# MySQL LIKE

Summary: in this tutorial, you will learn how to use the MySQL LIKE operator to query data based on a specified pattern.

## Introduction to MySQL LIKE operator

The LIKE operator is a logical operator that tests whether a string contains a specified pattern or not. Here is the syntax of the LIKE operator:

expression LIKE pattern ESCAPE escape\_character

Code language: SQL (Structured Query Language) (sql)

The LIKE operator is used in the [WHERE](https://www.mysqltutorial.org/mysql-where/) clause of the [SELECT](https://www.mysqltutorial.org/mysql-select-statement-query-data.aspx) , [DELETE](https://www.mysqltutorial.org/mysql-delete-statement.aspx), and [UPDATE](https://www.mysqltutorial.org/mysql-update-data.aspx) statements to filter data based on patterns.

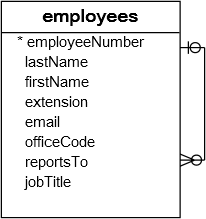
MySQL provides two wildcard characters for constructing patterns: percentage % and underscore \_ .

* The percentage ( % ) wildcard matches any string of zero or more characters.
* The underscore ( \_ ) wildcard matches any single character.

For example, s% matches any string starts with the character s such as sun and six. The se\_ matches any string starts with  se and is followed by any character such as see and sea.

## MySQL LIKE operator examples

Let’s practice with some examples of using the LIKE operator. We will use the following employees table from the [sample database](https://www.mysqltutorial.org/mysql-sample-database.aspx) for the demonstration:



### A) Using MySQL LIKE with the percentage (%) wildcard examples

This example uses the LIKE operator to find employees whose first names start with a:

SELECT

employeeNumber,

lastName,

firstName

FROM

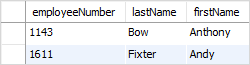
employees

WHERE

firstName LIKE 'a%';

Code language: SQL (Structured Query Language) (sql)

[Try It Out](https://www.mysqltutorial.org/tryit/query/mysql-like/" \l "1)

  
In this example, MySQL scans the whole employees table to find employees whose first names start with the character a and are followed by any number of characters.

This example uses the LIKE operator to find employees whose last names end with on e.g., Patterson, Thompson:

SELECT

employeeNumber,

lastName,

firstName

FROM

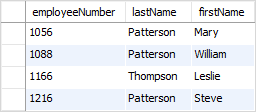
employees

WHERE

lastName LIKE '%on';

Code language: SQL (Structured Query Language) (sql)

[Try It Out](https://www.mysqltutorial.org/tryit/query/mysql-like/" \l "2)



If you know the searched string is embedded inside in the middle of a string, you can use the percentage ( % ) wildcard at the beginning and the end of the pattern.

For example, to find all employees whose last names contain on , you use the following query with the pattern %on%

SELECT

employeeNumber,

lastName,

firstName

FROM

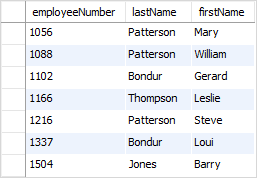
employees

WHERE

lastname LIKE '%on%';

Code language: SQL (Structured Query Language) (sql)

[Try It Out](https://www.mysqltutorial.org/tryit/query/mysql-like/" \l "3)



### B) Using MySQL LIKE with underscore( \_ ) wildcard examples

To find employees whose first names start with  T , end with m, and contain any single character between e.g., Tom , Tim, you use the underscore (\_) wildcard to construct the pattern as follows:

SELECT

employeeNumber,

lastName,

firstName

FROM

employees

WHERE

firstname LIKE 'T\_m';

Code language: SQL (Structured Query Language) (sql)

[Try It Out](https://www.mysqltutorial.org/tryit/query/mysql-like/" \l "4)



### C) Using MySQL LIKE operator with the NOT operator example

The MySQL allows you to combine the NOT operator with the LIKE operator to find a string that does not match a specific pattern.

Suppose you want to search for employees whose last names don’t start with the character B, you can use the NOT LIKE with a pattern as shown in the following query:

SELECT

employeeNumber,

lastName,

firstName

FROM

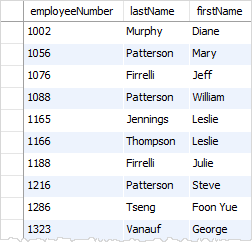
employees

WHERE

lastName NOT LIKE 'B%';

Code language: SQL (Structured Query Language) (sql)

[Try It Out](https://www.mysqltutorial.org/tryit/query/mysql-like/" \l "5)



Note that the pattern is not case sensitive, therefore, the b% or B% pattern returns the same result.

## MySQL LIKE operator with ESCAPE clause

Sometimes the pattern, which you want to match, contains wildcard character e.g., 10%, \_20, etc. In this case, you can use the ESCAPE clause to specify the escape character so that MySQL will interpret the wildcard character as a literal character. If you don’t specify the escape character explicitly, the backslash character \ is the default escape character.

For example, if you want to find products whose product codes contain the string \_20 , you can use the pattern %\\_20% as shown in the following query:

SELECT

productCode,

productName

FROM

products

WHERE

productCode LIKE '%\\_20%';

Code language: SQL (Structured Query Language) (sql)

[Try It Out](https://www.mysqltutorial.org/tryit/query/mysql-like/" \l "6)

Or you can specify a different escape character e.g., $ by using the ESCAPE clause:

SELECT

productCode,

productName

FROM

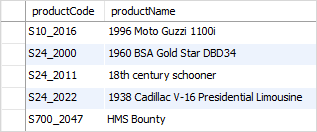
products

WHERE

productCode LIKE '%$\_20%' ESCAPE '$';

Code language: SQL (Structured Query Language) (sql)

[Try It Out](https://www.mysqltutorial.org/tryit/query/mysql-like/" \l "7)



The pattern %$\_20% matches any string that contains the \_20 string.

In this tutorial, you have learned how to use the MySQL LIKE operator to query data based on patterns, which is more flexible than using comparison operators.

https://www.mysqltutorial.org/mysql-like/