

# SQLite

**SQLite** (*/ˌɛskjuːəlˈlaɪt/*<sup>[3]</sup> or */ˈsiːkwəl.laɪt/*<sup>[4]</sup>) is a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

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.<sup>[5]</sup>

SQLite is a popular choice as embedded database software 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.<sup>[6]</sup> SQLite has bindings to many programming languages.

The source code for SQLite is in the public domain.<sup>[2]</sup>

## 1 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. 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 multi-tasked, though writes can only be performed sequentially.

Due to the server-less design, SQLite applications require less configuration than client-server databases. SQLite is called *zero-conf*<sup>[7]</sup> because it does not require service management (such as startup scripts) or access control based on GRANT and passwords. Access control is handled by means of File system permissions given to the database file itself. Databases in client-server systems use file system permissions which give access to the database files only to the daemon process.

Another implication of the serverless design is that several processes may need to be able to write to the database

file. In server-based databases, several writers will all connect to the same daemon, which is able to handle its locks internally. SQLite on the other hand has to rely on file-system locks. It has less knowledge of the other processes that are accessing the database at the same time. Therefore SQLite is not the preferred choice for write-intensive deployments.<sup>[8]</sup> However for simple queries with little concurrency, SQLite performance profits from avoiding the overhead of passing its data to another process.

SQLite uses PostgreSQL as a reference platform. “What would PostgreSQL do” is used to make sense of the SQL standard.<sup>[9][10]</sup> One major deviation is that, with the exception of primary keys, SQLite does not enforce type checking; the type of a value is dynamic and not strictly constrained by the schema (although the schema will trigger a conversion when storing, if such a conversion is potentially reversible).

## 2 History

D. Richard Hipp designed SQLite in the spring of 2000 while working for General Dynamics on contract with the United States Navy.<sup>[11]</sup> Hipp was designing software used aboard 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. Hipp based the syntax and semantics on PostgreSQL 6.5 documentation. In August 2000, version 1.0 of SQLite was released, with storage based on gdbm (GNU Database Manager). SQLite 2.0 replaced gdbm with a custom B-tree implementation, adding transaction capability. 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.<sup>[12]</sup>

## 3 Features

SQLite implements most of the SQL-92 standard for SQL but it lacks some features. For example it partially provides triggers, and it can’t write to views (however it

provides INSTEAD OF triggers that provide this functionality). While it provides complex queries, it still has limited **ALTER TABLE** function, as it can't modify or delete columns.<sup>[13]</sup>

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.<sup>[14]</sup> However, it can be implemented with constraints like **CHECK**(typeof(x)='integer').<sup>[11]</sup>

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.<sup>[15]</sup>

As of version 3.8.2 it is possible to create tables without **rowid**.<sup>[16]</sup>

SQLite full **Unicode** function is optional.<sup>[17]</sup>

## 4 Development and distribution

SQLite development stores revisions of its source code in **Fossil**, a **distributed version control system** that is itself built upon an SQLite database.<sup>[18]</sup>

A standalone **command-line** program is provided in SQLite's distribution 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 has automated **regression testing** prior to each release.<sup>[19]</sup> 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. The tests and test harnesses are proprietary.

## 5 Adoption

### 5.1 Programming languages

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

- **AutoIt** <sup>[20]</sup>
- **BASIC**
- **Delphi**
- **C**
- **C#**
- **C++**
- **Clipper//Harbour**
- **Common Lisp**
- **Curl**
- **D**
- **Delphi**
- **Elixir**
- **F#** <sup>[21]</sup>
- **FreeBASIC**
- **Free Pascal**
- **Go**
- **Haskell**
- **Haxe**
- **Java** (on JVM and DVM)
- **JavaScript**<sup>[22]</sup>
- **Julia**
- **Livecode**
- **Lua**
- **newLisp**
- **Objective-C** (on OS X and iOS)
- **OCaml**
- **Perl**<sup>[23]</sup>
- **PHP**
- **Pike**
- **PureBasic**
- **Python**<sup>[24]</sup>

- R
- REALbasic
- REBOL
- Ruby<sup>[25]</sup>
- Scheme
- Smalltalk
- Swift (on OS X and iOS)
- Tcl
- Visual Basic
- Xojo

## 5.2 Middleware

- ADO.NET adapter, initially developed by Robert Simpson, is maintained jointly with the SQLite developers since April 2010.<sup>[26]</sup>
- ODBC driver has been developed and is maintained separately by Christian Werner.<sup>[27]</sup> Werner's ODBC driver is the recommended connection method for accessing SQLite from OpenOffice.org.<sup>[28]</sup>
- COM (ActiveX) wrapper making SQLite accessible on Windows to scripted languages such as JScript and VBScript. This adds database capabilities to HTML Applications (HTA).<sup>[29]</sup>
- XULRunner uses SQLite

## 5.3 Web browsers

- The browsers Google Chrome, Opera, Safari and the Android Browser all allow for storing information in, and retrieving it from, a SQLite database within the browser, using the Web SQL Database technology, although this is rapidly becoming deprecated (namely superseded by IndexedDB).
- 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.

