mmsql 1.1 – 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`)

Die ausführliche Beschreibung aller Befehle, Optionen und Grenzen steht im Handbuch: `HANDBUCH_mmsql.pdf` (englisch: `MANUAL_mmsql.pdf`). Neu bei SQL? Am besten mit `TUTORIAL_mmsql.pdf` anfangen.

Voraussetzungen

- Colour Maximite 2, Generation 1, 2 oder 3
- MMBasic **8.00.00b1** oder neuer
- Der Ordner `sql` aus dem Firmware-Paket im Hauptverzeichnis der SD-Karte (`A:/sql/sql.mod`)
- Für Firmware 7.00.00RC6: **mmsql 1.0.1** (eigener Download)

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
`demo.sql`	Beispieldaten für das Tutorial

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 eines Befehls (z. B. `.help import`), und `.help keys`, `.help sql` und `.help utf8` erklären Editor, SQL und UTF-8.

Befehle

Gruppe	Befehle
Datenbank	`.open` `.close` `.info` `.databases` `.backup` `.vacuum` `.files` `.kill`
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 seitenweise, mit `WHERE` und `ORDER BY`.

- **Bearbeiten** einer Zeile Feld für Feld im Zeileneditor (`.edit tabelle schlüssel`); `.insert`, `.delete`, `.create` und `.alter` arbeiten genauso, mit einer Rückfrage, bevor etwas geändert wird.
- **Export** wie in der sqlite3-Shell: `.mode csv|list|insert|line|table`, `.headers`, `.separator`, `.once datei`, `.output datei`, `.dump`. CSV nach RFC 4180 mit CRLF-Zeilenenden.
- **Import** von CSV mit `.import [--skip N] datei.csv tabelle`, SQL-Skripte mit `.read datei.sql`.
- **Lange Texte und BLOBs** über 255 Zeichen werden vollständig angezeigt, exportiert, gedumpt, eingelesen und importiert (Option `maxtext`, Standard 32 KB).
- **Ctrl-C** bricht eine laufende SQL-Anweisung ab, und mmsql läuft weiter; es sagt genau, was SQLite rückgängig gemacht hat.
- **Transaktionen** im Blick: Der Prompt zeigt einen Stern, solange eine offen ist (`mmsql[club.db]*>`), und `.quit`, `.close` und `.open` fragen nach Commit oder Rollback.
- **Mehrere Anweisungen in einer Zeile**, getrennt mit `;`, jede mit eigenem Ergebnis.
- **Prüfen und Reparieren**: `.check [full]`, `.fkcheck`, `.reindex`, `.vacuum` und `.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.
- **Dateien**: `.files [muster]` listet Dateien, `.kill datei` löscht eine nach Rückfrage.
- Zeileneditor mit Verlauf (Pfeil hoch/runter), Pos1/Ende, Entf, Esc zum Leeren der Zeile. Die Optionen stehen in `mmsql.cfg`.

Byte für Byte gegen SQLite auf dem PC getestet: CSV-, Insert- und Dump-Export sowie die Rundreisen CSV → Datenbank → CSV und Dump → Datenbank → Dump, mit Anführungszeichen, Zeilenumbrüchen, Tabs, UTF-8, 64-Bit-Ganzzahlen, Texten bis 40 000 Zeichen und BLOBs.

Gut zu wissen

- **Eine Eingabe am Prompt hat höchstens 255 Zeichen** (ein MMBasic-String). Längere Anweisungen lassen sich mit `.read` aus einer Datei ausführen. Eine Anweisung muss in einer Zeile stehen.
- `.insert` und `.import` verarbeiten höchstens **20 Spalten** (Grenze der Firmware pro Befehl).
- **UTF-8**: Die CMM2-Schrift 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.
- **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`).
- **Firmware 8.00.00b1** hat einen kleinen Fehler in `LGETSTR$` (ein Byte zu viel bei genau 255 Zeichen); mmsql umgeht ihn. In der nächsten Firmware ist er behoben.

Dank

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