SQLite

Content

- What is SQLite?
- Getting Started
- SQLite Studio

Content

- What is SQLite?
- Getting Started
- SQLite Studio

What is SQLite?

SQLite

Small. Fast. Reliable.
Choose any three.

Home About Documentation Download License Support Purchase Search

About SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. Think of SQLite not as a replacement for Oracle but as a replacement for fopen()

SQLite is a compact library. With all features enabled, the library size can be less than 500KiB, depending on the target platform and compiler optimization settings. (64-bit code is

Executive Summary

- Full-featured SQL
- Billions and billions of deployments
- Single-file database
- Public domain source code
- All source code in one file (sqlite3.c)
- Small footprint
- Max DB size: 140 terabytes (2^{47} bytes)
- Max row size: 1 gigabyte
- Faster than direct file I/O
- Aviation-grade quality and testing
- Zero-configuration
- ACID transactions, even after power loss

Content

- What is SQLite?
- Getting Started
- SQLite Studio

Getting Started?

In 5 minutes or less

https://www.sqlite.org/quickstart.html

1. Download The Code
   - Get a copy of the prebuilt binaries for your machine
2. Create A New Database
   - At a shell or DOS prompt, enter: "sqlite3 test.db". This will create a new database named "test.db“
   - Enter SQL commands at the prompt to create and populate the new database
3. Write Programs That Use SQLite

```c
#include <stdio.h>
#include <sqlite3.h>

static int callback(void *NotUsed, int argc, char **argv, char **azColName){
  int i;
  for(i=0; i<argc; i++){
    printf("%s = %s\n", azColName[i], argv[i] ? argv[i] : "NULL");
  }
  printf("\n");
  return 0;
}
int main(int argc, char **argv){
  sqlite3 *db;
  char *zErrMsg = 0;
  int rc;

  if( argc!=3 ){
    fprintf(stderr, "Usage: %s DATABASE SQL-STATEMENT\n", argv[0]);
```

Limitations

SQLite supports a limited subset of **ALTER TABLE**.

It is not possible to rename a column, remove a column, or add/remove constraints from a table.

Workaround:

1. create new table (with applied changes)
2. copy all data
3. drop old table
4. rename the new one

Content

- What is SQLite?
- Getting Started
- SQLite Studio

SQLite Studio

https://github.com/pawelsalawa/sqlitestudio/releases