

# 第1回. リレーショナルデータベースを使ってみよう

- SQLite のインストール
- SQLite コマンドライン・インタフェースの起動と終了
- SQLite データベースの新規作成
- SQL を用いた**テーブル定義**

# SQLite とは

- **軽量 (Lite) である**

- パスワード, 権限の機能が無い.
- サーバクライアントでない. 通信がない.
- 設定なしで動く

- **SQL に関する数多くの機能がある**

索引, トランザクション, ビュー

- **オープンソース**

利用条件 : <http://www.sqlite.com/copyright.html>

オープンソースのマナー

利用条件は必ず確認.

剽窃しないこと. 無許可で改変して再配布しないこと.

# 事前準備. SQLite コマンドライン・インタフェースのダウンロード

- ① SQLite の Web ページを開く.

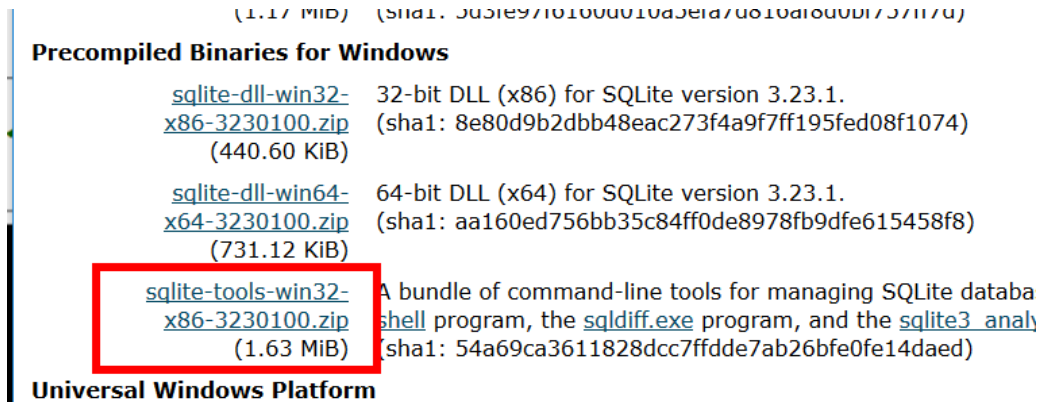
<http://www.sqlite.org/>

# 事前準備. SQLite コマンドライン・インタフェースのダウンロード

## ② 「Download」 をクリック

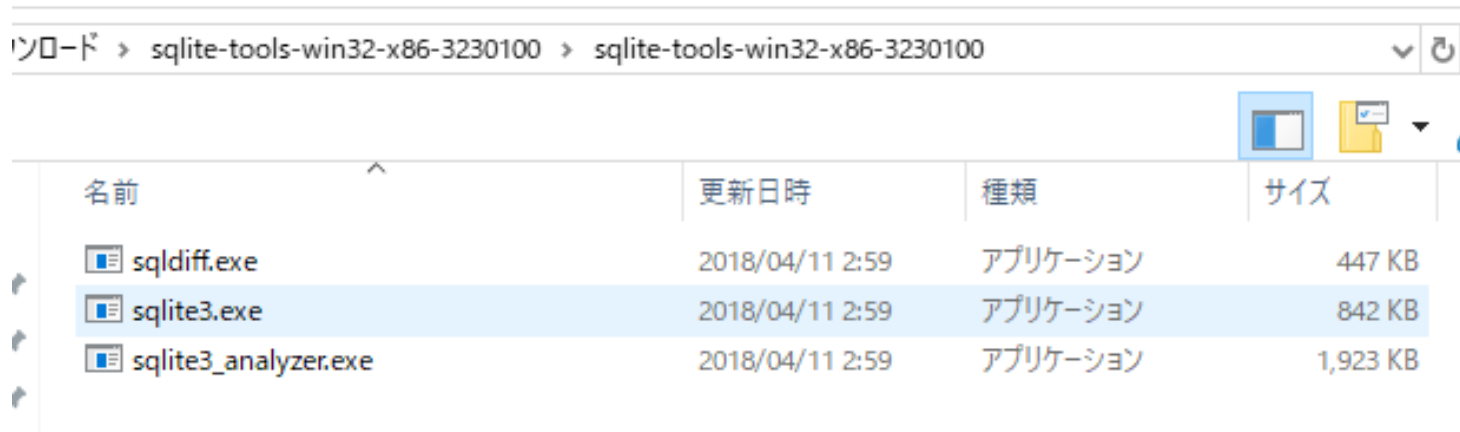


## ③ Windows 版のダウンロード



# 事前準備. SQLite コマンドライン・インタフェースのダウンロード

## ④ ダウンロードした .zip ファイルを展開（解凍）



名前	更新日時	種類	サイズ
sqldiff.exe	2018/04/11 2:59	アプリケーション	447 KB
sqlite3.exe	2018/04/11 2:59	アプリケーション	842 KB
sqlite3_analyzer.exe	2018/04/11 2:59	アプリケーション	1,923 KB

