

SQLITE - OPERATORS

http://www.tutorialspoint.com/sqlite/sqlite_operators.htm

Copyright © tutorialspoint.com

What is an Operator in SQLite?

An operator is a reserved word or a character used primarily in an SQLite statement's WHERE clause to perform operations, such as comparisons and arithmetic operations.

Operators are used to specify conditions in an SQLite statement and to serve as conjunctions for multiple conditions in a statement.

- Arithmetic operators
- Comparison operators
- Logical operators
- Bitwise operators

SQLite Arithmetic Operators:

Assume variable a holds 10 and variable b holds 20, then:

[Show Examples](#)

| Operator | Description | Example |
|----------|---|---------------------|
| + | Addition - Adds values on either side of the operator | a + b will give 30 |
| - | Subtraction - Subtracts right hand operand from left hand operand | a - b will give -10 |
| * | Multiplication - Multiplies values on either side of the operator | a * b will give 200 |
| / | Division - Divides left hand operand by right hand operand | b / a will give 2 |
| % | Modulus - Divides left hand operand by right hand operand and returns remainder | b % a will give 0 |

SQLite Comparison Operators:

Assume variable a holds 10 and variable b holds 20, then:

[Show Examples](#)

| Operator | Description | Example |
|----------|--|---------------------|
| == | Checks if the values of two operands are equal or not, if yes then condition becomes true. | a == b is not true. |
| = | Checks if the values of two operands are equal or not, if yes then condition becomes true. | a = b is not true. |
| != | Checks if the values of two operands are equal or not, if values are not equal then | a! = b is true. |

| | | |
|----|---|-----------------------|
| | condition becomes true. | |
| <> | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | $a <> b$ is true. |
| > | Checks if the values of left operand is greater than the value of right operand, if yes then condition becomes true. | $a > b$ is not true. |
| < | Checks if the values of left operand is less than the value of right operand, if yes then condition becomes true. | $a < b$ is true. |
| >= | Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true. | $a >= b$ is not true. |
| <= | Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true. | $a <= b$ is true. |
| !< | Checks if the value of left operand is not less than the value of right operand, if yes then condition becomes true. | $a! < b$ is false. |
| !> | Checks if the value of left operand is not greater than the value of right operand, if yes then condition becomes true. | $a! > b$ is true. |

SQLite Logical Operators:

Here is a list of all the logical operators available in SQLite.

[Show Examples](#)

| Operator | Description |
|----------|--|
| AND | The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause. |
| BETWEEN | The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value. |
| EXISTS | The EXISTS operator is used to search for the presence of a row in a specified table that meets certain criteria. |
| IN | The IN operator is used to compare a value to a list of literal values that have been specified. |
| NOT IN | The negation of IN operator which is used to compare a value to a list of literal values that have been specified. |
| LIKE | The LIKE operator is used to compare a value to similar values using wildcard operators. |
| GLOB | The GLOB operator is used to compare a value to similar values using wildcard operators. Also, GLOB is case sensitive, unlike LIKE. |
| NOT | The NOT operator reverses the meaning of the logical operator with which it is used. Eg. NOT EXISTS, NOT BETWEEN, NOT IN, etc. This is negate operator. |

| | |
|---------|--|
| OR | The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause. |
| IS NULL | The NULL operator is used to compare a value with a NULL value. |
| IS | The IS operator work like = |
| IS NOT | The IS operator work like != |
| | Adds two different strings and make new one. |
| UNIQUE | The UNIQUE operator searches every row of a specified table for uniqueness <i>noduplicates</i> . |

SQLite Bitwise Operators:

Bitwise operator works on bits and perform bit-by-bit operation. The truth table for & and | is as follows:

| p | q | p & q | p q |
|---|---|-------|-------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |

Assume if A = 60; and B = 13; now in binary format, they will be as follows:

```
A = 0011 1100
B = 0000 1101
-----
A&B = 0000 1100
A|B = 0011 1101
~A  = 1100 0011
```

The Bitwise operators supported by SQLite language are listed in the following table. Assume variable A holds 60 and variable B holds 13, then:

[Show Examples](#)

| Operator | Description | Example |
|----------|--|--|
| & | Binary AND Operator copies a bit to the result if it exists in both operands. | A & B will give 12 which is 0000 1100 |
| | Binary OR Operator copies a bit if it exists in either operand. | A B will give 61 which is 0011 1101 |
| ~ | Binary Ones Complement Operator is unary and has the effect of 'flipping' bits. | A will give -61 which is 1100 0011 in 2's complement form due to a signed binary number. |
| << | Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right | A << 2 will give 240 which is 1111 0000 |

operand.

>>

Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.

A >> 2 will give 15 which is 0000 1111

Loading [MathJax]/jax/output/HTML-CSS/jax.js