

what's sqlite?

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly. A complete SQLite database is stored in a single cross-platform disk file. SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted. SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.

SQLite history?

2000 - D. Richard Hipp designed SQLite for the purpose of no administration required for operating a program.
 2000 - In August, SQLite 1.0 released with GNU Database Manager.
 2011 - Hipp announced to add UNQL interface to SQLite DB and to develop UNQLite (Document oriented database).

Limitation

Only Left out join	no right/full outer join
alter table [add column]	no drop column, alter column,add constraint
trigger for each row	not for each statement
view read only	no insert/update/delete on a view
grant and revoke on OS file level	

installtion

1 download binary: <https://www.sqlite.org/download.html>

3 Create a folder C:\>sqlite and unzip above two zipped files in this folder, which will give you sqlite3.def, sqlite3.dll and sqlite3.exe files.

4 Add C:\>sqlite in your PATH environment variable and finally go to the command prompt and issue sqlite3 command

```
Linux: sqlite3
$tar xvfz sqlite-autoconf-3071502.tar.gz
$cd sqlite-autoconf-3071502
$./configure --prefix=/usr/local
$make
$make install
```

SQLite command

```
DDL: create alter drop
DML insert update delete
Query: select
dot cmd (no ;)
.help                .table ?pattern?
.schema              .databases
.schema sqlite_master
.header on
.width n1,n2,n3
```

cmd list

```
.archive ... Manage SQL archives
.auth ON|OFF Show authorizer callbacks
.backup ?DB? FILE Backup DB (default "main") to FILE
.bail on|off Stop after hitting an error. Default OFF
.binary on|off Turn binary output on or off. Default OFF
.cd DIRECTORY Change the working directory to DIRECTORY
.changes on|off Show number of rows changed by SQL
.check GLOB Fail if output since .testcase does not match
.clone NEWDB Clone data into NEWDB from the existing database
.databases List names and files of attached databases
.dbconfig ?op? ?val? List or change sqlite3_db_config() options
.dbinfo ?DB? Show status information about the database
.dump ?TABLE? ... Render all database content as SQL
.echo on|off Turn command echo on or off
.eqp on|off|full|... Enable or disable automatic EXPLAIN QUERY PLAN
.excel Display the output of next command in a spreadsheet
.exit ?CODE? Exit this program with return-code CODE
.expert EXPERIMENTAL. Suggest indexes for specified queries
.fullschema ?--indent? Show schema and the content of sqlite_stat tables
.headers on|off Turn display of headers on or off
.help ?-all? ?PATTERN? Show help text for PATTERN
.import FILE TABLE Import data from FILE into TABLE
.imposter INDEX TABLE Create imposter table TABLE on index INDEX
```



cmd list (cont)

```
.indexes ?TABLE? Show names of indexes
.limit ?LIMIT? ?VAL? Display or change the value of an SQLITE-
_LIMIT
.lint OPTIONS Report potential schema issues.
.load FILE ?ENTRY? Load an extension library
.log FILE|off Turn logging on or off. FILE can be stderr/stdout
.mode MODE ?TABLE? Set output mode
.nullvalue STRING Use STRING in place of NULL values
.once (-e|-x|FILE) Output for the next SQL command only to FILE
.open ?OPTIONS? ?FILE? Close existing database and reopen FILE
.output ?FILE? Send output to FILE or stdout if FILE is omitted
.parameter CMD ... Manage SQL parameter bindings
.print STRING... Print literal STRING
.progress N Invoke progress handler after every N opcodes
.prompt MAIN CONTINUE Replace the standard prompts
.quit Exit this program
.read FILE Read input from FILE
.restore ?DB? FILE Restore content of DB (default "main") from FILE
.save FILE Write in-memory database into FILE
.scanstats on|off Turn sqlite3_stmt_scanstatus() metrics on or off
.schema ?PATTERN? Show the CREATE statements matching
PATTERN
.selftest ?OPTIONS? Run tests defined in the SELFTEST table
.separator COL ?ROW? Change the column and row separators
.sha3sum ... Compute a SHA3 hash of database content
.shell CMD ARGS... Run CMD ARGS... in a system shell
.show Show the current values for various settings
.stats ?on|off? Show stats or turn stats on or off
.system CMD ARGS... Run CMD ARGS... in a system shell
.tables ?TABLE? List names of tables matching LIKE pattern TABLE
.testcase NAME Begin redirecting output to 'testcase-out.txt'
.timeout MS Try opening locked tables for MS milliseconds
.timer on|off Turn SQL timer on or off
.trace ?OPTIONS? Output each SQL statement as it is run
.vfsinfo ?AUX? Information about the top-level VFS
```

cmd list (cont)

```
.vfslst List all available VFSes
.vfsname ?AUX? Print the name of the VFS stack
.width NUM1 NUM2 ... Set column widths for "column" mode
```

