無料電子ブック

学習

sqlite

Free unaffiliated eBook created from
Stack Overflow contributors.

#sqlite

1
1: sqlite 2
2
Examples 2
2
2
2: PRAGMA 3
3
Examples 3
PRAGMA 3
3: sqlite3_stmtC API 4
4
Examples 4
4
4
5
4: 7
7
Examples 7
SQL 7
5: 8
8
Examples 8
TYPEOF 8
8
8
/ 9
ISO8601 9
9
Unix 9
10

11

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をいめる

バージョン

バージョン	な	
3.0		20040618
3.7.11	SELECT maxx、y	2012-03-20
3.8.3	CTE	2014-02-11

Examples

インストール

SQLiteは、、のバージョンのソースコードをダウンロードし、 `sqlite3.c`ファイルをプロジェクトにすることによって、アプリケーションにコンパイルされるCライブラリです。

くのスクリプト Perl 、 Python 、 Rubyなどやフレームワーク Androidなど はSQLiteをサポートしています。これは、インストールするのないSQLiteライブラリのビルトインコピーでわれます。

SQLのテストでは、コマンドラインシェル `sqlite3`または`sqlite3.exe` をするとです。すでにほとんどのLinuxディストリビューションにされています。 Windowsでは、コンパイルされたバイナリを`sqlite-tools`パッケージにダウンロードし 、どこかでします。

ドキュメンテーション

SQLiteにはすでになドキュメントがありますが 、ここではしないようにしてください。

オンラインでsqliteをいめるをむ https://riptutorial.com/ja/sqlite/topic/1753/sqliteをいめる

2: PRAGMAの

SQLiteのドキュメントには、 すべてのPRAGMAのリファレンスがあります 。

Examples

をするPRAGMA

ほとんどのPRAGMAステートメントは、のデータベースにのみします。つまり、データベースがかれたときにするがあります。

ただし、のPRAGMAはデータベースファイルにきみをい、いつでもできますただし、トランザクションではないもあります。

- アプリケーションID
- WALモードをまたはにするときのjournal_mode
- schema_version
- user_version
- wal_checkpoint

のPRAGMAは、にできないデータベースファイルのプロパティをするため、データベースへののののきみのにするがあります。

- auto_vacuum VACUUMのにすることもできます
- エンコーディング
- legacy_file_format
- page_size VACUUMのにすることもできます

えば

```
-- 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
```

オンラインでPRAGMAのをむ https://riptutorial.com/ja/sqlite/topic/5223/pragmaの

3: sqlite3_stmtプリペアドステートメントC API

ドキュメント Prepared Statementオブジェクト

Examples

ステートメントの

ステートメントは、 sqlite3_prepare_v2などのでされます。

プリペアドステートメントオブジェクトは 、 sqlite3_finalizeでクリーンアップするがあります。エラーのは、このことをれないでください。

パラメータをするは、 sqlite3_bind_xxxをしてをします。

のは、 sqlite3_stepがびされたときにされます。

```
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 */;
```

カーソルからのデータのみり

SELECTクエリはのステートメントとにされます。されたデータをみるには、ループでsqlite3_stepをびします。それはす

- SQLITE_ROWののデータがかどうか、または
- SQLITE_DONEそれがない、または
- のエラーコード。

せがをさないは、のでSQLITE_DONEがされます。

のからデータをみるには、 sqlite3_column_xxxをびします。

```
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 */;
```

されたステートメントをする

がされた、 sqlite3_resetをびすと、のにり、されます。

、ステートメントはじままですが、パラメーターはされます。

```
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 */;
```

オンラインでsqlite3_stmtプリペアドステートメントC APIをむ
https://riptutorial.com/ja/sqlite/topic/5456/sqlite3-stmt-プリペアドステートメント-c-api-

4: コマンドラインのドットコマンド

き

sqlite3コマンドラインシェルは、のコマンドセットをしていますSQLiteライブラリをするプログラムではできません。ドキュメント sqlite3へのなコマンド

Examples

をSQLスクリプトとしてエクスポートおよびインポートする

データベースのエクスポートは、な2のプロセスです。

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

テーブルをエクスポートするのはかなりています

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

.dumpをするにファイルを.outputでするがあります。それの、テキストはにされます。

インポートはさらにです。

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

オンラインでコマンドラインのドットコマンドをむ https://riptutorial.com/ja/sqlite/topic/3789/コマンドラインのドットコマンド

5: データ

SQLiteバージョン3のデータ

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
```

ブールの

ブールの、SQLiteは0と1います

```
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
```

のをする

SQLiteはをし、されたをします。

```
> 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
```

ただし、されたはのにされます。

をするには、 typeofでをするがあります 。

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

そのようなを'null'にするがあるは、に'null'するがあります。

/の

SQLiteには、またはのにのデータはありません。

ISO8601

ビルトインキーワード`CURRENT_DATE` 、 `CURRENT_TIME` 、および`CURRENT_TIMESTAMP`は、ISO8601のをします。

```
> 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
```

このようなは、 みみの/によってもされます 。

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

ジュリアンの

みみの/は、をユリウスとしてします。

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

`julianday()`は、サポートされているの/をユリウスにします。

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

Unixタイムスタンプ

みみの/は、 `unixepoch`をしてをUnixタイムスタンプとしてできます。

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

`strftime()`は、サポートされているの/のをUnixのタイムスタンプにできます。

```
> SELECT strftime('%s', '2016-07-08 12:34:56');
1467981296
```

サポートされていないフォーマット

/のをのでデータベースにすることはですが、みみの/はそれらをせず、NULLをします。

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

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

オンラインでデータをむ https://riptutorial.com/ja/sqlite/topic/5252/データ

クレジット

S. No		Contributors
1	sqliteをいめる	CL., Community, e4c5, H. Pauwelyn
2	PRAGMAの	CL., springy76
3	sqlite3_stmtプリペアドステートメントC API	CL.
4	コマンドラインのドットコマンド	CL., e4c5, James Toomey, Lasse Vågsæther Karlsen, ravenspoint, Thinkeye
5	データ	CL.