.

The following example returns NULL because the value of the left side of the IN operator is NULL:

SELECT NULL IN (1,2,3);

+-----------------+

| NULL IN (1,2,3) |

+-----------------+

| NULL |

+-----------------+

1 row in set (0.00 sec)

The following example also returns NULL because the 0 is not equal to any value in the list and the list has one NULL:

SELECT 0 IN (1 , 2, 3, NULL);

+-----------------------+

| 0 IN (1 , 2, 3, NULL) |

+-----------------------+

| NULL |

+-----------------------+

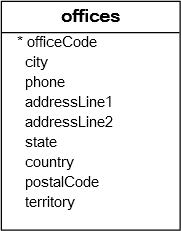
1 row in set (0.00 sec)

The following example also returns NULL because NULL is not equal to any value in the list and the list has one NULL. Note that NULL is not equal to NULL.

SELECT NULL IN (1 , 2, 3, NULL);

MySQL IN operator examples

See the following offices table from the sample database:



The following example uses the IN operator to find the offices located in the USA and France:

SELECT

officeCode,

city,

phone,

country

FROM

offices

WHERE

country IN ('USA' , 'France');

+------------+---------------+-----------------+---------+

| officeCode | city | phone | country |

+------------+---------------+-----------------+---------+

| 1 | San Francisco | +1 650 219 4782 | USA |

| 2 | Boston | +1 215 837 0825 | USA |

| 3 | NYC | +1 212 555 3000 | USA |

| 4 | Paris | +33 14 723 4404 | France |

+------------+---------------+-----------------+---------+

4 rows in set (0.01 sec)

You can also get the same result with the OR operator like this:

SELECT

officeCode,

city,

phone

FROM

offices

WHERE

country = 'USA' OR country = 'France';

In case the list has many values, you need to construct a very long statement with multiple OR operators. Hence, the IN operator allows you to shorten the query and make it more readable.

Summary

* Use the IN operator to check if a value is in a set of values.
* Use the IN operator to form a condition for the WHERE clause.

# **MySQL NOT IN**

## Introduction to the MySQL NOT IN operator

The NOT operator negates the IN operator:

value NOT IN (value1, value2, value2)

The NOT IN operator returns one if the value doesn’t equal any value in the list. Otherwise, it returns 0.

The following example uses the NOT IN operator to check if the number 1 is NOT IN the list (1,2,3):

SELECT 1 NOT IN (1,2,3);

Output:

+------------------+

| 1 NOT IN (1,2,3) |

+------------------+

| 0 |

+------------------+

1 row in set (0.00 sec)

It returns 0 (false) because 1 is NOT IN the list is false.

The following example uses the NOT IN operator to check if 0 is NOT IN the list (1,2,3):

SELECT 0 NOT IN (1,2,3);

Output:

+------------------+

| 0 NOT IN (1,2,3) |

+------------------+

| 1 |

+------------------+

1 row in set (0.00 sec)

The NOT IN operator returns NULL if the value on the left side of the IN operator is NULL. For example:

SELECT NULL NOT IN (1,2,3);

Output:

+---------------------+

| NULL NOT IN (1,2,3) |

+---------------------+

| NULL |

+---------------------+

1 row in set (0.00 sec)

Technically, the NOT IN operator is equivalent to the following:

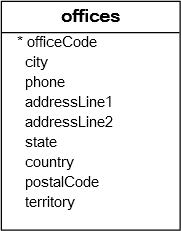
NOT (value = value1 OR value = value2 OR value = valu3)

Or:

value <> value1 AND value <> value2 AND value <> value3

## MySQL NOT IN operator example

We’ll use the offices table from the sample database to illustrates the NOT IN operator:



The following example uses the NOT IN operator to find the offices that do not locate in France and the USA:

SELECT

officeCode,

city,

phone

FROM

offices

WHERE

country NOT IN ('USA' , 'France')

ORDER BY

city;

Output:

+------------+--------+------------------+

| officeCode | city | phone |

+------------+--------+------------------+

| 7 | London | +44 20 7877 2041 |

| 6 | Sydney | +61 2 9264 2451 |

| 5 | Tokyo | +81 33 224 5000 |

+------------+--------+------------------+

3 rows0 in set (0.02 sec)

## Summary

* Use the MySQL NOT IN to check if a value doesn’t match any value in a list.