SQLITE TUTORIAL

SQLite LIKE

Summary: in this tutorial, you will learn how to query data based on pattern matching using SQLite `LIKE` operator.

Introduction to SQLite LIKE operator

Sometimes, you don’t know exactly the complete keyword that you want to query. For example, you may know that your most favorite song contains the word, `elevator` but you don’t know exactly the name.

To query data (https://www.sqlitetutorial.net/sqlite-select/) based on partial information, you use the `LIKE` operator in the `WHERE` (https://www.sqlitetutorial.net/sqlite-where/) clause of the `SELECT` (https://www.sqlitetutorial.net/sqlite-select/) statement as follows:

```
SELECT
        column_list
FROM
        table_name
WHERE
        column_1 LIKE pattern;
```

Note that you can also use the `LIKE` operator in the `WHERE` clause of other statements such as the `DELETE` and `UPDATE` .

SQLite provides two wildcards for constructing patterns. They are percent sign `%` and underscore `_` :

1. The percent sign `%` wildcard matches any sequence of zero or more characters.
2. The underscore `_` wildcard matches any single character.

The percent sign % wildcard examples

The `s%` pattern that uses the percent sign wildcard (`%`) matches any string that starts with `s` e.g., `son` and `so` .

The `%er` pattern matches any string that ends with `er` like `peter` , `clever` , etc.

And the `%per%` pattern matches any string that contains `per` such as `percent` and `peeper` .

The underscore _ wildcard examples

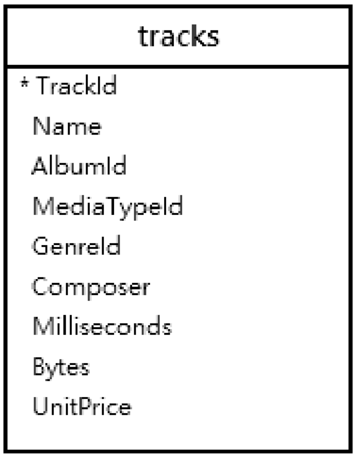
The `h_nt` pattern matches `hunt` , `hint` , etc. The `__pple` pattern matches `topple` , `supple` , `tipple` , etc.
Note that SQLite `LIKE` operator is case-insensitive. It means `"A" LIKE "a"` is true.

However, for Unicode characters that are not in the ASCII ranges, the `LIKE` operator is case sensitive e.g., `"Ä" LIKE "ä"` is false.
In case you want to make `LIKE` operator works case-sensitively, you need to use the following PRAGMA (https://www.sqlite.org/pragma.html#pragma_case_sensitive_like) :

```
PRAGMA case_sensitive_like = true;
```

SQLite LIKE examples

We'll use the table `tracks` in the sample database (https://www.sqlitetutorial.net/sqlite-sample-database/) for the demonstration.



tracks

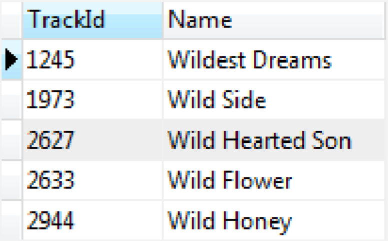
* TrackId
Name
AlbumId
MediaTypeId
GenreId
Composer
Milliseconds
Bytes
UnitPrice

To find the tracks whose names start with the `Wild` literal string, you use the percent sign `%` wildcard at the end of the pattern.

```
SELECT
	trackid,
	name
FROM
```

```
        tracks
WHERE
        name LIKE 'Wild%'
```

Try It ›

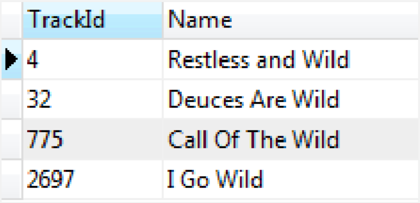


	TrackId	Name
▶	1245	Wildest Dreams
	1973	Wild Side
	2627	Wild Hearted Son
	2633	Wild Flower
	2944	Wild Honey

To find the tracks whose names end with `Wild` word, you use `%` wildcard at the beginning of the pattern.

```
SELECT
        trackid,
        name
FROM
        tracks
WHERE
        name LIKE '%Wild'
```

