

SQLite dot commands, you can enter ".help" at any time. For example:

```
sqlite> .help
.backup ?DB? FILE      Backup DB (default "main") to FILE
.bail on|off           Stop after hitting an error.  Default OFF
.binary on|off         Turn binary output on or off.  Default OFF
.clone NEWDB           Clone data into NEWDB from the existing database
.databases             List names and files of attached databases
.dbinfo ?DB?          Show status information about the database
.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
.indexes ?TABLE?       Show names of all indexes
                        If TABLE specified, only show indexes for tables
                        matching LIKE pattern TABLE.
.limit ?LIMIT? ?VAL?   Display or change the value of an SQLITE_LIMIT
.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:
                        ascii    Columns/rows delimited by 0x1F and 0x1E
                        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 strings
                        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
.scanstats on|off      Turn sqlite3_stmt_scanstatus() metrics on or off
.schema ?TABLE?        Show the CREATE statements
                        If TABLE specified, only show tables matching
                        LIKE pattern TABLE.
.separator COL ?ROW?   Change the column separator and optionally the row
                        separator for both the output mode and .import
.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>
```

Anatomy of a SELECT command:

SELECT

 <column,...> *(use * for all columns)*

FROM <table>

WHERE <condition>

ORDER BY <column,...>

Anatomy of a SELECT JOIN command:

SELECT

 <column,...>

FROM <table1>

 JOIN <table2>

 ON <table1.column> = <table2.column>

 ...

WHERE <condition>

ORDER BY <column,...>

Anatomy of a CREATE TABLE command:

CREATE TABLE <table name> (

 <column name> <data type> <constraint>,

 ...

);

Anatomy of an INSERT command:

INSERT INTO <table name> (column,...)

 VALUES (values,...);