MySQL WHERE

Here, you will learn how to use the MySQL WHERE clause in the SELECT statement to filter rows from the result set.

## **Introduction to MySQL WHERE clause**

The WHERE clause allows you to specify a search condition for the rows returned by a query. The following shows the syntax of the WHERE clause:

SELECT

select\_list

FROM

table\_name

WHERE

search\_condition;

The search\_condition is a combination of one or more expressions using the logical operator AND, OR and NOT.

In MySQL, a predicate is a Boolean expression that evaluates to TRUE, FALSE, or UNKNOWN.

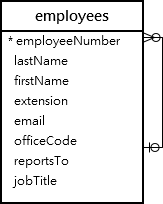
The SELECT statement will include any row that satisfies the search\_condition in the result set.

Besides the SELECT statement, you can use the WHERE clause in the UPDATE or DELETE statement to specify which rows to update or delete.

When executing a SELECT statement with a WHERE clause, MySQL evaluates the WHERE clause after the FROM clause and before the SELECT and ORDER BY clauses:

## **MySQL WHERE clause examples**

We’ll use the employees table from the sample database for the demonstration.



### **1) Using MySQL WHERE clause with equality operator example**

The following query uses the WHERE clause to find all employees whose job titles are Sales Rep:

SELECT

lastname,

firstname,

jobtitle

FROM

employees

WHERE

jobtitle = 'Sales Rep';

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

| lastname | firstname | jobtitle |

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

| Jennings | Leslie | Sales Rep |

| Thompson | Leslie | Sales Rep |

| Firrelli | Julie | Sales Rep |

| Patterson | Steve | Sales Rep |

| Tseng | Foon Yue | Sales Rep |

| Vanauf | George | Sales Rep |

| Bondur | Loui | Sales Rep |

| Hernandez | Gerard | Sales Rep |

| Castillo | Pamela | Sales Rep |

| Bott | Larry | Sales Rep |

| Jones | Barry | Sales Rep |

| Fixter | Andy | Sales Rep |

| Marsh | Peter | Sales Rep |

| King | Tom | Sales Rep |

| Nishi | Mami | Sales Rep |

| Kato | Yoshimi | Sales Rep |

| Gerard | Martin | Sales Rep |

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

17 rows in set (0.00 sec)

In this example, the SELECT statement examines all rows of the employees table and selects only rows whose values in the jobTitle column are Sales Rep.

### **2) Using MySQL WHERE clause with the AND operator**

The following example uses the WHERE clause to find employees whose job titles are Sales Rep and office codes are 1:

SELECT

lastname,

firstname,

jobtitle,

officeCode

FROM

employees

WHERE

jobtitle = 'Sales Rep' AND

officeCode = 1;

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

| lastname | firstname | jobtitle | officeCode |

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

| Jennings | Leslie | Sales Rep | 1 |

| Thompson | Leslie | Sales Rep | 1 |

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

2 rows in set (0.00 sec)

In this example, the expression in the WHERE clause uses the AND operator to combine two conditions:

jobtitle = 'Sales Rep' AND officeCode = 1;

The AND operator evaluates to TRUE only if both expressions evaluate to TRUE. Therefore, the query returns rows whose values in the jobTitle column is Sales Rep and officeCode is 1.

### **3) Using MySQL WHERE clause with OR operator**

This query finds employees whose job title is Sales Rep or employees who locate the office with office code 1:

SELECT

lastName,

firstName,

jobTitle,

officeCode

FROM

employees

WHERE

jobtitle = 'Sales Rep' OR

officeCode = 1

ORDER BY

officeCode ,

jobTitle;

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

| lastName | firstName | jobTitle | officeCode |

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

| Murphy | Diane | President | 1 |

| Bow | Anthony | Sales Manager (NA) | 1 |

| Jennings | Leslie | Sales Rep | 1 |

| Thompson | Leslie | Sales Rep | 1 |

| Firrelli | Jeff | VP Marketing | 1 |

| Patterson | Mary | VP Sales | 1 |

| Firrelli | Julie | Sales Rep | 2 |

| Patterson | Steve | Sales Rep | 2 |

| Tseng | Foon Yue | Sales Rep | 3 |

| Vanauf | George | Sales Rep | 3 |

| Bondur | Loui | Sales Rep | 4 |

| Hernandez | Gerard | Sales Rep | 4 |

| Castillo | Pamela | Sales Rep | 4 |

| Gerard | Martin | Sales Rep | 4 |

| Nishi | Mami | Sales Rep | 5 |

| Kato | Yoshimi | Sales Rep | 5 |

| Fixter | Andy | Sales Rep | 6 |

| Marsh | Peter | Sales Rep | 6 |

| King | Tom | Sales Rep | 6 |

| Bott | Larry | Sales Rep | 7 |

| Jones | Barry | Sales Rep | 7 |

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

21 rows in set (0.00 sec)

The OR operator evaluates to TRUE only if one of the expressions evaluates to TRUE:

jobtitle = 'Sales Rep' OR officeCode = 1

Therefore, the query returns any employee who has the job title Sales Rep or office code 1.

