무료 전자 책

배우기

sqlite

Free unaffiliated eBook created from
Stack Overflow contributors.

#sqlite

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: sqlite

It is an unofficial and free sqlite ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official sqlite.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: sqlite

3.0		2004-06-18
3.7.11	SELECT max (x), y	2012-03-20
3.8.3	CTE	2014-02-11

Examples

SQLite `sqlite3.c` C .

(Perl , Python , Ruby) (: Android) SQLite . SQLite .

SQL (`sqlite3` `sqlite3.exe`) . Linux . Windows , `sqlite-tools` .

SQLite . .

sqlite : https://riptutorial.com/ko/sqlite/topic/1753/sqlite-

2: PRAGMA

SQLite PRAGMA .

Examples

PRAGMA

PRAGMA . , .

PRAGMA ( ).

- application_id
- WAL journal_mode
- schema_version
- user_version
- wal_checkpoint

PRAGMA .

- auto_vacuum (VACUUM)
-
- legacy_file_format
- page_size (VACUUM)

:

```
-- open a connection to a not-yet-existing DB file
PRAGMA page_size = 4096;
PRAGMA auto_vacuum = INCREMENTAL;
CREATE TABLE t(x);              -- database is created here, with the above settings
```

PRAGMA : https://riptutorial.com/ko/sqlite/topic/5223/pragma-

3: sqlite3_stmt : (C API)

: Prepared Statement

Examples

sqlite3_prepare_v2 () .

sqlite3_finalize () . .

sqlite3_bind_xxx () .

sqlite3_step () .

```
const char *sql = "INSERT INTO MyTable(ID, Name) VALUES (?, ?)";
sqlite3_stmt *stmt;
int err;

err = sqlite3_prepare_v2(db, sql, -1, &stmt, NULL);
if (err != SQLITE_OK) {
    printf("prepare failed: %s\n", sqlite3_errmsg(db));
    return /* failure */;
}

sqlite3_bind_int (stmt, 1, 42);                          /* ID */
sqlite3_bind_text(stmt, 2, "Bob", -1, SQLITE_TRANSIENT); /* name */

err = sqlite3_step(stmt);
if (err != SQLITE_DONE) {
    printf("execution failed: %s\n", sqlite3_errmsg(db));
    sqlite3_finalize(stmt);
    return /* failure */;
}

sqlite3_finalize(stmt);
return /* success */;
```

SELECT . sqlite3_step () . .

- SQLITE_ROW :
- SQLITE_DONE :
- .

SQLITE_DONE .

sqlite3_column_xxx () .

```
const char *sql = "SELECT ID, Name FROM MyTable";
sqlite3_stmt *stmt;
int err;

err = sqlite3_prepare_v2(db, sql, -1, &stmt, NULL);
```

```
if (err != SQLITE_OK) {
    printf("prepare failed: %s\n", sqlite3_errmsg(db));
    return /* failure */;
}

for (;;) {
    err = sqlite3_step(stmt);
    if (err != SQLITE_ROW)
        break;

    int         id   = sqlite3_column_int (stmt, 0);
    const char *name = sqlite3_column_text(stmt, 1);
    if (name == NULL)
        name = "(NULL)";
    printf("ID: %d, Name: %s\n", id, name);
}

if (err != SQLITE_DONE) {
    printf("execution failed: %s\n", sqlite3_errmsg(db));
    sqlite3_finalize(stmt);
    return /* failure */;
}

sqlite3_finalize(stmt);
return /* success */;
```

sqlite3_reset () .

