

SQLITE OVERVIEW

http://www.tutorialspoint.com/sqlite/sqlite_overview.htm

Copyright © tutorialspoint.com

This tutorial helps you to understand what is SQLite, how it differs from SQL, why it is needed and the way in which it handles the applications Database.

SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is one of the fastest-growing database engines around, but that's growth in terms of popularity, not anything to do with its size. The source code for SQLite is in the public domain.

What is SQLite?

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is the one database, which is zero-configured, that means like other database you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. The SQLite accesses its storage files directly.

Why SQLite?

- SQLite does not require a separate server process or system to operate *serverless*.
- SQLite comes with zero-configuration, which means no setup or administration needed.
- 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 is self-contained, which means no external dependencies.
- SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
- SQLite supports most of the query language features found in the SQL92 *SQL2* standard.
- SQLite is written in ANSI-C and provides simple and easy-to-use API.
- SQLite is available on UNIX *Linux, MacOS - X, Android, iOS* and Windows *Win32, WinCE, WinRT*.

History:

- 2000 -- D. Richard Hipp had 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 *Documentorienteddatabase*.

SQLite Limitations:

There are few unsupported features of SQL92 in SQLite which are shown below:

Feature	Description
RIGHT OUTER JOIN	Only LEFT OUTER JOIN is implemented.
FULL OUTER JOIN	Only LEFT OUTER JOIN is implemented.

ALTER TABLE	The RENAME TABLE and ADD COLUMN variants of the ALTER TABLE command are supported. The DROP COLUMN, ALTER COLUMN, ADD CONSTRAINT not supported.
Trigger support	FOR EACH ROW triggers are supported but not FOR EACH STATEMENT triggers.
VIEWS	VIEWS in SQLite are read-only. You may not execute a DELETE, INSERT, or UPDATE statement on a view.
GRANT and REVOKE	The only access permissions that can be applied are the normal file access permissions of the underlying operating system.

SQLite Commands:

The standard SQLite commands to interact with relational databases are similar as SQL. They are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into groups based on their operational nature:

DDL - Data Definition Language:

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

DML - Data Manipulation Language:

Command	Description
INSERT	Creates a record
UPDATE	Modifies records
DELETE	Deletes records

DQL - Data Query Language:

Command	Description
SELECT	Retrieves certain records from one or more tables