

SQLITE - COMMANDS

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

Copyright © tutorialspoint.com

This chapter will take you through simple and useful commands used by SQLite programmers. These commands are called SQLite dot commands and exception with these commands is that they should not be terminated by a semi-colon ; .

Let's start with typing a simple **sqlite3** command at command prompt which will provide you SQLite command prompt where you will issue various SQLite commands.

```
$sqlite3
SQLite version 3.3.6
Enter ".help" for instructions
sqlite>
```

For a listing of the available dot commands, you can enter ".help" at any time. For example:

```
sqlite>.help
```

Above command will display a list of various important SQLite dot commands, which are as follows:

Command	Description
.backup ?DB? FILE	Backup DB <i>default " main "</i> to FILE
.bail ON OFF	Stop after hitting an error. Default OFF
.databases	List names and files of attached databases
.dump ?TABLE?	Dump the database in an SQL text format. If TABLE specified, only dump tables matching LIKE pattern TABLE.
.echo ON OFF	Turn command echo on or off
.exit	Exit SQLite prompt
.explain ON OFF	Turn output mode suitable for EXPLAIN on or off. With no args, it turns EXPLAIN on.
.headers ON OFF	Turn display of headers on or off
.help	Show this message
.import FILE TABLE	Import data from FILE into TABLE
.indices ?TABLE?	Show names of all indices. If TABLE specified, only show indices for tables matching LIKE pattern TABLE.
.load FILE ?ENTRY?	Load an extension library
.log FILE off	Turn logging on or off. FILE can be stderr/stdout
.mode MODE	Set output mode where MODE is one of: <ul style="list-style-type: none">• csv Comma-separated values• column Left-aligned columns.• html HTML <table> code• insert SQL insert statements for TABLE

- **line** One value per line
- **list** Values delimited by .separator string
- **tabs** Tab-separated values
- **tcl** TCL list elements

<code>.nullvalue STRING</code>	Print STRING in place of NULL values
<code>.output FILENAME</code>	Send output to FILENAME
<code>.output stdout</code>	Send output to the screen
<code>.print STRING...</code>	Print literal STRING
<code>.prompt MAIN CONTINUE</code>	Replace the standard prompts
<code>.quit</code>	Exit SQLite prompt
<code>.read FILENAME</code>	Execute SQL in FILENAME
<code>.schema ?TABLE?</code>	Show the CREATE statements. If TABLE specified, only show tables matching LIKE pattern TABLE.
<code>.separator STRING</code>	Change separator used by output mode and .import
<code>.show</code>	Show the current values for various settings
<code>.stats ON OFF</code>	Turn stats on or off
<code>.tables ?PATTERN?</code>	List names of tables matching a LIKE pattern
<code>.timeout MS</code>	Try opening locked tables for MS milliseconds
<code>.width NUM NUM</code>	Set column widths for "column" mode
<code>.timer ON OFF</code>	Turn the CPU timer measurement on or off

Let's try **.show** command to see default setting for your SQLite command prompt.

```
sqlite>.show
  echo: off
  explain: off
  headers: off
  mode: column
nullvalue: ""
  output: stdout
separator: "| "
  width:
sqlite>
```

Make sure there is no space in between sqlite> prompt and dot command, otherwise it will not work.

Formatting output

You can use the following sequence of dot commands to format your output the way I have listed down in this tutorial:

```
sqlite>.header on
sqlite>.mode column
sqlite>.timer on
```

```
sqlite>
```

Above setting will produce the output in the following format:

ID	NAME	AGE	ADDRESS	SALARY
1	Paul	32	California	20000.0
2	Allen	25	Texas	15000.0
3	Teddy	23	Norway	20000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0

CPU Time: user 0.000000 sys 0.000000

The sqlite_master Table

The master table holds the key information about your database tables and it is called **sqlite_master**. You can see its schema as follows:

```
sqlite>.schema sqlite_master
```

This will produce the following result:

```
CREATE TABLE sqlite_master (  
  type text,  
  name text,  
  tbl_name text,  
  rootpage integer,  
  sql text  
);
```

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