## 5.4 Web application frameworks

- Bugzilla, Mozilla's bug database written in Perl can use SQLite to store data and settings.
- Django, a Python web framework, works with SQLite3 by default.<sup>[30]</sup>

- As of version 7, Drupal, a PHP-based content management system for making websites and blogs, has an option to install using SQLite.<sup>[31]</sup>
- Trac's default database management system is also SQLite.
- Ruby on Rails' default database management system is also SQLite.
- web2py, a Python web framework, default database management system is also SQLite.

## 5.5 Various

- Skype is a widely deployed application that uses SQLite.<sup>[32]</sup>
- Adobe Systems uses SQLite as its file format in Adobe Photoshop Lightroom, a standard database in Adobe AIR, and internally within Adobe Reader.<sup>[33]</sup>
- 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 capability built in to both the desktop and web frameworks.

## 5.6 Operating systems

SQLite is included by default in:<sup>[33]</sup>

- Blackberry's BlackBerry 10 OS
- Microsoft's Windows Phone 8
- 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 used by the core package management system.
- illumos
- Oracle Solaris 10 where the Service Management Facility database is serialized for booting.

- Apple<sup>[34]</sup> 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, and in iOS for storage of text messages on the iPhone.
- Tizen

## 6 Ports

SQLite was ported to JavaScript, in an implementation named `sql.js`. It was done by passing the SQLite source code through the Emscripten compiler.()

## 7 See also

- SpatiaLite
- Comparison of relational database management systems
- List of relational database management systems
- Flat file database
- SQL Server Compact
- H2 (DBMS)
- Berkeley DB

## 8 Citations

- [1] "Recent SQLite News". 2015-07-29. Retrieved 2015-05-29.
- [2] "SQLite Copyright". *sqlite.org*. Retrieved May 17, 2010.
- [3] D. Richard Hipp (presenter) (May 31, 2006). *An Introduction to SQLite* (video). Google Inc. Event occurs at 00:01:14. Retrieved March 23, 2010. [...] *ess-kju-ellipse* [...]
- [4] D. Richard Hipp (presenter) (May 31, 2006). *An Introduction to SQLite*. Google Inc. Event occurs at 00:48:15. Retrieved March 23, 2010. [...] *sequelite* [...]
- [5] Owens, Michael (2006). "Chapter 4: SQL". In Gilmore, Jason; Thomas, Keir. *The Definitive Guide to SQLite*. D. Richard Hipp (foreword), Preston Hagar (technical reviewer). Apress. p. 133. ISBN 978-1-59059-673-9. Retrieved 30 December 2014.
- [6] "Most Widely Deployed SQL Database Estimates". *Sqlite.org*. Retrieved May 11, 2011.
- [7] SQLite web site, *SQLite Is A Zero-Configuration Database*
- [8] SQLite web site, *Appropriate Uses For SQLite*
- [9] <https://lwn.net/Articles/601144/>
- [10] <https://www.pgcon.org/2014/schedule/events/736.en.html>
- [11] Owens, Michael (2006). *The Definitive Guide to SQLite*. Apress. doi:10.1007/978-1-4302-0172-4\_1. ISBN 978-1-59059-673-9.
- [12] "Interview: Richard Hipp on UnQL, a New Query Language for Document Databases". *InfoQ*. August 4, 2011. Retrieved October 5, 2011.
- [13] "SQL Features That SQLite Does Not Implement". *SQLite*. January 1, 2009. Retrieved October 14, 2009.
- [14] "Frequently Asked Questions". *SQLite*. January 26, 2009. Retrieved February 7, 2009.
- [15] "Write Ahead Logging in SQLite 3.7". *SQLite*. Retrieved September 3, 2011. WAL provides more concurrency as readers do not block writers and a writer does not block readers. Reading and writing can proceed concurrently
- [16] *ReleaseLog SQLite.org*, visited 8th December 2013
- [17] "Case-insensitive matching of Unicode characters does not work". *SQLite Frequently Asked Questions*. *Sqlite.org*. Retrieved May 11, 2011.
- [18] "Fossil: Fossil Performance". *Fossil-scm.org*. August 23, 2009. Retrieved September 12, 2009.
- [19] "How SQLite Is Tested". *SQLite*. Retrieved September 12, 2009.
- [20] : SQLite user defined functions Reference
- [21] *SQLProvider* - A general SQL database type provider
- [22] *JSPDO* JavaScript database access abstraction API
- [23] *DBD::SQLite*: Perl DBI Interface to SQLite
- [24] *PySQLite*: Python bindings for SQLite
- [25] *SQLite/Ruby*: Ruby bindings for SQLite
- [26] <http://system.data.sqlite.org/index.html/doc/trunk/www/index.wiki>
- [27] <http://www.ch-werner.de/sqliteodbc/>
- [28] [http://documentation.openoffice.org/HOW\\_TO/data\\_source/SQLite.pdf](http://documentation.openoffice.org/HOW_TO/data_source/SQLite.pdf)
- [29] "sqlite — Sqlite Wrappers". *SQLite*. February 7, 2009. Retrieved February 7, 2009.
- [30] "Databases". *Django Documentation*. Django Software Foundation. Retrieved 13 November 2012.
- [31] "Drupal 7".
- [32] "Skype client using SQLite?". *Mail-archive.com*. August 28, 2007. Retrieved June 14, 2010.
- [33] "Well-Known Users Of SQLite". *sqlite.org*. Retrieved 2015-08-05.
- [34] "Show Download History List of All Files Ever Downloaded Within Mac OS X".

