

SQLite

Nov 14, 2011



<http://www.sqlite.org/>
<http://www.sqlite.org/docs.html>

What is it?

SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world. The source code for SQLite is in the public domain.

from <http://www.sqlite.org/>

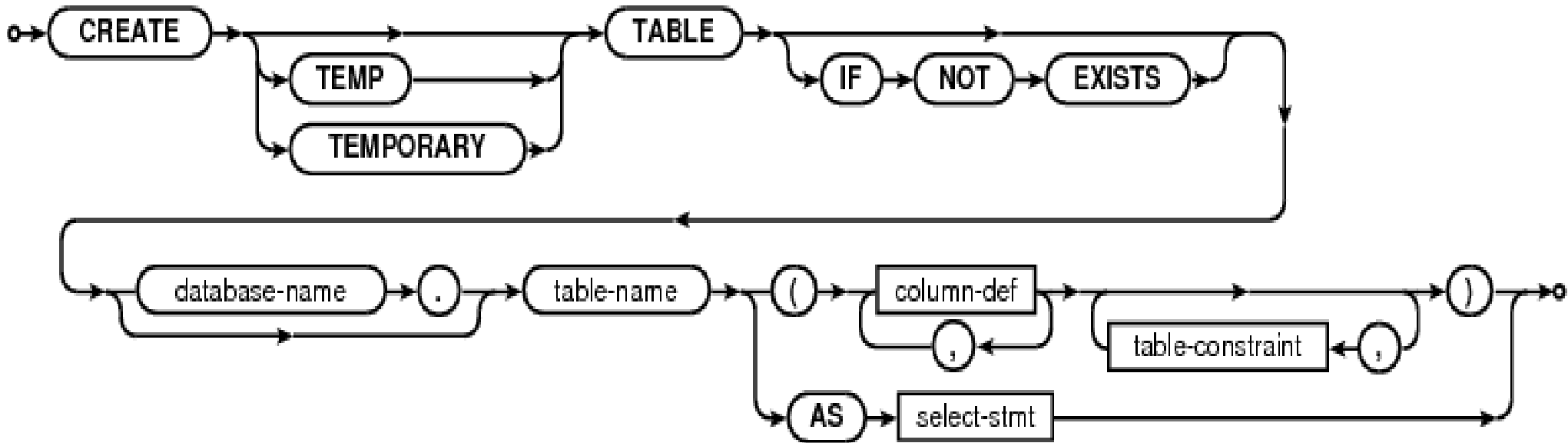
- What does all that mean?

MySQL vs SQLite Server vs Serverless

- When to use each?
- Strengths of MySQL
- Strengths of SQLite

SQL

- <http://www.sqlite.org/syntaxdiagrams.html>
- create-table-stmt



- column-def.



Command Line

```
chadd@zeus:~> sqlite3 movies.db
```

```
SQLite version 3.6.23
```

```
Enter ".help" for instructions
```

```
Enter SQL statements terminated with a ";"
```

```
sqlite> CREATE TABLE Movies
```

```
...> (MovieID INTEGER NOT NULL,
```

```
...> TITLE TEXT(512) NOT NULL,
```

```
...> PRIMARY KEY (MovieID));
```

```
sqlite> INSERT INTO Movies VALUES (1, "Star Wars");
```

```
sqlite> INSERT INTO Movies VALUES (2, "Empire");
```

```
sqlite> SELECT * FROM Movies;
```

```
1|Star Wars
```

```
2|Empire
```

```
sqlite> .quit
```

Not Implemented

- <http://www.sqlite.org/omitted.html>
 - As compared to SQL92
- Right & Full Outer Join
- Complete Alter Table
- Complete Trigger
- Writing to Views
- Grant & Revoke

Data Types

- <http://www.sqlite.org/datatype3.html>
- INTEGER (signed, at most 8 bytes)
- REAL (8-byte IEEE floating point)
- TEXT (UTF-8, UTF-16BE or UTF-16LE)
- BLOB
- NULL

- Date & Time
 - no date and time
 - store as
 - TEXT “YYYY-MM-DD HH:MM:SS.SSSS”
 - REAL (number days since noon in Greenwich on November 24, 4714 B.C. proleptic Gregorian calendar)
 - INTEGER (Unix Time: the number of seconds since 1970-01-01 00:00:00 UTC)

Type Affinity

- What is the recommended data type for a column?
 - TEXT
 - null, text, blob
 - NUMERIC
 - any of the 5 types
 - INTEGER
 - same as numeric
 - REAL
 - like numeric, but forces ints to floats
 - NONE
 - wildcard

