

	<b>SQLite v3.3.8.1</b>	
	<b>Software library for RDBMS</b>	
	<b>System Requirements:</b> Operating system : Windows 2000 / XP / 2003 / Vista / 2008 / Windows 7 / 2008 R2	
	<b>Développé par:</b>	<a href="http://www.sqlite.org">www.sqlite.org</a>
Open source	<b>Personne à contacter</b>	jeanpierre.latour@smals.be

### Fonctions

**SQLite** est une **base de données relationnelle livrée sous la forme d'une librairie**. A la différence des bases de données traditionnelles, SQLite ne s'exécute pas en tant que processus standalone, appelé par les applications clientes, mais est liée à et donc partie intégrale (embedded) d'une application cliente. L'application cliente interagit avec SQLite par le biais de simples appels de fonctions, ce qui contribue à réduire la latence lors des accès aux données - Nous sommes ici en single et non en inter-process communication.

SQLite est **ACID compliant** et supporte les ordres **SQL-92** standards fondamentaux, en oubliant volontairement les ordres « ésotériques » peu utilisés par les développeurs. Ceci va dans le sens de la philosophie générale du produit dont les maîtres mots sont, dans l'ordre : robustesse, administration simple, faible occupation-mémoire, vitesse et fonctionnalités. Dans ses premières versions elle supportait les accès concurrents en lecture mais pas en écriture. Depuis la version 3.7 cette restriction peut être levée en activant le **WAL** (Write-Ahead Logging - mécanisme de gestion des modifications qui écrit d'abord dans un fichier de log avant de les appliquer sur la base de données).

- |   |   |  |
|---|---|--|
| <ul style="list-style-type: none"> <li>• <a href="#">aggregate functions</a></li> <li>• <a href="#">ALTER TABLE</a></li> <li>• <a href="#">ANALYZE</a></li> <li>• <a href="#">ATTACH DATABASE</a></li> <li>• <a href="#">BEGIN TRANSACTION</a></li> <li>• <a href="#">comment</a></li> <li>• <a href="#">COMMIT TRANSACTION</a></li> <li>• <a href="#">core functions</a></li> <li>• <a href="#">CREATE INDEX</a></li> <li>• <a href="#">CREATE TABLE</a></li> <li>• <a href="#">CREATE TRIGGER</a></li> <li>• <a href="#">CREATE VIEW</a></li> </ul> | <ul style="list-style-type: none"> <li>• <a href="#">CREATE VIRTUAL TABLE</a></li> <li>• <a href="#">date and time functions</a></li> <li>• <a href="#">DELETE</a></li> <li>• <a href="#">DETACH DATABASE</a></li> <li>• <a href="#">DROP INDEX</a></li> <li>• <a href="#">DROP TABLE</a></li> <li>• <a href="#">DROP TRIGGER</a></li> <li>• <a href="#">DROP VIEW</a></li> <li>• <a href="#">END TRANSACTION</a></li> <li>• <a href="#">EXPLAIN</a></li> <li>• <a href="#">expression</a></li> <li>• <a href="#">INDEXED BY</a></li> </ul> | <ul style="list-style-type: none"> <li>• <a href="#">INSERT</a></li> <li>• <a href="#">keywords</a></li> <li>• <a href="#">ON CONFLICT clause</a></li> <li>• <a href="#">PRAGMA</a></li> <li>• <a href="#">REINDEX</a></li> <li>• <a href="#">RELEASE SAVEPOINT</a></li> <li>• <a href="#">REPLACE</a></li> <li>• <a href="#">ROLLBACK TRANSACTION</a></li> <li>• <a href="#">SAVEPOINT</a></li> <li>• <a href="#">SELECT</a></li> <li>• <a href="#">UPDATE</a></li> <li>• <a href="#">VACUUM</a></li> </ul> |
|---|---|--|

Etant donné sa compacité (+/- 350 KB – dépendant de la plate-forme et du mode de compilation [32 ou 64 bits]) et sa philosophie « **zero administration** », SQLite est très largement utilisée dans les **mobile devices** et les **systèmes embarqués**, limités par la mémoire et la puissance du processeur.

SQLite est bien appropriée pour les phases de développement et de test, ou pour des démonstrations, là où installer une vraie base client/serveur peut s'avérer finalement trop lourd et compliqué par rapport aux objectifs poursuivis.

### Conclusions / Recommandations

SQLite est un choix judicieux comme alternative ou en complément d'un système de gestion de fichiers, par exemple pour gérer les paramètres d'une application, voire pour mettre en place un système de cache en local. Sa réputation n'est plus à faire comme solution de choix pour les mobile devices et les systèmes embarqués. Couplée au plugin Mozilla SQLite Manager ou à SQLite Expert Personal, elle permet de disposer d'une alternative complète et gratuite aux RDBMS client-serveur dans les cas de figure où ceux-ci ne s'imposent pas. A retenir pour les systèmes stand-alone et/ou mettant l'accent sur de faibles coûts de support.