## 9 References

- Allen, Grant; Owens, Mike (November 5, 2010). *The Definitive Guide to SQLite* (2nd ed.). Apress. p. 368. ISBN 1-4302-3225-0.
- Kreibich, Jay A. (August 17, 2010). *Using SQLite* (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)* (1st ed.). Sams. p. 336. ISBN 0-672-32685-X.

## 10 External links

- [Official website](#)
- [SQLite at DMOZ](#)
- [An Introduction to SQLite on YouTube](#)

## 11 Text and image sources, contributors, and licenses

### 11.1 Text

- **SQLite** *Source:* <https://en.wikipedia.org/wiki/SQLite?oldid=678427314> *Contributors:* Damian Yerrick, AxelBoldt, Bryan Derksen, Fnielsen, Fubar Obfusco, Nate Silva, Boleslav Bobcik, Pnm, Wwwwolf, Graue, Haakon, Glenn, Nikai, Wael Ellithy, Tobias Conradi, Timwi, Aquatopia~enwiki, Sanxiyn, Qu1j0t3, Furrykef, Theta682, Topbanana, Betterworld, Hankwang, Chealer, Peak, Rursus, Rrjanbiah, Delpino, Hif, Ruakh, Mattflaschen, David Gerard, Wjgilmore, Risk one, Sdfisher, AlistairMcMillan, Eequor, Brockert, Dave2, Mag-icmike, Neile, Tgwena, Dnas, Two Bananas, Abdull, Moxfyre, RandalSchwartz, Kate, SimonEast, Rfl, RossPatterson, Rich Farmbrough, Pmsyyz, Drano, Mecanismo, Smyth, Scullder, TOR, \*drew, Mattisgoo, Kwamikagami, Egrabczewski, Slicky, Giraffedata, DG~enwiki, Sleske, Minghong, Tablizer, Ronark, Stephan Leeds, Gpvos, KTC, Mindmatrix, ChrisNoe, Shello, DanBishop, Bkkbrad, Scjessey, Apokrif, Choas~enwiki, GregorB, Waldir, Cbyneorne, Havarhen, Turnstep, Rgbea, Yoghurt, Magister Mathematicae, Qwertys, RadioActive~enwiki, Philipolson, Gudeldar, Gilesmorant, Beeveer, Fred Bradstadt, Zero0w, FlaBot, Jstaniek, Lmatt, Alvin-cs, Visor, Bg-white, YurikBot, Argav, Barefootguru, Kimchi.sg, Toehead2001, Mipadi, Mschumacher69, Irishguy, Sekelsenmat, Bota47, Paul Magnusson, H@r@ld, Mainstreetmark, CWenger, JLaTondre, Wikimercenary, Luk, Lethalmonk, SmackBot, Reedy, Optikos, DStoykov, Thumperward, Mdwh, DHN-bot~enwiki, Ned Scott, TripleF, Huji, PotatoEater, Compctech, Frap, Arulprasad, JonathanWakely, Mwtoews, Where, Ohconfucius, ThurnerRupert, Korval, Javier Odom, Rorx, Soumyasch, Heelmijnlevenlang, Jrouquie, A. Parrot, Makyen, Mchirico, Hu12, Theone00, Saltlakejohn, FatalError, J Milburn, SqlPac, Paolosupernova, AndrewHowse, Cydebot, Roberta F., Mrinal.kant~enwiki, Ebrahim, Jayapalchandran, Thijs!bot, DivideByZero14, Qwyrxian, DmitTrix, Srprez, SpacePacket, Jaxelrod, Gioto, MBParker, Isilanes, Temblast, Dougher, Lfstevens, VictorAnyakin, Drhipp, Deflective, Od1n, Inverse.chi, Greensburger, Alex LE, Zhongchen, Who.am.i, Aulis2003, Otterfan, Beethoven05, Sappy, Cander0000, Gwern, Arcturis, Eugenwpg, RobertCailliau, AVRS, MartinBot, Devin Asay, Yaron K., Wikitiki89, Sarathbs, Arite, Grshiplett, Cooldude7273, Daxx wp, Jaguar83, Kriplozoik~enwiki, Volkris, SexyBern, MBlakley, TXiKiBoT, FFMG, Wykypydy, Pascal sa, DavidHalko, Darkzen.bps, Hareesh.t, Cmbeelby, Hello71, Svick, Scartboy, Nnemo, K14m-AWB, Stachelfisch, Bigown, Niceguyedc, Fenke, Melizg, Bbb2007, Djbriddock, Streambag, Steph1978~enwiki, Amanda hoic, Johnuniq, SF007, DumZiBoT, XLinkBot, Zbig Lebowsky, C. A. Russell, Jmkim dot com, Addbot, Nuno Brito, AkhtaBot, 0x54097DAA, MrOllie, LaaknorBot, Ginosbot, Staticmain, Bro4, Jorel314, Yobot, Legobot II, Specious, Wonderfl, AnomieBOT, Götz, Therealgeeves, Cecilia-Pang, MaterialsScientist, Anybody, Neurolysis, ArthurBot, Xtremejames183, LilHelpa, Xqbot, PabloCastellano, Baest, LordArtemis, GrouchoBot, RibotBOT, U1024, Canadianacademic, Molyneux.peter, FrescoBot, Gerardo cabero~enwiki, Wikiwikiwu, Glider87, Sae1962, Gabrielepx, Kwiki, Winterst, Simple Bob, Alex degarate, Tim baroon, Jeanfarias, Hoo man, Jandalhandler, Salvidrim!, Latios, Zo8shong, Chris Caven, Ungzd, Dchestnykh, Lotje, Rapha222, Ybungalobill, Rro4785, Leonelhs, Saint1997, Kirbylover4000, TankMiche, Xaxn1234, Najeeb1010, MVelliste, Philippe.petrinko, Madooo12, Palosirkka, Noncoercive, Quadunit404, Vlops, Dexp, Many delicious meats, Dsimpson dcsi, Marco1713, Danim, Helpful Pixie Bot, BG19bot, Gurghe, Mark Arsten, Patuck, ZeaForUs, OzOle, Xephyrous, Marco1317, Vdvluc, JamesHaigh, Samyulg, Captain Conundrum, Syntaxerrorz, Marvinella, ScotXW, Claw of Slime, Permafrost46, Donosauro, EfiHerbst, Fabricio Correa~enwiki, SuperCookyMan, Maffeinet and Anonymous: 372

