# mmsql 1.2 – SQLite manager for the Colour Maximite 2

mmsql is a command line tool, written in MMBasic, to look into, change, check, repair, import and export SQLite databases on the Colour Maximite 2. It works on the VGA/HDMI screen and on a serial terminal (e.g. Tera Term) at the same time, and it follows the commands of the `sqlite3` shell, so it feels familiar to anyone who has used SQLite on a PC.

© 2026 Matthias Hollatz – MIT License (see `LICENSE.txt`)

The full description of all commands, options and limits is in the user manual: `MANUAL_mmsql.pdf` (German: `HANDBUCH_mmsql.pdf`). New to SQL? Start with the tutorial (separate download `mmsql_tutorial.zip`).

## Requirements

- Colour Maximite 2, generation 1, 2 or 3
- MMBasic **8.00.00b1** or later (8.00.00b2 recommended)
- The `sql` folder from the firmware package in the root of the SD card (`A:/sql/sql.mod`)
- For firmware 7.00.00RC6 use **mmsql 1.0.1** (separate download)

## Installation

Copy these files into one folder, e.g. `A:/SQL/` (with XMODEM: `xmodem receive "mmsql.bas"` and so on):

| File | Purpose |
|---|---|
| `mmsql.bas` | the program |
| `mmsql_en.hlp` | English help (default) |
| `mmsql_de.hlp` | German help (`.set helpfile mmsql_de.hlp`) |
| `LICENSE.txt` | MIT License |

mmsql creates two more files there: `mmsql.cfg` (options) and `mmsql.his` (command history).

## Quick start

```
RUN "mmsql", "A:/sql/computers.db"
.tables
.columns computer
.browse computer WHERE year < 1982 ORDER BY year
SELECT maker, count(*) FROM computer GROUP BY maker;
.quit
```

Commands start with a dot. Anything else is run as SQL. `.help` lists all commands, `.help name` shows the details of one (e.g. `.help import`), and `.help keys`, `.help sql` and `.help utf8` explain the editor, SQL and UTF-8.

## New in 1.2

- **A real editor at the prompt**: SQL over up to 20 lines with a status line – F3 new line, F2 run. Pasted SQL from the PC is recognised and kept together.
- **One rule for input**: one line runs with Enter; several lines are a block, each statement ends with `;`.
- **Long statements and queries** over 255 characters, with results – at the prompt, in blocks and in scripts.
- **History** of statements of any length, also after a restart; PgUp/PgDn.
- **Scripts as in sqlite3**: `.read` takes statements over several lines, shows query results and knows dot commands.
- **No more 20-column limit**: `.import` without limits for columns and long fields, `.insert` up to 64 columns.
- **Database in memory** (`:memory:`) – mmsql starts without a file.
- **New output**: `.mode markdown`, `.mode json`, `.width`, `.print`, `.set blob`.
- **`.dump`** keeps the real type of every value (BLOB, text, 2.0 as REAL, very large integers).
- **More commands**: `.restore`, `.version`, search patterns for `.tables`, `.schema`, `.indexes`, `.triggers`; `.set screen`, `.set width 0`.
- Shorter prompt `[retro.db]>`; terminal and screen show exactly the same.

**Note:** `.read` now needs a `;` at the end of every statement, as `sqlite3` does. Old scripts with one statement per line without `;` must get a `;` at the end of each line. Dumps from mmsql have always had it.

All changes: `CHANGES.txt`.

## Commands

| Group | Commands |
|---|---|
| Database | `.open` `.close` `.info` `.backup` `.restore` `.vacuum` `.files` `.kill` |
| Look | `.tables` `.indexes` `.triggers` `.schema` `.columns` `.browse` `.databases` |
| Data | `.insert` `.edit` `.delete` `.import` `.read` |
| Structure | `.create` `.alter` `.reindex` |
| Check | `.check` `.fkcheck` `.fk` |
| Output | `.mode` `.headers` `.separator` `.nullvalue` `.width` `.once` `.output` `.dump` |
| Settings | `.set` `.timer` `.bail` `.echo` `.print` `.cls` `.version` `.help` `.quit` |

## Still there from 1.1

- **Browse** tables page by page, with `WHERE` and `ORDER BY`; **edit** a row field by field (`.edit`, `.insert`, `.delete`, `.create`, `.alter`), with a question before anything is changed.
- **Export** as in the sqlite3 shell: `.mode csv|list|insert|line|table|markdown|json`, `.headers`, `.separator`, `.once`, `.output`, `.dump`. CSV follows RFC 4180 with CRLF line ends.
- **Long texts and BLOBs** over 255 characters in full (option `maxtext`, default 32 KB).

- **Ctrl-C** stops a running SQL statement and mmsql goes on; it tells you exactly what SQLite has undone.
- **Transactions** are watched: the prompt shows a star while one is open (`[club.db]*>`), and `.quit`, `.close` and `.open` ask whether to commit or roll back.
- **Check and repair**: `.check [full]`, `.fkcheck`, `.reindex`, `.vacuum`, `.backup`.
- **Several databases** with `ATTACH 'other.db' AS o;`.

Tested against SQLite on a PC: CSV, insert and dump export and the round trips CSV → database → CSV and dump → database → dump, including quotes, line breaks, tabs, UTF-8, 64-bit integers, long texts and BLOBs. An automatic test suite for the CMM2 is available as a separate download (`mmsql_testsuite.zip`).

## Good to know

- **One line in the editor is at most 255 characters** (an MMBasic string). Longer statements are spread over several lines (F3) or run from a script with `.read`.
- Values typed in `.insert` and `.edit` are at most 255 characters; longer ones go in with `INSERT` or `UPDATE`.
- **UTF-8**: the CMM2 font has no umlauts etc. By default each UTF-8 character is shown as one placeholder (·), so columns stay aligned; files and the database keep real UTF-8. `.set utf8 raw` sends the bytes unchanged.
- **CSV** cannot tell NULL from an empty field; use `.dump` to keep NULL.
- The SQLite module has no `EXPLAIN`, `ANALYZE`, WAL mode, maths functions or extensions; `datetime('now')` gives the time of the CMM2 clock (no time zones). A database saved in WAL mode on a PC must be switched back there first (`PRAGMA journal_mode=DELETE;`).

## Credits

- SQL support in MMBasic and the SQLite module by Peter Mather; MMBasic by Geoff Graham and Peter Mather.
- SQLite is in the public domain (https://www.sqlite.org/copyright.html).