


SQLite

SQLite

	
Developer(s)	D. Richard Hipp
Initial release	August 2000
Latest stable	3.8.4.2 (March 26, 2014) ^[1]
Written in	C
Operating system	Cross-platform
Size	658 KiB
Type	RDBMS (embedded)
License	Public domain
Website	sqlite.org ^[2]

SQLite (/ˌɛskjuːɛlˈlaɪt/ or /ˈsiːkwəl.laɪt/) is a relational database management system contained in a C programming library. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it.

SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems, among others. SQLite has many bindings to programming languages.

The source code for SQLite is in the public domain.

Design

Unlike client–server database management systems, the SQLite engine has no standalone processes with which the application program communicates. Instead, the SQLite library is linked in and thus becomes an integral part of the application program. (In this, SQLite follows the precedent of Informix SE of c. 1984 ^[3]) The library can also be called dynamically. The application program uses SQLite's functionality through simple function calls, which reduce latency in database access: function calls within a single process are more efficient than inter-process communication. SQLite stores the entire database (definitions, tables, indices, and the data itself) as a single cross-platform file on a host machine. It implements this simple design by locking the entire database file during writing. SQLite read operations can be multitasked, though writes can only be performed sequentially.

History

D. Richard Hipp designed SQLite in the spring of 2000 while working for General Dynamics on contract with the United States Navy. Hipp was designing software used on board guided missile destroyers, which were originally based on HP-UX with an IBM Informix database back-end. The design goals of SQLite were to allow the program to be operated without installing a database management system or requiring a database administrator. In August 2000, version 1.0 of SQLite was released, based on gdbm (GNU Database Manager). SQLite 2.0 replaced gdbm with a custom B-tree implementation, adding support for transactions. SQLite 3.0, partially funded by America Online, added internationalization, manifest typing, and other major improvements.

In 2011 Hipp announced his plans to add an UnQL interface to SQLite databases and to develop *UnQLite*, an embeddable document-oriented database. Howard Chu ported SQLite 3.7.7.1 to use Openldap MDB instead of the original Btree code and called it sqlightning. One test of inserting 1000 records was 20 times faster.^{[4][5]}

Features

SQLite implements most of the SQL-92 standard for SQL but it lacks some features. For example it has partial support for triggers, and it can't write to views (however it supports INSTEAD OF triggers that provide this functionality). While it supports complex queries, it still has limited ALTER TABLE support, as it can't modify or delete columns.

SQLite uses an unusual type system for an SQL-compatible DBMS; instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms it is *dynamically typed*. Moreover, it is *weakly typed* in some of the same ways that Perl is: one can insert a string into an integer column (although SQLite will try to convert the string to an integer first, if the column's preferred type is integer). This adds flexibility to columns, especially when bound to a dynamically typed scripting language. However, the technique is not portable to other SQL products. A common criticism is that SQLite's type system lacks the data integrity mechanism provided by statically typed columns in other products. The SQLite web site describes a "strict affinity" mode, but this feature has not yet been added. However, it can be implemented with constraints like `CHECK (typeof(x)='integer')`.

Several computer processes or threads may access the same database concurrently. Several read accesses can be satisfied in parallel. A write access can only be satisfied if no other accesses are currently being serviced. Otherwise, the write access fails with an error code (or can automatically be retried until a configurable timeout expires). This concurrent access situation would change when dealing with temporary tables. This restriction is relaxed in version 3.7 when write-ahead logging (WAL) is turned on enabling concurrent reads and writes.

A standalone program called `sqlite3` is provided that can be used to create a database, define tables within it, insert and change rows, run queries and manage an SQLite database file. This program is a single executable file on the host machine. It also serves as an example for writing applications that use the SQLite library.

SQLite full Unicode support is optional.

SQLite has automated regression testing prior to each release. Over 2 million tests are run as part of a release's verification. Starting with the August 10, 2009 release of SQLite 3.6.17, SQLite releases have 100% branch test coverage, one of the components of code coverage.

As of version 3.8.2 it's possible to create tables without rowid.^[6]

Development

SQLite development stores revisions of its source code in Fossil, a distributed version control system that is itself built upon an SQLite database.

Adoption

Programming languages

SQLite has bindings for a large number of programming languages, including :

- | | | |
|--------------------|---------------------------------|-------------------------|
| • BASIC | • Haskell | • PureBasic |
| • Delphi | • Java | • Python ^[7] |
| • C | • JavaScript ^[8] | • R |
| • C# | • Livecode | • REALbasic |
| • C++ | • Lua | • REBOL |
| • Clipper//Harbour | • newLisp | • Ruby ^[9] |
| • Common Lisp | • Objective-C (on OS X and iOS) | • Scheme |
| • Curl | • OCaml | • Smalltalk |
| • D | • Perl ^[10] | • Tcl |
| • Free Pascal | • PHP | • Visual Basic |
| | • Pike | |

Middleware

- ADO.NET adapter, initially developed by Robert Simpson, is maintained jointly with the SQLite developers since April 2010.^[11]
- ODBC driver has been developed and is maintained separately by Christian Werner.^[12] Werner's ODBC driver is the recommended connection method for accessing SQLite from OpenOffice.org.^[13]
- COM (ActiveX) wrapper making SQLite accessible on Windows to scripted languages such as JScript and VBScript. This adds database capabilities to HTML Applications (HTA).