Syntax

```
case insensitive in genera;\           GLOB and glob not same

space and comments: - //

statement ;

analyze [database_name.[table_name]

alter table [add column|rename to ...]

attach database 'dbname' as 'alias'     DETACH DATABASE 'Alias-
name'                                     Name';

begin [exclusive transaction;

commit;

CREATE TRIGGER database_name BEFORE INSERT ON
table_name FOR EACH ROW begin ....end.trigger_name

CREATE VIRTUAL TABLE database_name.table_name USING
weblog( access.log );

EXPLAIN INSERT statement...;

EXPLAIN QUERY PLAN SELECT statement...;

WHERE column_name GLOB { PATTERN };

PRAGMA table_info(table_name);

PRAGMA cache_size = 1024;

RELEASE savepoint_name;

ROLLBACK TO SAVEPOINT savepoint_name;

VACUUM;
```

data type

```
null

integer

real

text      UTF-8, UTF-16BE or UTF-16LE

blob      a blob of data, stored exactly as it was input

Type affinity
```



By **Jianmin Feng (taotao)**
cheatography.com/taotao/

Not published yet.
 Last updated 6th June, 2019.
 Page 2 of 5.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish
 Yours!
<https://apollopad.com>

data type (cont)

text (null, text, blob)

numeric (all 5)

integer (all 5, casting exception)

real(all 5, force integer to float)

NONE no coerce

boolean(0,1)

date: text, real (days since noon, 11/24/4714BC),integer (second from 1970-0-01 UTC)

first scope

sqlite3 test.db sqlite>

.databases main: C:\Users\jmfeng\test.db

sqlite3 test.db .dump >testdb.sql

sqlite3 testdb.db<testdb.sql

sqlite3 attach database 'test.db' as 'test';

database main and temp reserved for primary database and temp database (temp objects)

create table ... ; .tables .schema table_name

insert into t (...) values(...)

insert into select

select ... from where .header on | .mode column|.width
..... #, #, #...

operator: (+-*/%) (== != <> > < >= <= !< !>) (and between exists in not in like GLOB not or is null is is not || unique) (& | ~ << >>)

expression (values + operators + functions)

select current_timestamp; select 3+2;

LIKE % _

GLOBE ? [1] * [0,n] GLOBE '200';GLOBE '2??3'

LIMIT [no of row] offset [row num]

advanced sqlite - pragma

pragma pragma_name=value; set environmental var and state flag

pragma database.auto_vacuum=0 None, full or incremental shrink
[1|2] file

cache_size=pages; default 2000, minimal 10 pages

PRAGMA case_sensitive_like = [true|false]; for LIKE operator,default false

PRAGMA count_changes = [true|false]; DML return, default false

PRAGMA database_list;

PRAGMA encoding = format; control string encoding UTF-8, UTF-16le, or UTF-16be

PRAGMA [database.]freelist_count;

PRAGMA database.journal_mode =delete|truncate|persist|memory|off;

PRAGMA [database.]max_page_count = max_page;The default value is 1,073,741,823 which is one giga-page, default 1K/page, total 1TB size

PRAGMA [database.]page_count;

PRAGMA [database.]page_size = bytes; 0.5k~32k, set the page size and then vacuum it immediately to change the database size

PRAGMA parser_trace = [true|false]; default false

PRAGMA [database.]schema_version = number; keep DDL count

PRAGMA database.secure_delete = [true|false]; default false

PRAGMA sql_trace = [true|false];

PRAGMA [database.]synchronous 0-OFF 1-noreml 2-full
=

PRAGMA temp_store = mode; 0 default|file, 1 file, 2 memory

PRAGMA temp_store_directory = 'directory_path';

PRAGMA [database.]user_version = number;

PRAGMA writable_schema = [true|false]; system table sqlite_ will be able modified



By **Jianmin Feng** (taotao)
cheatography.com/taotao/

Not published yet.
Last updated 6th June, 2019.
Page 3 of 5.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>

advanced sqlite

not null, default, unique, primary, CHECK(SALARY > 0)

Cross join | inner join | left outer join

Union | union all (duplicates)

