

xenial (1) [sqlite3.1.gz](#)

Provided by: [sqlite3_3.11.0-1ubuntu1_i386](#) 

NAME

sqlite3 - A command line interface for SQLite version 3

SYNOPSIS

sqlite3 [options] [databasefile] [SQL]

SUMMARY

sqlite3 is a terminal-based front-end to the SQLite library that can evaluate queries interactively and display the results in multiple formats. **sqlite3** can also be used within shell scripts and other applications to provide batch processing features.

DESCRIPTION

To start a **sqlite3** interactive session, invoke the **sqlite3** command and optionally provide the name of a database file. If the database file does not exist, it will be created. If the database file does exist, it will be opened.

For example, to create a new database file named "mydata.db", create a table named "memos" and insert a couple of records into that table:

```
$ sqlite3 mydata.db
SQLite version 3.8.8
Enter ".help" for instructions
sqlite> create table memos(text, priority INTEGER);
sqlite> insert into memos values('deliver project description', 10);
sqlite> insert into memos values('lunch with Christine', 100);
sqlite> select * from memos;
deliver project description|10
lunch with Christine|100
sqlite>
```

If no database name is supplied, the ATTACH sql command can be used to attach to existing or create new database files. ATTACH can also be used to attach to multiple databases within the same interactive session. This is useful for migrating data between databases, possibly changing the schema along the way.

Optionally, a SQL statement or set of SQL statements can be supplied as a single argument. Multiple statements should be separated by semi-colons.

For example:

```
$ sqlite3 -line mydata.db 'select * from memos where priority > 20;'
text = lunch with Christine
priority = 100
```

SQLITE META-COMMANDS

The interactive interpreter offers a set of meta-commands that can be used to control the output format, examine the currently attached database files, or perform administrative operations upon the attached databases (such as rebuilding indices). Meta-commands are always prefixed with a dot (.).

A list of available meta-commands can be viewed at any time by issuing the '.help' command. For example:

```

sqlite> .help
.backup ?DB? FILE    Backup DB (default "main") to FILE
.bail on|off        Stop after hitting an error.  Default OFF
.clone NEWDB        Clone data into NEWDB from the existing database
.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
.eqp on|off         Enable or disable automatic EXPLAIN QUERY PLAN
.exit              Exit this program
.explain ?on|off?   Turn output mode suitable for EXPLAIN on or off.
                    With no args, it turns EXPLAIN on.
.fullschema        Show schema and the content of sqlite_stat tables
.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 ?TABLE? Set output mode where MODE is one of:
                    csv    Comma-separated values
                    column Left-aligned columns.  (See .width)
                    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
.nullvalue STRING  Use STRING in place of NULL values
.once FILENAME     Output for the next SQL command only to FILENAME
.open ?FILENAME?   Close existing database and reopen FILENAME
.output ?FILENAME? Send output to FILENAME or stdout
.print STRING...   Print literal STRING
.prompt MAIN CONTINUE Replace the standard prompts
.quit             Exit this program
.read FILENAME     Execute SQL in FILENAME
.restore ?DB? FILE Restore content of DB (default "main") from FILE
.save FILE         Write in-memory database into FILE
.schema ?TABLE?    Show the CREATE statements
                    If TABLE specified, only show tables matching
                    LIKE pattern TABLE.
.separator STRING ?NL? Change separator used by output mode and .import
                    NL is the end-of-line mark for CSV
.shell CMD ARGS... Run CMD ARGS... in a system shell
.show             Show the current values for various settings
.stats on|off     Turn stats on or off
.system CMD ARGS... Run CMD ARGS... in a system shell
.tables ?TABLE?    List names of tables
                    If TABLE specified, only list tables matching
                    LIKE pattern TABLE.
.timeout MS       Try opening locked tables for MS milliseconds
.timer on|off     Turn SQL timer on or off
.trace FILE|off   Output each SQL statement as it is run
.vfsname ?AUX?    Print the name of the VFS stack
.width NUM1 NUM2 ... Set column widths for "column" mode
                    Negative values right-justify

```

sqlite>

OPTIONS

sqlite3 has the following options:

-bail Stop after hitting an error.

-batch Force batch I/O.

-column

Query results will be displayed in a table like form, using whitespace characters to separate the columns and align the output.

-cmd command

run command before reading stdin

-csv Set output mode to CSV (comma separated values).

-echo Print commands before execution.

-init file

Read and execute commands from file, which can contain a mix of SQL statements and meta-commands.

-[no]header

Turn headers on or off.

-help Show help on options and exit.

-html Query results will be output as simple HTML tables.

-interactive

Force interactive I/O.

-line Query results will be displayed with one value per line, rows separated by a blank line. Designed to be easily parsed by scripts or other programs

-list Query results will be displayed with the separator (|, by default) character between each field value. The default.

-mmap N

Set default mmap size to N

-nullvalue string

Set string used to represent NULL values. Default is " (empty string).

-separator separator

Set output field separator. Default is '|'.

-stats Print memory stats before each finalize.

-version

Show SQLite version.

-vfs name

Use name as the default VFS.

INIT FILE

sqlite3 reads an initialization file to set the configuration of the interactive environment. Throughout initialization, any previously specified setting can be overridden. The sequence of initialization is as follows:

o The default configuration is established as follows:

```
mode          = LIST
separator     = "|"
main prompt   = "sqlite> "
continue prompt = " ...> "
```

o If the file `~/.sqliterc` exists, it is processed first. can be found in the user's home directory, it is read and processed. It should generally only contain meta-commands.

o If the `-init` option is present, the specified file is processed.

o All other command line options are processed.

SEE ALSO

<http://www.sqlite.org/cli.html>

The `sqlite3-doc` package.

AUTHOR

This manual page was originally written by Andreas Rottmann <rotty@debian.org>, for the Debian GNU/Linux system (but may be used by others). It was subsequently revised by Bill Bumgarner <bbum@mac.com> and further updated by Laszlo Boszormenyi <gcs@debian.hu> .

Fri Oct 31 10:41:31 EDT 2014

SQLITE3(1)