### 11.2 Images

- **File:Ambox\_important.svg** *Source:* [https://upload.wikimedia.org/wikipedia/commons/b/b4/Ambox\\_important.svg](https://upload.wikimedia.org/wikipedia/commons/b/b4/Ambox_important.svg) *License:* Public domain *Contributors:* Own work, based off of Image:Ambox scales.svg *Original artist:* Dsmurat (talk · contribs)
- **File:Commons-logo.svg** *Source:* <https://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg> *License:* ? *Contributors:* ? *Original artist:* ?
- **File:Computer-aj\_aj\_ashton\_01.svg** *Source:* [https://upload.wikimedia.org/wikipedia/commons/c/c1/Computer-aj\\_aj\\_ashton\\_01.svg](https://upload.wikimedia.org/wikipedia/commons/c/c1/Computer-aj_aj_ashton_01.svg) *License:* CC0 *Contributors:* ? *Original artist:* ?
- **File:Crystal\_Clear\_device\_cdrom\_unmount.png** *Source:* [https://upload.wikimedia.org/wikipedia/commons/1/10/Crystal\\_Clear\\_device\\_cdrom\\_unmount.png](https://upload.wikimedia.org/wikipedia/commons/1/10/Crystal_Clear_device_cdrom_unmount.png) *License:* LGPL *Contributors:* All Crystal Clear icons were posted by the author as LGPL on kde-look; *Original artist:* Everaldo Coelho and YellowIcon;
- **File:Free\_Software\_Portal\_Logo.svg** *Source:* [https://upload.wikimedia.org/wikipedia/commons/6/67/Nuvola\\_apps\\_emacs\\_vector.svg](https://upload.wikimedia.org/wikipedia/commons/6/67/Nuvola_apps_emacs_vector.svg) *License:* LGPL *Contributors:*
- Nuvola\_apps\_emacs.png *Original artist:* Nuvola\_apps\_emacs.png: David Vignoni

### 11.3 Content license

- Creative Commons Attribution-Share Alike 3.0