### **4) Using MySQL WHERE clause with the BETWEEN operator example**

The BETWEEN operator returns TRUE if a value is in a range of values:

expression BETWEEN low AND highCode language: SQL (Structured Query Language) (sql)

The following query finds employees who locate in offices whose office code is from 1 to 3:

SELECT

firstName,

lastName,

officeCode

FROM

employees

WHERE

officeCode BETWEEN 1 AND 3

ORDER BY officeCode;

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

| firstName | lastName | officeCode |

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

| Diane | Murphy | 1 |

| Mary | Patterson | 1 |

| Jeff | Firrelli | 1 |

| Anthony | Bow | 1 |

| Leslie | Jennings | 1 |

| Leslie | Thompson | 1 |

| Julie | Firrelli | 2 |

| Steve | Patterson | 2 |

| Foon Yue | Tseng | 3 |

| George | Vanauf | 3 |

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

10 rows in set (0.00 sec)

### **5) Using MySQL WHERE clause with the LIKE operator example**

The LIKE operator evaluates to TRUE if a value matches a specified pattern.

To form a pattern, you use the % and \_ wildcards. The % wildcard matches any string of zero or more characters while the \_ wildcard matches any single character.

The following query finds the employees whose last names end with the string 'son':

SELECT

firstName,

lastName

FROM

employees

WHERE

lastName LIKE '%son'

ORDER BY firstName;

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

| firstName | lastName |

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

| Leslie | Thompson |

| Mary | Patterson |

| Steve | Patterson |

| William | Patterson |

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

4 rows in set (0.00 sec)

### **6) Using MySQL WHERE clause with the IN operator example**

The IN operator returns TRUE if a value matches any value in a list.

value IN (value1, value2,...)

The following example uses the WHERE clause with the IN operator to find employees who locate in the office with office codes 1, 2, 3.

SELECT

firstName,

lastName,

officeCode

FROM

employees

WHERE

officeCode IN (1 , 2, 3)

ORDER BY

officeCode;

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

| firstName | lastName | officeCode |

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

| Diane | Murphy | 1 |

| Mary | Patterson | 1 |

| Jeff | Firrelli | 1 |

| Anthony | Bow | 1 |

| Leslie | Jennings | 1 |

| Leslie | Thompson | 1 |

| Julie | Firrelli | 2 |

| Steve | Patterson | 2 |

| Foon Yue | Tseng | 3 |

| George | Vanauf | 3 |

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

10 rows in set (0.00 sec)

### **7) Using MySQL WHERE clause with the IS NULL operator**

To check if a value is NULL or not, you use the IS NULL operator, not the equal operator (=). The IS NULL operator returns TRUE if a value is NULL.

value IS NULL

In the database world, NULL is a marker that indicates that a value is missing or unknown. And NULL is not equivalent to the number 0 or an empty string.

The following statement uses the WHERE clause with the IS NULL operator to get the rows with the values in the reportsTo column are NULL:

SELECT

lastName,

firstName,

reportsTo

FROM

employees

WHERE

reportsTo IS NULL;

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

| lastName | firstName | reportsTo |

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

| Murphy | Diane | NULL |

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

1 row in set (0.01 sec)

### **8) Using MySQL WHERE clause with comparison operators**

The following table shows the comparison operators that you can use to form the expression in the WHERE clause.

| **Operator** | **Description** |
| --- | --- |
| = | Equal to. You can use it with almost any data type. |
| <> or != | Not equal to |
| < | Less than. You typically use it with numeric and date/time data types. |
| > | Greater than. |
| <= | Less than or equal to |
| >= | Greater than or equal to |

The following query uses the not equal to (<>) operator to find all employees who are not the Sales Rep:

SELECT

lastname,

firstname,

jobtitle

FROM

employees

WHERE

jobtitle <> 'Sales Rep';

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

| lastname | firstname | jobtitle |

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

| Murphy | Diane | President |

| Patterson | Mary | VP Sales |

| Firrelli | Jeff | VP Marketing |

| Patterson | William | Sales Manager (APAC) |

| Bondur | Gerard | Sale Manager (EMEA) |

| Bow | Anthony | Sales Manager (NA) |

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

6 rows in set (0.00 sec)

The following query finds employees whose office code is greater than 5:

SELECT

lastname,

firstname,

officeCode

FROM

employees

WHERE

officecode > 5;

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

| lastname | firstname | officeCode |

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

| Patterson | William | 6 |

| Bott | Larry | 7 |

| Jones | Barry | 7 |

| Fixter | Andy | 6 |

| Marsh | Peter | 6 |

| King | Tom | 6 |

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

6 rows in set (0.00 sec)

The following query returns employees with office code less than or equal to 4 (<=4):

SELECT

lastname,

firstname,

officeCode

FROM

employees

WHERE

officecode <= 4;

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

| lastname | firstname | officeCode |

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

| Murphy | Diane | 1 |

| Patterson | Mary | 1 |

| Firrelli | Jeff | 1 |

| Bondur | Gerard | 4 |

| Bow | Anthony | 1 |

| Jennings | Leslie | 1 |

| Thompson | Leslie | 1 |

| Firrelli | Julie | 2 |

| Patterson | Steve | 2 |

| Tseng | Foon Yue | 3 |

| Vanauf | George | 3 |

| Bondur | Loui | 4 |

| Hernandez | Gerard | 4 |

| Castillo | Pamela | 4 |

| Gerard | Martin | 4 |

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

15 rows in set (0.00 sec)

## **Summary**

* Use the WHERE clause to filter rows by a condition.
* MySQL evaluates the WHERE clause after the FROM clause and before the SELECT and ORDER BY clauses.