CREATE TRIGGER trigger_name [BEFORE|AFTER] UPDATE OF column_name ON table_name BEGIN ... END;

DML; FOR EACH ROW; old. new.; when; before after; on table_name; raise()

SELECT name FROM sqlite_master WHERE type = 'trigger';

DROP TRIGGER trigger_name;

CREATE INDEX index_name ON table_name;

CREATE UNIQUE INDEX index_name on table_name (column_name_list);

implicit index on primary key and unique key

DROP INDEX salary_index;

avoid indexes (small table, DML freq, a lot null | freq manipulate)

SELECT|DELETE|UPDATE column1, column2... INDEXED BY (index_name) table_name WHERE (CONDITION);

ACID: Atomicity Consistency Isolation Durability

BEGIN TRANSACTION

COMMIT

ROLLBACK

integer autoincrement

advanced sqlite - injection

```
$name = "Qadir"; DELETE FROM users;";
@ $db-> query( " SELECT * FROM users WHERE
username = '{$name}' ");
if (preg_match( " /^ \w{8,20} $/ ", $_GET[ 'user_name' ], $matches )) {
    $db = new SQLite Database( 'filename' );
    $result = @ $db-> query( " SELECT * FROM
users WHERE username = $matches[ 0 ] );
} else {
    echo " username not accepted ";
}
if (get_magic_quotes_gpc()) {
    $name = sqlite_escape_string( $name );
}
$result = @ $db-> query( " SELECT * FROM users
WHERE username = '{$name}' ");
```

explain, vacuum

```
EXPLAIN QUERY PLAN [SQLite Query]
VACUUM command rebuilds the database file from scratch, cleans the main database by copying its contents to a temporary database file and reloading the original database file from the copy. used for cleaning free pages, align data to be contiguous, clean data structure. rowid changed, unless integer primary key explicitly. main db only, not on attached. fail if active tx, no-op for in memory db
$sqlite3 database_name " VACUUM ; "
sqlite> VACUUM;
sqlite> VACUUM table_name;
sqlite> PRAGMA auto_vacuum = NONE; -- 0 means disable auto vacuum
sqlite> PRAGMA auto_vacuum = FULL; -- 1 means enable full auto vacuum
sqlite> PRAGMA auto_vacuum = INCREMENTAL; -- 2 means enable incremental vacuum
$sqlite3 database_name " PRAGMA auto_vacuum ; "
```

date times

date(timestring, modifiers...)	YYYY-MM-DD
time(timestring, modifiers...)	HH:MM:SS
datetime(timestring, modifiers...)	
julianday(timestring, modifiers...)	
strftime(timestring, modifiers...)	
YYYY-MM-DD HH:MM:SS.SSS now	
%d %f %H %j %J %m %M %s %S %w %W %Y %%	
SELECT date('now');	2013-05-07
SELECT date('now','start of month','+1 month','-1 day');	2013-05-31
SELECT datetime(1092941466, 'unixepoch');	2004-08-19 18:51:06
SELECT datetime(1092941466, 'unixepoch', 'localtime');	2004-08-19 13:51:06
SELECT strftime('%s','now');	1393348134
SELECT julianday('now') - julianday('1776-07-04');	86798.709-4695023

date times (cont)

```
SELECT strftime('%s','now') - strftime('%s','-
2004-01-01 02:34:56');
```

```
SELECT date('now','start of year','+9
months','weekday 2'); first duesday in oct
2013-10-01
```

```
SELECT time('12:00', 'localtime'); 05:00:00
```

```
SELECT time('12:00', 'utc'); 19:00:00
```

useful functions

count, max, min avg,sum, abs

random -9223372036854775808 and +9223372036854775807.

upper, lower,length,substr,instr, replace

sqlite_version()

java and sqlite

```
https://www.tutorialspoint.com/sqlite/sqlite_java.htm
1 download jar
2 add to build path
import java.s ql.*;
public class SQLiteJDBC {
    public static void main( String args[] ) {
        Con nection c = null;

        try {
            Cla ss.f or Nam e("o rg.s ql -
ite.JD BC");
            c = Driver Man age r.g etC onn ect -
ion ("jd bc: sql ite :te st.d b");
        } catch ( Exception e ) {
            Sys tem.er r.p rintln( e.getC las -
s().ge tName() + ": " + e.getM ess age() );
            Sys tem.ex it(0);
        }
        Sys tem.ou t.p rin tln ("Opened database
succes sfu lly ");
    }
}
```