Web browsers

- Mozilla Firefox and Mozilla Thunderbird store a variety of configuration data (bookmarks, cookies, contacts etc.) in internally managed SQLite databases, and even offer an add-on to manage SQLite databases.
- Google's Chrome browser
- The Opera Internet suite and browser uses SQLite 3.7.9 for managing WebSQL databases. This is noted in `opera:about`, although without the mention of WebSQL (databases can be managed through `opera:webdatabases`).

Web application frameworks

- Django, a Python web framework, supports SQLite3 by default.
- As of version 7, Drupal, a PHP-based content management system for making websites and blogs, has an option to install using SQLite.
- Ruby on Rails' default database management system is also SQLite.
- web2py, a Python web framework, default database management system is also SQLite.

Various

- Skype is a widely deployed application that uses SQLite.
- Adobe Systems uses SQLite as its file format in Adobe Photoshop Lightroom, a standard database in Adobe AIR, and internally within Adobe Reader.
- The Service Management Facility, used for service management within the Solaris and OpenSolaris operating systems, uses SQLite internally.
- Flame, a malware program used for cyberespionage, used SQLite to store the data it collects.
- The Xojo Programming Language has SQLite support built in to both the desktop and web frameworks.

Operating systems

Because of its small size, SQLite is well suited to embedded systems, and is also included in:

- Blackberry's BlackBerry 10 OS
- Microsoft's Windows Phone 8
- Apple's iOS
- Symbian OS
- Nokia's Maemo
- Google's Android
- Linux Foundation's MeeGo
- LG's webOS
- NetBSD
- OpenBSD
- FreeBSD where starting with 10-RELEASE version it is the core package management system.

However, it is also suitable for desktop operating systems; Apple adopted it as an option in OS X's Core Data API from the original implementation in Mac OS X 10.4 onwards, and also for administration of videos and songs on the iPhone.

Citations

- [1] http://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/SQLite&action=edit
- [2] <http://sqlite.org/>
- [3] <http://www.iiug.org/faqs/informix-faq/ifaq01.htm.1#1.2>
- [4] MDB: A Memory-Mapped Database and Backend for OpenLDAP (<http://ldapcon.org/downloads/chu-paper.pdf>), Howard Chu, [MDB: A Memory-Mapped Database and Backend for OpenLDAP LDAPCon 2011].
- [5] sqlightning source code (<http://gitorious.org/mdb/sqlightning#more>).
- [6] ReleaseLog (<http://sqlite.org/news.html>) SQLite.org, visited 8th December 2013
- [7] PySQLite (<http://trac.edgewall.org/wiki/PySQLite>): Python bindings for SQLite
- [8] JSPDO (<http://code.google.com/p/v8-juice/wiki/JSPDO>) JavaScript database access abstraction API
- [9] SQLite/Ruby (<http://rubyforge.org/projects/sqlite-ruby>): Ruby bindings for SQLite
- [10] DBD::SQLite (<https://metacpan.org/module/DBD::SQLite>): Perl DBI Interface to SQLite
- [11] <http://system.data.sqlite.org/index.html/doc/trunk/www/index.wiki>
- [12] <http://www.ch-werner.de/sqliteodbc/>
- [13] http://documentation.openoffice.org/HOW_TO/data_source/SQLite.pdf

References

- Allen, Grant; Owens, Mike (November 5, 2010). *The Definitive Guide to SQLite* (<http://apress.com/book/view/1430232250>) (2nd ed.). Apress. p. 368. ISBN 1-4302-3225-0.
- Kreibich, Jay A. (August 17, 2010). *Using SQLite* (<http://oreilly.com/catalog/9780596521196>) (1st ed.). O'Reilly Media. p. 528. ISBN 0-596-52118-9.
- van der Lans, Rick F. (September 7, 2009). *The SQL Guide to SQLite* (1st ed.). lulu.com. p. 542. ISBN 0-557-07676-5.
- Newman, Chris (November 9, 2004). *SQLite (Developer's Library)* (<http://www.informit.com/store/product.aspx?isbn=067232685X>) (1st ed.). Sams. p. 336. ISBN 0-672-32685-X.

