Operators of SQLite

Victor Harris

CIS-150

07/06/2021

Operators of SQLite

The word ‘operator’ refers to the words and characters used in SQLite statements to construct logical expressions that can be used to control, organize, and manipulate data elements based on the values contained within them. Through the use of these operators, statements that utilize the WHERE clause can perform comparisons, as well as arithmetical operations based on specific conditions. In SQLite, there are distinct type of operators: Arithemetic, Comparison, Logical and Bitwise each serving a unique role utilizing the values stored within a database.

**Arithmetic Operators**

Arithmetical operations in SQLite can be performed using the following operators to construct arguments based on numeric values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operator | Character/Symbol | Description | Example INPUT  “SELECT \* | Resulting  Argument |
| Addition | + | The addition operator is used to add values, and returns a sum of those values | 1+1  2+2  100+10 | 2  4  110 |
| Subtraction | - | The subtraction operator effectively flips the value of the value on the right, and then adds it to the value on the left. | 1 - 1  2 - 2  100 - (-10)  100 - 10 | 0  0  110  90 |
| Multiplication | \* | The Multiplication operator multiplies two values | 1 \* 1  2 \* 2  100 \* 10 | 1  4  1000 |
| Division | / | The Division operator divides the value on the left by the value on the right | 1 / 1  2 / 2  100 / 10 | 1  1  10 |
| Modulus | % | Divides the value on the left and by the value on the right, and returns the remainder | 1 / 1  2 / 2  100 / 10  107 / 10 | 0  0  0  7 |

**Comparison Operators**

Comparison operators are used to comparing arguments and values against each other, and returns either a TRUE, FALSE, or NULL value in return:

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Description | EXAMPLE COMPARISONS | EXAMPLE RETURN VALUES |
| == | Evaluates two values and returns whether they are equal. | 1 == 1  2 == 1  100 == 100  100 == -100 | TRUE  FALSE  TRUE  FALSE |
| = | Also evaluates whether values are equal or not. | 1 = 1  2 = 1  100 = 100  100 = -100 | TRUE  FALSE  TRUE  FALSE |
| != | The != operator check to see if values are NOT Equal | 1 <> 1  2 <> 1  100 <> 100  100 <> -100 | FALSE  TRUE  FALSE  TRUE |
| <> | The <> operator check to see if values are NOT Equal | 1 <> 1  2 <> 1  100 <> 100  100 <> -100 | FALSE  TRUE  FALSE  TRUE |
| > | The > operator compares two values, and returns true when the left operand is greater in value than the right operand | 1 > 1  2 > 1  100 > 100  100 > -100 | FALSE  TRUE  FALSE  TRUE |
| < | The > operator compares two values, and returns true when the left operand is a lower value than the right operand | 1 < 1  2 < 1  100 < 100  100 < -100 | FALSE  FALSE  FALSE  FALSE |
| >= | The >= operator compares two values, and returns true when the left operand is greater or equal in value to the right operand | 1 >= 1  2 >= 1  100 >= 100  100 >= -100 | TRUE  TRUE  TRUE  TRUE |
| <= | The <= operator compares two values, and returns true when the left operand is less or equal than the right operand | 1 <= 1  2 <= 1  100 <= 100  100 <= -100 | TRUE  FALSE  TRUE  FALSE |
| !< | Checks to see if the left operand is NOT less than the right operand | 1 !< 1  2 !< 1  100 !< 100  100 !< -100 | TRUE  TRUE  TRUE  TRUE |
| !> | Checks to see if the left operand is NOT greater than the right operand | 1 !> 1  2 !> 1  100 !> 100  100 !> -100 | TRUE  TRUE  TRUE  FALSE |

**Logical Operators**

Logical Operators are used to modify operations and arguments used in SQLite statements, they can be used to further specify conditions, as well as compare values when using the WHERE clause:

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Description | SQLite Example | Example Result |
| AND | Used to specify multiple conditions | SELECT \* FROM TABLE WHERE [value1 > value2] AND [value2 != 0] | Returns all records in which value 1 is greater than value 2, as long as value 2 does not have a value of 0 |
| BETWEEN | Used to check for records that have values within a range of values. The BETWEEN operator is inclusive, and will include values that are equal in value to the argument being used | DELETE \* FROM TABLE WHERE [value1] BETWEEN 10 AND 20 | Deletes records of a table where value1 is greater or equal to 10, but no greater than 20. |
| EXISTS | Searches for a record of a table that meets the specified criteria | SELECT \* FROM TABLE WHERE EXISTS (SELECT \* FROM TABLE2 WHERE TABLE.field == TABLE2.field) | Returns all records from when TABLE.field is equal to TABLE2.field. |
| IN / NOT IN | Compares a value to a list of specified literal values, evaluates true when values match the given expression | SELECT \* FROM TABLE WHERE field IN (“Apple”) | Returns all records of TABLE where the value of field is “Apple” |
| LIKE | Compares a value to values that are similar using a wildcard operator. | SELECT \* FROM TABLE WHERE name LIKE “J%” | Returns all records where the value of name begins with “J” |
| GLOB | Similar to LIKE, is case sensitive | SELECT \* FROM TABLE WHERE name GLOB “%S” | Returns all record where the value of the name field ends with “S” |
| NOT | NEGATES other Logical Operators | SELECT \* FROM TABLE WHERE field IS NOT 2 | Returns all records from TABLE that the field value is 2 |
| OR | Used to specify multiple conditions, and returns true if either condition is met | SELECT \* FROM TABLE WHERE field1 == 1  OR field2 == 1 | Returns all records that have a field1 or field2 value of 1 |
| IS NULL | Compares values to NULL | SELECT \* FROM TABLE WHERE field IS NULL | Returns all records that have a NULL value in field |
| IS | Functions like the ==/ = operator | SELECT \* FROM TABLE WHERE (field) IS 0 | Returns all records of TABLE that have a field value of 0 |
| || | Concatenates two string values | SELECT first || “ “ || last AS contact | Returns the values of first and last with a space between them as ‘contact’ |

**Bitwise Operators**

Bitwise operators are used to perform bit-by-bit operations. Although they are not part of the SQL standard, the following bitwise operators are available in SQLite:

|  |  |
| --- | --- |
| Operator | Function |
| & | Copies the bit if it exists in both the left and right operands |
| | | Copies the bit if it exists in either operands |
| << | Moves the left operand by the right operand number of bits |
| >> | Same as left shift, but to the right |

Sources

Kreibich, Jay. *Using SQLite: Small. Fast. Reliable. Choose Any Three.* 1st ed., O’Reilly Media, 2010.

“SQLite: LIKE Condition.” *TechOnTheNet*, www.techonthenet.com/sqlite/like.php. Accessed 6 July 2021.

“SQLite Operators - Tutlane.” *Tutlane.Com*, www.tutlane.com/tutorial/sqlite/sqlite-operators. Accessed 6 July 2021.

“SQLite Tutorial - An Easy Way to Master SQLite Fast.” *SQLite Tutorial*, 30 Apr. 2021, www.sqlitetutorial.net.

*SQLite Tutorial - Tutorialspoint*. www.tutorialspoint.com/sqlite. Accessed 6 July 2021