# SQLite3の使い方

## データベースとは

データベースというと「表」をイメージしますが、ここでいうデータベースはいくつもの表を管理するシステムをいいます。

データベースでは「表」を「テーブル(TABLE)」といいます。

## SQLite3の特徴

１つのコンピュータの中にデータベースを作ります。ユーザーはそのコンピュータを使用している人です。

## MySQLの特徴

サーバー・クライアント方式です。ネット上にMySQLサーバーを置いて、複数のコンピュータからアクセスすることが可能です。この場合、ユーザーが作成され、各ユーザーにはパスワードが発行されます。

ログイン後、各ユーザーはデータベースを作成できますが、他のユーザーの作成したデータベースにはアクセスできないように設定してあるのが望ましいやり方です。

XAMPPの場合、各ローカルマシンにMySQLがインストールされますが、サーバーもクライアントも両方インストールされます。つまり、自分のコンピュータでMySQLサーバーを動かし、それにユーザーとパスワードを作成して、ログイン処理をおこなって、作業をするということになります。

## データベースの作成

XAMPPコントロールパネルの「shell」をクリックして、シェルプロンプトを開きます。

以下のコマンドを入力します。

# sqlite3 test.db

test.dbというデータベースが自動的に作成されて、sqlite3が起動します。

もし、test.dbが存在していたら、そのデータベースが選択されて起動します。

## テーブルの作成

**CREATE TABLE テーブル名(カラム名, ...) ;**

テーブル（表）を作成します。以下のような表を考えます。

|  |  |  |  |
| --- | --- | --- | --- |
| ID | name | age | address |
| 1 | Tanaka | 34 | Nara |
| 2 | Suzuki | 25 | Osaka |
| 3 | Kimura | 41 | Kyoto |

以下のコマンドを入力します。(各カラムは 「,」(カンマ)で区切る）

sqlite> create table person ( id, name, age, address ) ;

テーブルができたことを確認します。

sqlite> .tables

person というテーブルができたことが確認できます。

ついでにテーブルの削除のしかたに触れておきます。

sqlite> create table test ( id, name ) ;

sqlite> .tables

table

sqlite> drop table test ;

これでテーブルは削除されています。

## データの入力

**INSERT INTO テーブル名 VALUES ( データ, ... ) ;**

データを入力してみます。

sqlite> insert into person values ( 1, 'Tanaka', 34, 'Nara' ) ;

1行入力すると、次は、キーボードの↑キーで、以前に入力した行が表示されるので、Backspaceキーで末尾から削除して、適当なところから入力すると楽です。

sqlite> insert into person values ( 2, 'Suzuki', 25, 'Osaka' ) ;

sqlite> insert into person values ( 3, 'Kimura', 41, 'Kyoto' ) ;

## データの表示

**SELECT \* FROM テーブル名;**

入力したデータを表示してみます。次のコマンドを入力してください。

sqlite> select \* from person;

以下のように表示されます。

id = 1

name = Tanaka

age = 34

address = Nara

id = 2

name = Suzuki

age = 25

address = Osaka

id = 3

name = Kimura

age = 41

address = Kyoto

表示方法を変えてみます。

次のようにコマンドを入力してください。

sqlite> .mode column

sqlite> .header on

先ほどと同じようにテーブルを表示します。

sqlite> select \* from person;

id name age address

---------- ---------- ---------- ----------

1 Tanaka 34 Nara

2 Suzuki 25 Osaka

3 Kimura 41 Kyoto

## カラムを指定して入力

**INSERT INTO テーブル名(カラム名) VALUES(値);**

あるいは

**INSERT INTO テーブル名(カラム名1, カラム名2, ...) VALUES(値1, 値2, ...);**

次のデータを入力します。

|  |  |  |  |
| --- | --- | --- | --- |
| ID | name | age | address |
| 4 | Saito | 38 |  |

address欄はまだ未定です。次のコマンドを入力します。

sqlite> insert into person (id, name, age)

...> values (4, 'Saito', 38);

テーブルを見てみます。

sqlite> select \* from person;

id name age address

---------- ---------- ---------- ----------

1 Tanaka 34 Nara

2 Suzuki 25 Osaka

3 Kimura 41 Kyoto

4 Saito 38

## データの更新

**UPDATE テーブル名 SET カラム名 = データ WHERE 条件式 ;**

その後、Saitoさんのaddressが判明したので、修正します。

sqlite> update person set address = 'Hyogo' where id = 4;

sqlite> select \* from person;

id name age address

---------- ---------- ---------- ----------

1 Tanaka 34 Nara

2 Suzuki 25 Osaka

3 Kimura 41 Kyoto

4 Saito 38 Hyogo

## データの並べ替え

**SELECT \* FROM テーブル名 ORDER BY カラム名 [ASC | DESC] ;**

年齢の昇順に並べかえて表示します。

sqlite> select \* from person order by age asc;

