<https://unetway.com/tutorial/sqlite>

## Операторы SQLite

Все операторы SQLite начинаются с любых ключевых слов, таких как SELECT, INSERT, UPDATE, DELETE, ALTER, DROP и т. д., И все инструкции заканчиваются точкой с запятой (;).

### **SQLite ANALYZE**

ANALYZE;

**or**

ANALYZE database\_name;

**or**

ANALYZE database\_name.table\_name;

### **SQLite AND / OR**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** CONDITION-1 {**AND**|**OR**} CONDITION-2;

### **SQLite ALTER TABLE**

**ALTER** **TABLE** table\_name **ADD** **COLUMN** column\_def...;

### **SQLite ALTER TABLE (Rename)**

**ALTER** **TABLE** table\_name RENAME **TO** new\_table\_name;

### **SQLite ATTACH DATABASE**

ATTACH DATABASE 'DatabaseName' **As** 'Alias-Name';

### **SQLite BEGIN TRANSACTION**

**BEGIN**;

or

**BEGIN** EXCLUSIVE **TRANSACTION**;

### **SQLite BETWEEN**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** column\_name BETWEEN val-1 **AND** val-2;

### **SQLite COMMIT**

**COMMIT**;

### **SQLite CREATE INDEX**

**CREATE** INDEX index\_name

**ON** table\_name ( column\_name **COLLATE** NOCASE );

### **SQLite CREATE UNIQUE INDEX**

**CREATE** **UNIQUE** INDEX index\_name

**ON** table\_name ( column1, column2,...columnN);

### **SQLite CREATE TABLE**

**CREATE** **TABLE** table\_name(

column1 datatype,

column2 datatype,

column3 datatype,

.....

columnN datatype,

**PRIMARY** **KEY**( one **or** more columns )

);

### **SQLite CREATE TRIGGER**

**CREATE** **TRIGGER** database\_name.trigger\_name

**BEFORE** **INSERT** **ON** table\_name **FOR** **EACH** **ROW**

**BEGIN**

stmt1;

stmt2;

....

**END**;

### **SQLite CREATE VIEW**

**CREATE** **VIEW** database\_name.view\_name **AS**

**SELECT** statement....;

### **SQLite CREATE VIRTUAL TABLE**

**CREATE** VIRTUAL **TABLE** database\_name.table\_name **USING** weblog( access.log );

or

**CREATE** VIRTUAL **TABLE** database\_name.table\_name **USING** fts3( );

### **SQLite COMMIT TRANSACTION**

**COMMIT**;

### **SQLite COUNT**

**SELECT** **COUNT**(column\_name)

**FROM** table\_name

**WHERE** CONDITION;

### **SQLite DELETE**

**DELETE** **FROM** table\_name

**WHERE** {CONDITION};

### **SQLite DETACH DATABASE**

DETACH DATABASE 'Alias-Name';

### **SQLite DISTINCT**

**SELECT** **DISTINCT** column1, column2....columnN

**FROM** table\_name;

### **SQLite DROP INDEX**

**DROP** INDEX database\_name.index\_name;

### **SQLite DROP TABLE**

**DROP** **TABLE** database\_name.table\_name;

### **SQLite DROP VIEW**

**DROP** INDEX database\_name.view\_name;

### **SQLite DROP TRIGGER**

**DROP** INDEX database\_name.trigger\_name;

### **SQLite EXISTS**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** column\_name **EXISTS** (**SELECT** \* **FROM** table\_name );

### **SQLite EXPLAIN**

EXPLAIN **INSERT** statement...;

or

EXPLAIN QUERY PLAN **SELECT** statement...;

### **SQLite GLOB**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** column\_name GLOB { PATTERN };

### **SQLite GROUP BY**

**SELECT** **SUM**(column\_name)

**FROM** table\_name

**WHERE** CONDITION

**GROUP** **BY** column\_name;

### **SQLite HAVING**

**SELECT** **SUM**(column\_name)

**FROM** table\_name

**WHERE** CONDITION

**GROUP** **BY** column\_name

**HAVING** (arithematic function condition);

### **SQLite INSERT INTO**

**INSERT** **INTO** table\_name( column1, column2....columnN)

**VALUES** ( value1, value2....valueN);

### **SQLite IN**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** column\_name **IN** (val-1, val-2,...val-N);

### **SQLite Like**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** column\_name **LIKE** { PATTERN };

### **SQLite NOT IN**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** column\_name **NOT** **IN** (val-1, val-2,...val-N);

### **SQLite ORDER BY**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** CONDITION

**ORDER** **BY** column\_name {**ASC**|**DESC**};

### **SQLite PRAGMA**

**PRAGMA** pragma\_name;

For example:

**PRAGMA** page\_size;

**PRAGMA** cache\_size = 1024;

**PRAGMA** table\_info(table\_name);

### **SQLite RELEASE SAVEPOINT**

RELEASE savepoint\_name;

### **SQLite REINDEX**

REINDEX collation\_name;

REINDEX database\_name.index\_name;

REINDEX database\_name.table\_name;

### **SQLite ROLLBACK**

**ROLLBACK**;

or

**ROLLBACK** **TO** SAVEPOINT savepoint\_name;

### **SQLite SAVEPOINT**

SAVEPOINT savepoint\_name;

### **SQLite SELECT Statement**

**SELECT** column1, column2....columnN

**FROM** table\_name;

### **SQLite UPDATE**

**UPDATE** table\_name

**SET** column1 = value1, column2 = value2....columnN=valueN

[ **WHERE** CONDITION ];

### **SQLite VACUUM**

VACUUM;

### **SQLite WHERE**

**SELECT** column1, column2....columnN

**FROM** table\_name

**WHERE** CONDITION;