.

```
const char *sql = "INSERT INTO MyTable(ID, Name) VALUES (?, ?)";
sqlite3_stmt *stmt;
int err;

err = sqlite3_prepare_v2(db, sql, -1, &stmt, NULL);
if (err != SQLITE_OK) {
    printf("prepare failed: %s\n", sqlite3_errmsg(db));
    return /* failure */;
}

for (...) {
    sqlite3_bind_int (stmt, 1, ...);   /* ID */
    sqlite3_bind_text(stmt, 2, ...);   /* name */

    err = sqlite3_step(stmt);
    if (err != SQLITE_DONE) {
        printf("execution failed: %s\n", sqlite3_errmsg(db));
        sqlite3_finalize(stmt);
        return /* failure */;
    }

    sqlite3_reset(stmt);
}

sqlite3_finalize(stmt);
return /* success */;
```

sqlite3_stmt : (C API) : https://riptutorial.com/ko/sqlite/topic/5456/sqlite3-stmt------c-api-

4:

: SQLite 3

Examples

TYPEOF

```
sqlite> SELECT TYPEOF(NULL);
null
sqlite> SELECT TYPEOF(42);
integer
sqlite> SELECT TYPEOF(3.141592653589793);
real
sqlite> SELECT TYPEOF('Hello, world!');
text
sqlite> SELECT TYPEOF(X'0123456789ABCDEF');
blob
```

SQLite `0` `1` .

```
sqlite> SELECT 2 + 2 = 4;
1
sqlite> SELECT 'a' = 'b';
0
sqlite> SELECT typeof('a' = 'b');
integer
```

```
> CREATE TABLE Users ( Name, IsAdmin );
> INSERT INTO Users VALUES ('root', 1);
> INSERT INTO Users VALUES ('john', 0);
> SELECT Name FROM Users WHERE IsAdmin;
root
```

SQLite .

```
> CREATE TABLE Test (
      Col1 INTEGER,
      Col2 VARCHAR(2),       -- length is ignored, too
      Col3 BLOB,
      Col4,                  -- no type required
      Col5 FLUFFY BUNNIES    -- use whatever you want
  );
> INSERT INTO Test VALUES (1, 1, 1, 1, 1);
> INSERT INTO Test VALUES ('xxx', 'xxx', 'xxx', 'xxx', 'xxx');
> SELECT * FROM Test;
1   1   1   1   1
xxx xxx xxx xxx xxx
```

.

typeof () .

```
CREATE TABLE Tab (
    Col1 TEXT   CHECK (typeof(Col1) = 'text' AND length(Col1) <= 10),
    [...]
);
```

('null' 'null' .)

/

SQLite .

ISO8601

CURRENT_DATE , CURRENT_TIME CURRENT_TIMESTAMP ISO8601 .

```
> SELECT CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP;
CURRENT_DATE  CURRENT_TIME  CURRENT_TIMESTAMP
------------  ------------  -------------------
2016-07-08    12:34:56      2016-07-08 12:34:56
```

/ .

```
> SELECT strftime('%Y', '2016-07-08');
2016
```

/ .

```
> SELECT datetime(2457578.02425926);
2016-07-08 12:34:56
```

julianday() / julianday() .

```
> SELECT julianday('2016-07-08 12:34:56');
2457578.02425926
```

/ unixepoch Unix .

```
> SELECT datetime(0, 'unixepoch');
1970-01-01 00:00:00
```

strftime() / Unix .

```
> SELECT strftime('%s', '2016-07-08 12:34:56');
```

```
1467981296
```

/ / NULL .

```
> SELECT time('1:30:00');   -- not two digits

> SELECT datetime('8 Jul 2016');
□
```

: https://riptutorial.com/ko/sqlite/topic/5252/-

5:

`sqlite3` (SQLite). : `sqlite3`

Examples

SQL

.

```
sqlite> .output mydatabase_dump.sql
sqlite> .dump
```

.

```
sqlite> .output mytable_dump.sql
sqlite> .dump mytable
```

`.dump` `.output` . .

.

```
sqlite> .read mytable_dump.sql
```

: https://riptutorial.com/ko/sqlite/topic/3789/---

S. No		Contributors
1	sqlite	CL., Community, e4c5, H. Pauwelyn
2	PRAGMA	CL., springy76
3	sqlite3_stmt : (C API)	CL.
4		CL.
5		CL., e4c5, James Toomey, Lasse Vågsæther Karlsen, ravenspoint, Thinkeye