Example

```
CREATE TABLE t1(  
  t TEXT,      -- text affinity by rule 2  
  nu NUMERIC,  -- numeric affinity by rule 5  
  i INTEGER,   -- integer affinity by rule 1  
  r REAL,     -- real affinity by rule 4  
  no BLOB      -- no affinity by rule 3  
);  
  
-- Values stored as TEXT, INTEGER, INTEGER, REAL, TEXT.  
INSERT INTO t1 VALUES('500.0', '500.0', '500.0', '500.0', '500.0');  
SELECT typeof(t), typeof(nu), typeof(i), typeof(r), typeof(no) FROM t1;  
text|integer|integer|real|text  
  
-- Values stored as TEXT, INTEGER, INTEGER, REAL, REAL.  
DELETE FROM t1;  
INSERT INTO t1 VALUES(500.0, 500.0, 500.0, 500.0, 500.0);  
SELECT typeof(t), typeof(nu), typeof(i), typeof(r), typeof(no) FROM t1;  
text|integer|integer|real|real  
  
-- Values stored as TEXT, INTEGER, INTEGER, REAL, INTEGER.  
DELETE FROM t1;  
INSERT INTO t1 VALUES(500, 500, 500, 500, 500);  
SELECT typeof(t), typeof(nu), typeof(i), typeof(r), typeof(no) FROM t1;  
text|integer|integer|real|integer  
  
-- BLOBs are always stored as BLOBs regardless of column affinity.  
DELETE FROM t1;  
INSERT INTO t1 VALUES(x'0500', x'0500', x'0500', x'0500', x'0500');  
SELECT typeof(t), typeof(nu), typeof(i), typeof(r), typeof(no) FROM t1;  
blob|blob|blob|blob|blob
```

Comparison Example

```
CREATE TABLE t1(  
    a TEXT,          -- text affinity  
    b NUMERIC,      -- numeric affinity  
    c BLOB,         -- no affinity  
    d               -- no affinity  
);  
  
-- Values will be stored as TEXT, INTEGER, TEXT, and INTEGER respectively  
INSERT INTO t1 VALUES('500', '500', '500', 500);  
SELECT typeof(a), typeof(b), typeof(c), typeof(d) FROM t1;  
text|integer|text|integer  
  
-- Because column "a" has text affinity, numeric values on the  
-- right-hand side of the comparisons are converted to text before  
-- the comparison occurs.  
SELECT a < 40,    a < 60,    a < 600 FROM t1;  
0|1|1  
  
-- Text affinity is applied to the right-hand operands but since  
-- they are already TEXT this is a no-op; no conversions occur.  
SELECT a < '40', a < '60', a < '600' FROM t1;  
0|1|1  
  
-- Column "b" has numeric affinity and so numeric affinity is applied  
-- to the operands on the right. Since the operands are already numeric,  
-- the application of affinity is a no-op; no conversions occur. All  
-- values are compared numerically.  
SELECT b < 40,    b < 60,    b < 600 FROM t1;  
0|0|1
```

Vacuum



http://www.sqlite.org/lang_vacuum.html

- Rebuild entire Database
 - No Server!
 - fragmented database
 - lots of empty space in the database file

PHP!

```
<?php
    if ($db = sqlite_open('movies')) {

        $query = sqlite_query($db,
            'SELECT MovieID, Title FROM
            Movies
            WHERE MovieID > 1');

        while( $row = $q->sqlite_fetch_array($query) )
        {
            print $row['Title'] . " " . $row['MovieID'];
        }
sqlite_close($db);
?>
```

<http://www.php.net/manual/en/book.sqlite.php>
<http://www.php.net/manual/en/function.sqlite-fetch-array.php>

C/C++ interface

```
#include <sqlite3.h>

sqlite3 *psDB;
sqlite3_stmt *psStmt;

sqlite3_open_v2(..., &psDB, ...); // open the DB

sqlite3_prepare_v2(..., &psStmt, ...); // build a Query

sqlite3_bind_int(psStmt, ...); // set a parameter

while ( sqlite3_step(psStmt) == SQLITE_ROW ) // retrieve one row
{
    sqlite3_column_int(psStmt, ...); //retrieve one column
    sqlite3_column_text(psStmt, ...); //retrieve one column

    // do something useful
}

sqlite3_finalize(psStmt);
sqlite3_close(psDB);
```

Java/Android

<http://developer.android.com/reference/android/database/sqlite/package-summary.html>

package

android.database.sqlite

Since: API Level 1

Interfaces

SQLiteCursorDriver	A driver for SQLiteCursors that is used to create them and gets notified by the cursors it creates on significant events in their lifetimes.
SQLiteDatabase.CursorFactory	Used to allow returning sub-classes of Cursor when calling query.
SQLiteTransactionListener	A listener for transaction events.

Classes

SQLiteClosable	An object created from a SQLiteDatabase that can be closed.
SQLiteCursor	A Cursor implementation that exposes results from a query on a SQLiteDatabase .
SQLiteDatabase	Exposes methods to manage a SQLite database.
SQLiteOpenHelper	A helper class to manage database creation and version management.
SQLiteProgram	A base class for compiled SQLite programs.
SQLiteQuery	A SQLite program that represents a query that reads the resulting rows into a CursorWindow .
SQLiteQueryBuilder	This is a convenience class that helps build SQL queries to be sent to SQLiteDatabase objects.
SQLiteStatement	A pre-compiled statement against a SQLiteDatabase that can be reused.

Other DB connection techniques

- Open Database Connectivity
 - ODBC
 - JDBC
 - Each database may have their own set of connectors
 - <http://dev.mysql.com/downloads/connector/odbc/5.1.html>
 - <http://dev.mysql.com/downloads/connector/j/5.1.html>