sqlite3.exe を使う

# SQLite データベースの新規作成

データベース論理名: C:¥SQLite¥mydb  
で, SQLite データベースの新規作成

- ① 前もって Windows で **C:¥SQLite** というディレクトリ（フォルダ）を作成しておく
- ② sqlite3.exe を実行

名前	更新日時	種類
soldiff.exe	2018/04/11 2:59	アプリケーション
sqlite3.exe	2018/04/11 2:59	アプリケーション
sqlite3_analyzer.exe	2018/04/11 2:59	アプリケーション

- ③ 新しい画面が開くので確認

```
e:¥Downloads¥sqlite-tools-win32-x86-3230100¥sqlite-tools-win32-x86-3230100¥sqlite3.exe
SQLite version 3.23.1 2018-04-10 17:39:29
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

# SQLite データベースの新規作成

データベース論理名: C:¥SQLite¥mydb  
で, SQLite データベースの新規作成

④ 「.open --new」で, SQLite データベースの**新規作成**

```
.open --new C:¥¥SQLite¥¥mydb
```

```
sqlite> .open --new C:¥¥SQLite¥¥mydb  
sqlite>
```

- ・このファイルが**すでに存在するとき**は, **いったん削除**され, 空のファイルが**新規作成**される
- ・このファイルが**存在しないとき**は, 新しい SQLite データベースのファイルが**新規作成**される

↑ > PC > ローカルディスク (C:) > SQLite

名前	更新日時	種類	サイズ
mydb	2018/05/07 22:54	ファイル	0 KB

# SQL を用いたテーブル定義

テーブル P(id, name, weight) の定義

## 【SQL プログラム】

```
create table P (  
  id integer primary key not null,  
  name text,  
  weight real );
```

e:\Downloads\sqlite-tools-win32-x86-3230100\sqlite-tools-v

```
SQLite version 3.23.1 2018-04-10 17:39:29  
Enter ".help" for usage hints.  
Connected to a transient in-memory database.  
Use ".open FILENAME" to reopen on a persistent database.  
sqlite> .open C:\SQLite\mydb  
sqlite> create table P (  
  ...> id integer primary key not null,  
  ...> name text,  
  ...> weight real );  
sqlite>
```

## データ型

空値 **NULL**

テキスト（文字列） **char, text**

数値 **integer, real**

日付や時刻など **datetime**

ブール値 **bool** など

**primary key not null**

**主キー**であり、**NULL** になることは**ない**（**非空**）という**制約**の指定



# SQL を用いたテーブルの削除

## テーブル P の削除

### 【SQL プログラム】

```
drop table P;
```

```
e:\Downloads\sqlite-tools-win32-x86-3230100\sqlite-tools-v
SQLite version 3.23.1 2018-04-10 17:39:29
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open C:\SQLite\mydb
sqlite> create table P (
...>   id integer primary key not null,
...>   name text,
...>   weight real );
sqlite> drop table P;
sqlite>
```

# SQLite コマンドライン・インタフェースの終了

次のコマンドを実行

```
.exit
```

```
sqlite> .exit
```

# 主キーと一貫性制約について

## 【SQL プログラム】

```
CREATE TABLE score_records (  
  name TEXT NOT NULL,  
  score INTEGER NOT NULL CHECK ( score >= 0 AND score <=100 ),  
  student_name TEXT NOT NULL,  
  created_at DATETIME NOT NULL,  
  UNIQUE (name, student_name) );
```

### UNIQUE (name, student\_name)

{name, student\_name} が**主キー**.

SQLite で**複数属性が主キーになる**ときの書き方

### NOT NULL

レコードが変更される時「NULLでないこと」をチェック

### CHECK ( score >= AND score <= 100 )

レコードが変更される時「値が 0 以上 100 以下であること」をチェック

# 終わりに

Web ページ: <http://www.kkaneko.jp/dblab/intro/>

その他, さまざまな資料を金子研究室 Web ページで公開  
しています

<http://www.kkaneko.jp/index-j.html>

謝辞 : SQLite の作者に感謝します