

## Einführung in SQLite als Backend für MS Access

**Problem:** Sie arbeiten mit MS Access unter Windows, möchten für ihr Projekt Daten sammeln und diese später einer Gemeinschaft zur Verfügung stellen ohne proprietäre Formate zu verwenden oder servergebundene Datenbanksysteme vorauszusetzen.

**Lösung:** Sie verwenden [SQLite](#) als Datenbackend oder exportieren die Daten abschließend in dieses System. Letzteres wäre natürlich auch über einen CSV-Import möglich, wesentlich eleganter ist aber natürlich eine korrekte Entsprechung der Datenbankstruktur in SQL einschließlich der Abhängigkeiten (foreign key).

### 1. ODBC

Die Verbindung zwischen SQLite und MS Access wird mit [ODBC](#) hergestellt, wofür ein entsprechender Treiber installiert werden muss ([SQLite-ODBC Treiber](#) von Christian Werner).

ODBC-Schnittstellen können unter Windows als Nutzer-, System- oder Datei-DSN gespeichert werden. Erstellt werden Sie aus der Anwendung heraus (z.B. Externe Daten → ODBC) oder vom Betriebssystem aus (C:\Windows\System32\odbcad32.exe oder C:\Windows\SysWOW64\odbcad32.exe). Eventuell werden Sie wegen fehlender Schreibrechte von einer System-DNS auf eine Nutzer-DNS verwiesen. Die System- und Nutzer-DSN werden in der Registry von Windows gespeichert, die Datei-DSN im ausgewählten Verzeichnis.



Abb. 1 Neue Datenquelle definieren.

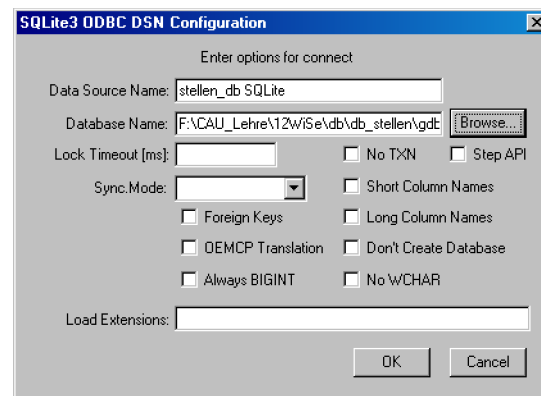


Abb. 2 ODBC-DSN mit den wesentlichen Angaben zum Namen der Verbindung und dem Pfad zur SQLite-Datenbank.

Mögliche und bekannte Probleme beim Einrichten der ODBC-Schnittstelle:

- Die ODBC-Schnittstelle funktioniert nur als Nutzer- oder System DSN, bei einer Datei-DSN tritt ein Fehler auf, der auf den „falschen“ Treiberverweis unter „DRIVER=“ zurückzuführen ist. Die Eintragung von „C:\Windows\System32\sqlite3odbc.dll“ bringt keine Lösung. Verwenden Sie ausschließlich Nutzer- oder System-DSN.
- Die Schnittstelle ist extrem langsam. Eine mögliche Lösung bietet das Beenden der Ablaufverfolgung in der ODBC-Definition: „C:\Windows\System32\odbcad32.exe“, hier Register „Ablaufverfolgung“.

### 2. Zugriff von MS Access

Die Verbindung wird aus MS Access über „Externe Daten“ → „ODBC“ → „Herstellen einer Verknüpfung“ realisiert. Die Tabellen werden mit einem Blauen Pfeil und einer Weltkugel gekennzeichnet. Im Popup bei Mouserover wird die Syntax der ODBC-Schnittstelle angezeigt.

Möglich Fehler bei der Arbeit mit den verknüpften Tabellen:

- In den Feldern steht nach dem erneuten Öffnen der Tabelle nur noch „#Gelöscht“.  
Die ODBC-Schnittstelle benötigt für den Datenabgleich ein eindeutiges und benanntes Indexfeld. Das von SQLite automatisch generierte Feld rowid reicht nicht.  
Lösung: Erstellen Sie das Indexfeld eindeutig mit „rowID“ (integer primary key autoincrement).  
Erstellen sie ggf. zusätzliche Indices mit CONSTRAINT indexname UNIQUE (attribut1,...);
- Neue Datensätze lassen sich problemlos einfügen, die Werte nachträglich aber nicht mehr verändern. Das Problem liegt vermutlich in der eindeutigen Zuweisung der Datensätze zwischen Frontend und Backend. Bei einigen SQL-Datenbanksystemen hilft hier möglicherweise die Verwendung des Attributes timestamp im Backend ([Forum Microsoft](#)). Bei SQLite lässt sich dieses Problem so nicht lösen.  
Lösung: Verwenden Sie Aktualisierungsabfragen (UPDATE) für das Ändern der Daten. Verzichten Sie auf das Editieren der Daten in einem Recordset sondern verwenden Sie „docmd.runsql (SQL)“.
- Der Zugriff von mehreren Clients ist theoretisch nicht möglich, da SQLite für den Einzelanwender konzipiert ist. Ein schneller Test mit einer geöffneten SQL-Datenbank und zwei ODBC-Zugriffen aus zwei MS Access Datenbanken erlaubte aber das Anfügen von Datensätzen, lieferte korrekte Indexwerte und zutreffende Fehlermeldungen bei Schlüsselverletzungen.

### 3. Daten veröffentlichen bzw. archivieren

Wenn Sie Ihre Daten veröffentlichen wollen, sollten Sie mindestens zwei Objekte liefern:

- die Daten selbst und
- die Dokumentation der Struktur und ggf. Erläuterungen zu Werten.

Für die Daten erstellen aus SQLite heraus einen Export in SQL-Anweisungen (dump), das Ergebnis ist eine Textdatei (vermutlich UTF-8) mit Anweisungen für die Tabellenstruktur und die zugehörigen Daten. Diese Datenstruktur ist hervorragend archivierbar, wesentliche Fehlerquelle ist hier nur die Lesbarkeit des Datenstroms bzw. Fehler auf dem Datenträger. Die SQLite Datenbank, ggf. mit der verwendeten SQLite-Version (z.B. [sqlite-shell-osx-x86-3071502.zip](#)), kann von jedem Nutzer ohne zusätzliche Kosten geöffnet werden. Die dritte und unter archivalischen Gesichtspunkten schlechteste Variante ist die MS Access Datenbank selbst, auch wenn hierzu möglicherweise eine [Laufzeitversion](#) zur Verfügung steht.

Für die der Datenbankstruktur steht in MS Access der „Dokumentierer“ zur Verfügung:

„Datenbanktools“ → „Datenbankdokumentierer“. Hier können für Tabellen, Abfragen, Formulare, Berichte und Programmierung (Makros und VBA-Module) Berichte auf unterschiedlichem Detailstufen für die Rezeption durch Menschen erstellt werden, die in einem gängigen Format, z.B. PDF, archiviert werden können. In SQL sind die im „dump“ enthaltenen CREATE-Anweisungen sowohl durch Programme als auch durch Menschen lesbar.

Autor: Christoph Rinne