External links

- Official website (<http://www.sqlite.org/>)
 - SQLite (<http://www.dmoz.org/Computers/Software/Databases/SQLite>) on the Open Directory Project
 - An Introduction to SQLite (<https://www.youtube.com/watch?v=giAMt8Tj-84>) on YouTube
-

Article Sources and Contributors

SQLite *Source:* <http://en.wikipedia.org/w/index.php?oldid=600050898> *Contributors:* *drew, 0x54097DAA, A. Parrot, AVRS, Abdull, Alex LE, Alex degarate, AlistairMcMillan, Alvin-cs, Amanda hoic, AndrewHowse, Anybody, Aquatopia, Arcturis, Arite, Arulprasad, Aulis2003, AxelBoldt, Baest, Barefootguru, Bbb2007, Beethoven05, Beeveer, Bgwhite, Bigown, Bkkbrad, Boleslav Bobcik, Bro4, Brockert, Bryan Derksen, C. A. Russell, CWenger, Canadianacademic, Cander0000, Captain Conundrum, CeciliaPang, Chealer, Choas, Chris Caven, ChrisNoe, Claw of Slime, Cmbeelby, Compctech, Cooldude7273, DG, DStoykov, Damian Yerrick, DanBishop, Danim, Darkzen.bps, Dave2, David Gerard, Daxx wp, Dchestnykh, Deflective, Delpino, Devin Asay, Dexp, DivideByZero14, Djbriddock, DmiTrix, Dnas, Dougher, Drano, Drhipp, Dsimpson dcsi, Ebraminio, Eequor, Egrabczewski, Eugenwpg, FFMG, FatalError, Fenke, Fnielsen, Frap, Fred Bradstadt, Fubar Obfusco, Furrykef, GabrielePx, Gerardo cabero, Gilesmorant, Gioto, Glenn, Glider87, Graue, Greensburger, GregorB, Grshiplett, Gudeldar, Gwern, Götz, H@rld, Haakon, Hankwang, Hareesh.t, Havarhen, Heelmijnlevenlang, Hif, Hoo man, Hu12, Huji, Inverse.chi, Irishguy, Isilanes, J Milburn, JLaTondre, Jaguar83, JamesHaigh, Jandalhandler, Javier Odom, Jaxelrod, Jayapalchandran, Jeanfarias, Jmkim dot com, Johnnuiq, JonathanWakely, Jorel314, Jrouquie, Jstaniek, KTC, Kate, Kimchi.sg, Kirbylover4000, Kl4m-AWB, Korval, Kriplozoik, Kwamikagami, Kwiki, Leonelhs, Lethalmonk, Lfstevens, LilHelpa, Lmatt, LordArtemis, Lotje, Luk, MBParker, MBlakley, MVelliste, Madooo12, Magicmike, Magister Mathematicae, Mainstreetmark, Makyen, Many delicious meats, Marco1317, Marco1713, Mark Arsten, Marvinella, MaterialsScientist, Mattisgoo, Mchirico, Mdwh, Mecanismo, Melizg, Mindmatrix, Minghong, Mipadi, Molyneux.peter, Moxfyre, MrOllie, Mrinal.kant, Mschumacher69, Mwtwoews, Najeeb1010, Nate Silva, Ned Scott, Neilc, Neurolysis, Niceguyedc, Nikai, Nnemo, Noncoercive, Nuno Brito, OdIn, Ohconfucius, Optikos, Otterfan, OzOle, PabloCastellano, Palosirkka, Paolosupernova, Pascal sa, Patuck, Paul Magnussen, Peak, Permafrost46, Philipolson, Philippe.petrinko, Pmsyyz, Pnm, Polar, PotatoEater, Qu1j0t3, Quadunit404, Qwyrxian, RadioActive, RandalSchwartz, Rapha222, Reedy, Rfl, Rgbea, Rich Farmbrough, Risk one, RobertCailliau, Roberta F., Ronark, Rorx, RossPatterson, Rrjanbiah, Ruakh, Rursus, SF007, Sae1962, Saltlakejohn, Sanxiyn, Sappy, Sarathbs, Scartboy, Scjessey, Scullder, Sdfisher, Sekelsenmat, SexyBern, Shello, SimonEast, Simple Bob, Sleske, Slicky, Smyth, Soumyasch, SpacePacket, Specious, SqlPac, Srprez, Stachelfisch, Staticmain, Steph1978, Stephan Leeds, Streambag, Superm401, Svick, Syntaxerrorz, TOR, Tablizer, TankMiche, Temblast, Tgwena, Theone00, Therealgeeves, Theta682, Thumperward, ThurnerRupert, Tim baroon, Timwi, Tobias Conradi, Toehead2001, TripleF, Turnstep, Two Bananas, U1024, Ungzd, Vdvluc, Visor, Vlops, Volkris, Wael Ellithy, Waldir, Where, Who.am.i, Wikimercenary, Wikitiki89, Wikiwikiw, Winterst, Wjgilmore, Wonderfl, Wwwolf, Wykypydya, Xephyrous, Xtremejames183, Ybungalobill, Yoghurt, Zaxn1234, Zbig Lebowsky, ZeaForUs, Zero0w, Zhongchen, Zo8shong, 469 anonymous edits

Image Sources, Licenses and Contributors

File:SQLite370.svg *Source:* <http://en.wikipedia.org/w/index.php?title=File:SQLite370.svg> *License:* Public Domain *Contributors:* Part of the SQLite documentation, which has been released by author D. Richard Hipp to the public domain. SVG conversion by Mike Toews.

License

Creative Commons Attribution-Share Alike 3.0
[//creativecommons.org/licenses/by-sa/3.0/](http://creativecommons.org/licenses/by-sa/3.0/)