SQLite GLOB

Summary: in this tutorial, you will learn how to use the SQLite `GLOB` operator to determine whether a string matches a specific pattern.

Introduction to the SQLite GLOB operator

The `GLOB` operator is similar to the `LIKE` (https://www.sqlitetutorial.net/sqlite-like/) operator. The `GLOB` operator determines whether a string matches a specific pattern.

Unlike the `LIKE` operator, the `GLOB` operator is **case sensitive** and uses the **UNIX wildcards.** In addition, the `GLOB` patterns do not have escape characters.

The following shows the wildcards used with the `GLOB` operator:

- The asterisk (*) wildcard matches any number of characters.
- The question mark (?) wildcard matches exactly one character.

On top of these wildcards, you can use the list wildcard [] to match one character from a list of characters. For example `[xyz]` match any single x, y, or z character.

The list wildcard also allows a range of characters e.g., [a-z] matches any single lowercase character from a to z. The `[a-zA-Z0-9]` pattern matches any single alphanumeric character, both lowercase, and uppercase.

Besides, you can use the character `^` at the beginning of the list to match any character except for any character in the list. For example, the `[^0-9]` pattern matches any single character except a numeric character.

SQLite GLOB examples

The following statement finds tracks whose names start with the string `Man` . The pattern `Man*` matches any string that starts with `Man` .

```
SELECT
        trackid,
```

```
        name
FROM
        tracks
WHERE
        name GLOB 'Man*';
```

Try It ›

The following statement gets the tracks whose names end with `Man` . The pattern `*Man` matches any string that ends with `Man` .

```
SELECT
        trackid,
        name
FROM
        tracks
WHERE
        name GLOB '*Man';
```

Try It ›

The following query finds the tracks whose names start with any single character (?), followed by the string `ere` and then any number of character (*).

```
SELECT
        trackid,
        name
FROM
        tracks
WHERE
        name GLOB '?ere*';
```

Try It ›

To find the tracks whose names contain numbers, you can use the list wildcard `[0-9]` as follows:

```
SELECT
        trackid,
        name
FROM
        tracks
WHERE
        name GLOB '*[1-9]*';
```

Try It ›

Or to find the tracks whose name does not contain any number, you place the character `^` at the beginning of the list:

```
SELECT
        trackid,
        name
FROM
        tracks
WHERE
        name GLOB '*[^1-9]*';
```

Try It ›

The following statement finds the tracks whose names end with a number.

```
SELECT
        trackid,
        name
FROM
        tracks
WHERE
        name GLOB '*[1-9]';
```

Try It ›

In this tutorial, you have learned how to use SQLite `GLOB` operator to test whether a string matches a specific pattern.