mmsql 1.0 – 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`)

Requirements

- Colour Maximite 2, Generation 1 or 2
- MMBasic **7.00.00RC6** or later (SQL support)
- The `sql` folder from `CMM2_Programs_7.00.00RC6.zip` in the root of the SD card (`A:/sql/sql.mod`)

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.

Commands

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

Highlights:

- **Browse** tables page by page, with `WHERE` and `ORDER BY`.
- **Edit** a row field by field: each value is shown in the line editor (`.edit table key`); `.insert`, `.delete`, `.create` and `.alter` work the same way, with a question before anything is changed.
- **Export** as in the sqlite3 shell: `.mode csv|list|insert|line|table`, `.headers`, `.separator`, `.once file`, `.output file`, `.dump`. CSV follows RFC 4180 with CRLF line ends.
- **Import** CSV with `.import [--skip N] file.csv table`, run SQL scripts with `.read file.sql`.
- **Check and repair**: `.check [full]`, `.fkcheck`, `.reindex`, `.vacuum`, and `.backup file.db` for a clean copy (`VACUUM INTO`).
- **Several databases** with `ATTACH 'other.db' AS o` – e.g. to copy tables between files.
- Line editor with history (Up/Down), Home/End, Del, Esc to clear the line.
- Options are saved in `mmsql.cfg`; `.set` lists them, `.set key value` changes one.

Tested byte for byte against SQLite on a PC: CSV and dump export, and the round trips CSV → database → CSV and dump → database → CSV, including quotes, line breaks, tabs, UTF-8 and 64-bit integers.

Good to know

- **One statement is at most 255 characters** (an MMBasic string); `.import` and `.insert` use at most 20 values.
- **Texts longer than 255 characters** cannot be read into MMBasic: mmsql shows them as `<text over 255 chars>` and exports them as `?`.
- **Ctrl-C** stops a long SQL statement but also ends mmsql (current firmware). The database stays intact; just start mmsql again.
- **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 (for a terminal set to UTF-8). Characters such as umlauts cannot be typed in mmsql.
- **CSV** cannot tell NULL from an empty field; use `.dump` to keep NULL.
- The SQLite module has no `EXPLAIN`, `ANALYZE`, WAL mode, math functions or extensions, and dates/times are UTC only. A database saved in WAL mode on a PC must be switched back first (`PRAGMA journal_mode=DELETE`).
- **Generation 1** works too, but large databases (over about 1 MB) are slow there, especially `.check full` and `.vacuum`.

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).