

SQLite は手軽に利用できる軽量なデータベースです。一般的な DBMS がデータベースサーバで動くミドルウェアで複数のクライアントからのアクセスを制御するのに対し、SQLite は単なる C 言語のライブラリです。直感的には、SQLite 形式のデータファイルに SQL を使ってアクセスすると考えると良いでしょう。ここでは、本授業の演習で使う程度のごく基本的な SQLite の使い方を紹介します。

■SQLite の起動

```
% sqlite3 <データベースファイル名>
```

起動すると

```
sqlite>
```

という sqlite 用プロンプトが表示される。

■SQL の入力

sqlite 用プロンプトで SQL を入力する。SQL の終わりに「;」をつける。

それまでは改行しても実行されない

```
sqlite> select *
...> from students;
g12345|山田花子|20|a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
g9420542|渡辺知恵美|10|807b94ccfa282089f2a436f9166a22fb9f9a0490
g1111111|テスト 試子|20|a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
...
sqlite>
```

■SQLite の終了

sqlite プロンプトで exit と入力する。はじめに「.」と入力するのが SQLite 用コマンド。

```
sqlite> .exit
```

■定義されているテーブル一覧を表示する

```
sqlite> .tables
```

■テーブルのスキーマ情報を表示する

```
sqlite> .schema テーブル名
```

■SQL をファイルから読み込んで実行する。

sqlite プロンプトで SQL 文を間違えると、実行しなおして結構面倒である。そこで、emacs などのエディタで SQL を記述してファイル保存し、それを sqlite プロンプトで読み込んで実行させるのが便利である。また SQL 文に日本語が含まれているときは sql プロンプトではうまくかけない場合があるので、特に便利。

SQL 文をファイルから読み込むには read コマンドを使って

```
sqlite> .read ファイル名
```

と入力する。

例) ファイル test.sql に以下の問合せを書いて保存する。

```
select * from students where grade='3';
```

sql プロンプトでこのファイルを読み込むには

```
sqlite> .read test.sql
```

と入力すると実行される。

- 結果出力をするときに入力された SQL 文も出力する。

```
sqlite> .echo on
```

.read コマンドを使うと, SQL 文がプロンプト上に表示されないので, 何の問合せに対して結果が出たのか分かりにくくなります. そこで, .echo コマンドを使って SQL 文を出力させるようにすると分かりやすくなるでしょう.

- 出力結果をみやすくする

最初の設定では問合せ結果は「SQL の入力」で示したように非常に単純で時に見辛い. そのため, 問合せをする前に以下の sqlite コマンドを入力して出力フォーマットをきれいにしよう.

- .header : ヘッダ (属性名) を表示するか設定する(on), または設定しない(off)

```
sqlite> .header on
```

- .mode : 出力モードを設定する

column	属性値を左揃えで出力
list	一行ごと出力
csv	カンマ区切りで出力
tabs	タブ区切りで出力

例) sqlite> .header on

```
sqlite> .mode column
```

```
sqlite> select std,name,grade from students where grade=20;
```

```
std      name      grade
-----
g12345    山田花子  20
g1111111 テスト    20
```