mmsql 1.0.1 – SQLite-Verwaltung für den Colour Maximite 2

mmsql ist ein Kommandozeilen-Werkzeug in MMBasic, mit dem man SQLite-Datenbanken auf dem Colour Maximite 2 untersuchen, ändern, prüfen, reparieren, importieren und exportieren kann. Es läuft gleichzeitig auf dem VGA/HDMI-Bildschirm und auf einem seriellen Terminal (z. B. Tera Term) und folgt den Befehlen der `sqlite3`-Shell. Wer SQLite vom PC kennt, findet sich also sofort zurecht.

© 2026 Matthias Hollatz – MIT-Lizenz (siehe `LICENSE.txt`)

Voraussetzungen

- Colour Maximite 2, Generation 1 oder 2
- MMBasic **7.00.00RC6** (SQL-Unterstützung)
- Für Firmware 8: **mmsql 1.1** (eigener Download, mit vielen Neuerungen)
- Der Ordner `sql` aus `CMM2_Programs_7.00.00RC6.zip` im Hauptverzeichnis der SD-Karte (`A:/sql/sql.mod`)

Installation

Diese Dateien in einen gemeinsamen Ordner kopieren, z. B. `A:/SQL/` (mit XMODEM: `xmodem receive "mmsql.bas"` usw.):

Datei	Zweck
`mmsql.bas`	das Programm
`mmsql_en.hlp`	englische Hilfe (Standard)
`mmsql_de.hlp`	deutsche Hilfe (`.set helpfile mmsql_de.hlp`)
`LICENSE.txt`	MIT-Lizenz

mmsql legt dort zwei weitere Dateien an: `mmsql.cfg` (Optionen) und `mmsql.his` (Befehlsverlauf).

Schnellstart

```
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
```

Befehle beginnen mit einem Punkt, alles andere wird als SQL ausgeführt. `.help` listet alle Befehle, `.help name` zeigt die Einzelheiten zu einem Befehl (z. B. `.help import`). `.help keys`, `.help sql` und `.help utf8` erklären Editor, SQL und UTF-8. Mit `.set helpfile mmsql_de.hlp` erscheint die Hilfe auf Deutsch.

Befehle

Gruppe	Befehle
Datenbank	`.open` `.close` `.info` `.databases` `.backup` `.vacuum`
Ansehen	`.tables` `.indexes` `.triggers` `.schema` `.columns` `.browse`
Daten	`.insert` `.edit` `.delete` `.import` `.read`
Struktur	`.create` `.alter` `.reindex`
Prüfen	`.check` `.fkcheck` `.fk`
Ausgabe	`.mode` `.headers` `.separator` `.nullvalue` `.once` `.output` `.dump`
Optionen	`.set` `.timer` `.bail` `.echo` `.cls` `.help` `.quit`

Das Wichtigste:

- **Blättern** in Tabellen Seite für Seite, mit `WHERE` und `ORDER BY`.
- **Bearbeiten** einer Zeile Feld für Feld: Jeder Wert steht vorbelegt im Zeileneditor (`.edit tabelle schlüssel`). `.insert`, `.delete`, `.create` und `.alter` arbeiten genauso und fragen nach, bevor etwas geändert wird.
- **Export** wie in der sqlite3-Shell: `.mode csv|list|insert|line|table`, `.headers`, `.separator`, `.once datei`, `.output datei`, `.dump`. CSV folgt RFC 4180 mit CRLF als Zeilenende.
- **Import** von CSV mit `.import [--skip N] datei.csv tabelle`, SQL-Skripte mit `.read datei.sql`.

- **Prüfen und Reparieren:** `.check [full]`, `.fkcheck`, `.reindex`, `.vacuum` sowie `.backup datei.db` für eine saubere Kopie (`VACUUM INTO`).
- **Mehrere Datenbanken** mit `ATTACH 'andere.db' AS o`, z. B. um Tabellen zwischen Dateien zu kopieren.
- Zeileneditor mit Verlauf (Hoch/Runter), Pos1/Ende, Entf und Esc zum Leeren der Zeile.
- Optionen werden in `mmsql.cfg` gespeichert. `.set` listet sie, `.set name wert` ändert eine.

Byte für Byte gegen SQLite am PC getestet: CSV- und Dump-Export sowie die Rundreisen CSV → Datenbank → CSV und Dump → Datenbank → CSV, einschließlich Anführungszeichen, Zeilenumbrüchen, Tabs, UTF-8 und 64-Bit-Ganzzahlen.

Gut zu wissen

- **Eine Anweisung darf höchstens 255 Zeichen lang sein** (ein MMBasic-String). `.import` und `.insert` verwenden höchstens 20 Werte.
- **Texte über 255 Zeichen** lassen sich nicht in MMBasic einlesen. mmsql zeigt sie als `<text over 255 chars>` an und exportiert sie als `?`.
- **Ctrl-C** bricht eine lange SQL-Anweisung ab, beendet aber auch mmsql (aktuelle Firmware). Die Datenbank bleibt heil; mmsql einfach neu starten.
- **UTF-8:** Der CMM2-Font hat keine Umlaute usw. Standardmäßig erscheint jedes UTF-8-Zeichen als ein Platzhalter (·), damit die Spalten gerade bleiben. Dateien und Datenbank behalten echtes UTF-8. `.set utf8 raw` sendet die Bytes unverändert (für ein Terminal mit UTF-8). Umlaute lassen sich in mmsql nicht eintippen.
- **CSV** kann NULL nicht von einem leeren Feld unterscheiden. Um NULL zu erhalten, `.dump` verwenden.
- Das SQLite-Modul hat kein `EXPLAIN`, `ANALYZE`, keinen WAL-Modus, keine Mathe-Funktionen und keine Erweiterungen; `datetime('now')` liefert die Uhrzeit der CMM2-Uhr (keine Zeitzonen). Eine Datenbank, die am PC im WAL-Modus gespeichert wurde, muss dort zuerst zurückgestellt werden (`PRAGMA journal_mode=DELETE`).
- **Generation 1** funktioniert ebenfalls, große Datenbanken (über etwa 1 MB) sind dort aber langsam, besonders bei `.check full` und `.vacuum`.

Änderungen in 1.0.1

- Behoben: Ein Wert mit genau 255 Zeichen, der mit einem Anführungszeichen endet, brachte `.mode csv`, `.mode insert` und `.dump` zum Absturz („256 is invalid“).
- Behoben: Zeilenumbrüche in einem Wert ergaben in `.mode line` am Terminal eine „Treppe“; jetzt beginnt eine neue, eingerückte Zeile.
- Kennt die Firmware `OPTION CRLF`, setzt mmsql sie auf CR, damit Enter keinen zusätzlichen leeren Prompt erzeugt.

Dank

- SQL-Unterstützung in MMBasic und das SQLite-Modul von Peter Mather; MMBasic von Geoff Graham und Peter Mather.
- SQLite ist Public Domain (https://www.sqlite.org/copyright.html).