## Fonctions (suite)

SQLite est bien appropriée pour les phases de développement et de test, ou pour des démonstrations, là où installer une vraie base client/serveur peut s'avérer finalement trop lourd et compliqué par rapport aux objectifs poursuivis.

Des outils gratuits sont disponibles pour visualiser et gérer la base de données. Citons en particulier le plugin **Mozilla SQLite Manager** et **SQLite Personal Manager**.

Un soin très poussé est apporté au testing du produit : environ 60% du code écrit pour SQLite existe seulement pour raison de test. Le plus gros de ce code de test est écrit en [Tcl](#). SQLite est réputé « **100 % branch test coverage** ».

De même, malgré son caractère open source **SQLite est très contrôlé en matière de contribution**. Il faut savoir que beaucoup d'acteurs utilise SQLite sans publiquement le dire. Les avocats de ces éditeurs veulent évidemment éviter que ne surviennent des problèmes de non-appartenance du code.

SQLite fait maintenant partie de PHP5, ainsi que de Mac OS X.

SQLite offre des APIs pour de nombreux langages : Basic, C, C++, Tcl, Java, Lua, PHP, Python, Javascript, R, REBOL, ...

Une caractéristique importante du produit est le **variable length record**. Alors que la plupart des RDBMs alloue de l'espace disque en fonction de la longueur déclarée des données stockées, SQLite utilise l'espace strictement nécessaire. Ceci offre des avantages immédiats en termes de compacité et de performance, étant donné que moins de données sont déplacées du et vers le disque. Autre avantage : le **manifest typing** plutôt que le static typing (en particulier intéressant si vous utilisez un langage qui supporte le type Variant – mais cette liberté présente ses dangers).

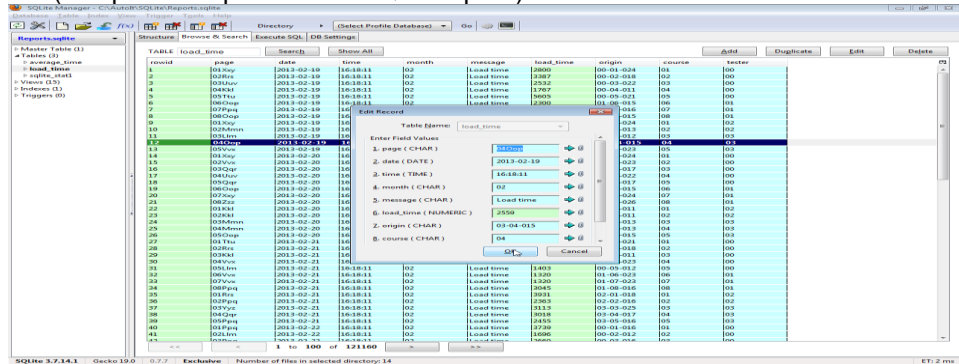
La principale **limitation** de SQLite porte **sur la commande ALTER**. Elle permet juste d'ajouter une colonne à la fin d'une table ou de changer le nom d'une table. Pour des opérations plus complexes sur la structure d'une table il est nécessaire de la recréer (voir par exemple le code SQL ci-après).

```

BEGIN TRANSACTION;
CREATE TEMPORARY TABLE
t1_backup(a,b);
INSERT INTO t1_backup SELECT
a,b FROM t1;
DROP TABLE t1;
CREATE TABLE t1(a,b);
INSERT INTO t1 SELECT a,b FROM
t1_backup;
DROP TABLE t1_backup;
COMMIT;

```

Signalons enfin l'existence de l'exécutable **SQLite3.exe** qui permet l'interaction avec la db en ligne de commande.



## Tests et Résultats

Nous avons utilisé SQLite dans le cadre d'un système stand-alone de monitoring de sites web et d'exploitation des logs associés, développé avec le langage de scripting AutoIT ([voir autre quick review](#)), pour le stockage moyen terme et pour bénéficier des possibilités de filtrage et de tri inhérents à une base de données. Egalement pour pouvoir profiter de certaines fonctions SQL au sein de nos développements.

Nous avons donc pu amplement vérifier la compacité, la fiabilité, la performance et la simplicité de gestion de SQLite, dans versions 32 et 64 bits. Nos développements sont restés limités à la plate-forme Windows.

Nos tests très approfondis ont démontré la facilité de prise en mains de SQLite, son orientation zero administration, sa fiabilité et sa performance.

La version originale de l'outil existe depuis 2001. Sa déjà longue histoire et un usage répandu (mais le plus souvent discret) lui confère des gages de pérennité.

## Budget

SQLite est une solution entièrement gratuite.