Try It ›



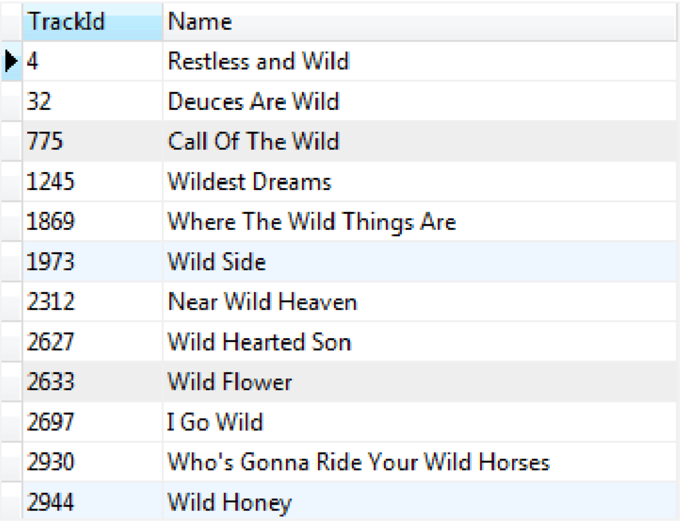
	TrackId	Name
▶	4	Restless and Wild
	32	Deuces Are Wild
	775	Call Of The Wild
	2697	I Go Wild

To find the tracks whose names contain the `Wild` literal string, you use `%` wildcard at the beginning and end of the pattern:

```
SELECT
        trackid,
        name
FROM
        tracks
```

```
WHERE
        name LIKE '%Wild%';
```

Try It ›



TrackId	Name
4	Restless and Wild
32	Deuces Are Wild
775	Call Of The Wild
1245	Wildest Dreams
1869	Where The Wild Things Are
1973	Wild Side
2312	Near Wild Heaven
2627	Wild Hearted Son
2633	Wild Flower
2697	I Go Wild
2930	Who's Gonna Ride Your Wild Horses
2944	Wild Honey

The following statement finds the tracks whose names contain: zero or more characters (`%`), followed by `Br`, followed by a character (`_`), followed by `wn`, and followed by zero or more characters (`%`).

```
SELECT
        trackid,
        name
FROM
        tracks
WHERE
        name LIKE '%Br_wn%';
```

Try It ›

SQLite LIKE with ESCAPE clause

If the pattern that you want to match contains `%` or `_`, you must use an escape character in an optional `ESCAPE` clause as follows:

```
column_1 LIKE pattern ESCAPE expression;
```

When you specify the `ESCAPE` clause, the `LIKE` operator will evaluate the `expression` that follows the `ESCAPE` keyword to a string which consists of a single character, or an escape character.

Then you can use this escape character in the pattern to include literal percent sign (%) or underscore (`_`). The `LIKE` operator evaluates the percent sign (`%`) or underscore (`_`) that follows the escape character as a literal string, not a wildcard character.

Suppose you want to match the string `10%` in a column of a table. However, SQLite interprets the percent symbol `%` as the wildcard character. Therefore, you need to escape this percent symbol `%` using an escape character:

```
column_1 LIKE '%10\%%' ESCAPE '\';
```

In this expression, the `LIKE` operator interprets the first % and last % percent signs as wildcards and the second percent sign as a literal percent symbol.

Note that you can use other characters as the escape character e.g., /, @, $.

Consider the following example:

First, create a table (https://www.sqlitetutorial.net/sqlite-create-table/) `t` that has one column:

```
CREATE TABLE t(
        c TEXT
);
```

Next, insert (https://www.sqlitetutorial.net/sqlite-insert/) some rows into the table `t` :

```
INSERT INTO t(c)
VALUES('10% increase'),
        ('10 times decrease'),
        ('100% vs. last year'),
        ('20% increase next year');
```

Then, query data from the `t` table:

```
SELECT * FROM t;
```

```
c
----------------------
10% increase
10 times decrease
100% vs. last year
20% increase next year
```

Fourth, attempt to find the row whose value in the `c` column contains the `10%` literal string:

```
SELECT c
FROM t
WHERE c LIKE '%10%%';
```

However, it returns rows whose values in the c column contains 10:

```
c
------------------
10% increase
10 times decrease
100% vs. last year
```

Fifth, to get the correct result, you use the `ESCAPE` clause as shown in the following query:

```
SELECT c
FROM t
WHERE c LIKE '%10\%%' ESCAPE '\';
```

Here is the result set:

```
c
------------
10% increase
```

In this tutorial, you have learned how to use SQLite `LIKE` operator to query data based on pattern matching using two wildcard characters percent sign (`%`) and underscore (`_`).