mmsql 1.2 – 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 dem Tutorial anfangen (englisch, eigener Download `mmsql_tutorial.zip`).

Voraussetzungen

- Colour Maximite 2, Generation 1, 2 oder 3
- MMBasic **8.00.00b1** oder neuer (8.00.00b2 empfohlen)
- Der Ordner `sql` aus dem Firmware-Paket im Hauptverzeichnis der SD-Karte (`A:/sql/sql.mod`)
- Für die Firmware 7.00.00RC6 gibt es **mmsql 1.0.1** (eigener Download)

Installation

Diese Dateien in einen 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 (z. B. `.help import`), und `.help keys`, `.help sql` und `.help utf8` erklären Editor, SQL und UTF-8.

Neu in 1.2

- **Ein richtiger Editor am Prompt**: SQL über bis zu 20 Zeilen mit Statuszeile – F3 neue Zeile, F2 ausführen. Am PC geschriebenes und eingefügtes SQL wird erkannt und zusammengehalten.
- **Eine Regel für die Eingabe**: Eine Zeile läuft mit Enter; mehrere Zeilen sind ein Block, jede Anweisung endet mit `;`.
- **Lange Anweisungen und Abfragen** über 255 Zeichen, mit Ergebnis – am Prompt, in Blöcken und in Skripten.
- **Verlauf** für Anweisungen jeder Länge, auch nach einem Neustart; Bild auf/ab.
- **Skripte wie in sqlite3**: `.read` nimmt Anweisungen über mehrere Zeilen, zeigt Ergebnisse von Abfragen und kennt Punkt-Befehle.
- **Keine Grenze von 20 Spalten mehr**: `.import` ohne Grenzen für Spalten und lange Felder, `.insert` bis 64 Spalten.
- **Datenbank im Speicher** (`:memory:`) – mmsql startet auch ohne Datei.
- **Neue Ausgabe**: `.mode markdown`, `.mode json`, `.width`, `.print`, `.set blob`.
- **`.dump`** erhält den echten Typ jedes Werts (BLOB, Text, 2.0 als REAL, sehr große Ganzzahlen).
- **Weitere Befehle**: `.restore`, `.version`, Suchmuster für `.tables`, `.schema`, `.indexes`, `.triggers`; `.set screen`, `.set width 0`.
- Kürzerer Prompt `[retro.db]>`; Terminal und Bildschirm zeigen genau dasselbe.

Achtung: `.read` braucht jetzt ein `;` am Ende jeder Anweisung, wie `sqlite3`. Alte Skripte mit einer Anweisung pro Zeile ohne `;` brauchen ein `;` am Ende jeder Zeile. Dumps aus mmsql hatten es schon immer.

Alle Änderungen: `CHANGES.txt`.

Befehle

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

Weiterhin aus 1.1

- **Blättern** in Tabellen, mit `WHERE` und `ORDER BY`; eine Zeile Feld für Feld **bearbeiten** (`.edit`, `.insert`, `.delete`, `.create`, `.alter`), mit Rückfrage, bevor etwas geändert wird.
- **Export** wie in der sqlite3-Shell: `.mode csv|list|insert|line|table|markdown|json`, `.headers`, `.separator`, `.once`, `.output`, `.dump`. CSV nach RFC 4180 mit CRLF-Zeilenenden.
- **Lange Texte und BLOBs** über 255 Zeichen vollständig (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 (`[club.db]*>`), und `.quit`, `.close` und `.open` fragen nach Commit oder Rollback.
- **Prüfen und Reparieren**: `.check [full]`, `.fkcheck`, `.reindex`, `.vacuum`, `.backup`.
- **Mehrere Datenbanken** mit `ATTACH 'andere.db' AS o;`.

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, langen Texten und BLOBs. Eine automatische Testreihe für den CMM2 gibt es als eigenen Download (`mmsql_testsuite.zip`).

Gut zu wissen

- **Eine Zeile im Editor hat höchstens 255 Zeichen** (ein MMBasic-String). Längere Anweisungen auf mehrere Zeilen verteilen (F3) oder mit `.read` aus einem Skript ausführen.
- Werte, die in `.insert` und `.edit` getippt werden, haben höchstens 255 Zeichen; längere mit `INSERT` oder `UPDATE` eingeben.
- **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;`).

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).