id name age address

---------- ---------- ---------- ----------

2 Suzuki 25 Osaka

1 Tanaka 34 Nara

4 Saito 38 Hyogo

3 Kimura 41 Kyoto

## データの抽出

特定の文字列を含むデータを取得する

**SELECT \* FROM テーブル名 WHERE カラム名 LIKE パターン;**

addressがnaraの人を抽出します。

sqlite> select \* from person where address like 'nara';

id name age address

---------- ---------- ---------- ----------

1 Tanaka 34 Nara

条件に合うデータをそのカラムだけ表示する

**SELECT カラム名 FROM テーブル名 WHERE 条件式;**

ageが30以上の人をidとnameとage で表示します。

sqlite> select id, name, age from person where age >= 30 ;

id name age

---------- ---------- ----------

1 Tanaka 34

3 Kimura 41

4 Saito 38

## カラムのデータ型を指定してテーブルを作成する

SQLite3は、各カラムのデータ型を指定しなくても、入力されたデータをみて、SQLite3が適切なデータ型に変換してくれます。

しかし、データ型を指定したい場合もあります。

今度は、テーブルの作成時に、各カラムのデータ型を指定してみます。

次のようなデータを考えてみます。

|  |  |  |  |
| --- | --- | --- | --- |
| ID | name | age | height |
| 1 | Taro | 13 | 143.5 |
| 2 | Hanako | 15 | 153.0 |
| 3 | Jiro | 17 | 170.3 |

テーブルを作成するコマンドはこのようになります。

**CREATE TABLE テーブル名 ( カラム名 データ型 , ...... ) ;**

カラムのデータ型には以下のようなものが指定できます。

|  |  |
| --- | --- |
| TEXT | 文字列 |
| NUMERIC | その他（日付・真偽値など） |
| INTEGER | 整数 |
| REAL | 浮動小数点型 |
| NONE | 指定なし（入力データそのまま） |

また、id については、次のような指定を加えてみます。

**PRIMARY KEY** …… 重複した値は指定できない。データ型が INTEGER の場合は連番が設定される。

**AUTOINCREMENT** …… 一度使われた値は二度使われない

そこで、以下のようにテーブルを作成してみます。各行はEnterキーで改行してください。

sqlite> create table student (

...> id integer primary key autoincrement ,

...> name text ,

...> age integer ,

...> height real );

データを入力します。

sqlite> insert into student (name, age, height) values('Taro', 13, 143.5);

sqlite> insert into student (name, age, height) values('Hanako', 15, 153.0);

sqlite> insert into student (name, age, height) values('Jiro', 17, 170.3);

データを見てみます。

sqlite> select \* from student;

id name age height

---------- ---------- ---------- ----------

1 Taro 13 143.5

2 Hanako 15 153.0

3 Jiro 17 170.3

自動で id が入力されています。また、小数点もはいっています。

**SQLite3 主なコマンド**

|  |  |
| --- | --- |
| .databases | データベース一覧 |
| .tables | テーブル一覧 |
| .quit | SQLite3を終了 |
| .exit | SQLite3を終了 |
| .help | ヘルプ |
| .show | 設定一覧 |
| .mode list | 区切り文字でカラムを区切り1行で出力（デフォルト） |
| 〃 line | 各カラムごとに行を分けて出力 |
| 〃 column | カラムごとに左揃えで出力 |
| .header on | カラム名（見出し）を出力 |
| 〃 off | カラム名（見出し）を出力しない |

参考URL

SQLiteコマンドの使い方 <https://www.dbonline.jp/sqlite/sqlite_command/>

SQLite入門 <https://www.dbonline.jp/sqlite/>

# 付録：MySQLでデータベースを作成する

## ユーザーを作成する

XAMPPコントロールパネルのshellを開きます。

MySQLにRoot でログインします。

# mysql -u root -p

Enter password:

パスワードは設定していないはずですから、そのままEnterキーをおします。

次のコマンドを入力して、どんなデータベースがあるのか確認しておきます。

mysql> show databases;

ユーザーを追加します。このユーザーに管理させるデータベース名を決めておきます。

今回は、testdb というデータベース名にしておきます。このデータベース名がないことを確認しておいてください。

ユーザー名をtestuser。パスワードも testuser にしておきます。

（ユーザー名とパスワードを同じにしておいたら、パスワードを忘れることはありません。セキュリティ上は好ましくないですが、開発用のデータベースであれば、大事なデータは登録しないので、大丈夫です）

|  |  |
| --- | --- |
| データベース名 | testdb |
| ユーザー名 | testuser |
| パスワード | testuser |

以下のコマンドを入力します。

mysql> grant all on testdb.\* to 'testuser'@'localhost' identified by 'testuser';

Query OKと表示されたら、うまくいっています。

ログアウトします。

mysql> quit

Bye

## データベースを作成する

先ほど作成したユーザーでログインします。パスワードはユーザー名と同じです。

# mysql -u testuser -p

Enter password:

どんなデータベースがあるか見てみます。

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

+--------------------+

information\_schema は、もとからあるデータベースです。つまり、まだ何も作っていません。

では、データベースを作成します。

mysql> create database testdb;

Query OK, 1 row affected (0.00 sec)

これでデータベースが作成できています。

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| testdb |

+--------------------+