

2. SQL 問い合わせ (SQLite3, Python を使用)

データベース演習

URL: <http://www.kkaneko.jp/cc/dbenshu/index.html>

概要

前半は、テーブル定義でのデータ型と主キーの指定を行う。後半は、SQL 問い合わせの結果を、カーソルで扱う。

Python と SQL を使う。

■ SQL 問い合わせ (SQL Query)

SQL 問い合わせは、リレーショナルデータベースに格納された 1 つまたは複数のテーブルを使う。例えば、次のように書くと、 m 個のテーブル T_1, T_2, \dots, T_m の直積集合から条件 $\langle expression \rangle$ を満足する行のみを選び、そうして出来たテーブルの属性 A_1, A_2, \dots, A_n を出力するという意味になる。

```
SELECT A1, A2, ..., An
FROM   T1, T2, ..., Tm
WHERE  <expression>
```

■ 条件 (Condition)

テーブル名とドット「.」と列名の並びを列の修飾名と呼ぶ。例えば、テーブル R の列 A の修飾名は $R.A$ である。文字定数は、**'X'** のように、シングルクォーテーションマーク「'」で囲む。比較演算子は $=, >, <, >=, <=, <>$ の 6 種類がある。探索条件は「**R.A > 20**」のように、列の修飾名と比較演算子と定数の並びである。扱うテーブルが 1 つのときは、列の修飾名の代わりに列名を使うことができる（「**A > 20**」のように）。条件は **and** や **or** で連結することができる。

準備

1. Python のインストール

Windows での Python のインストールは次のページで説明している.

<https://www.kkaneko.jp/tools/win/python.html>

2. pandas のインストール

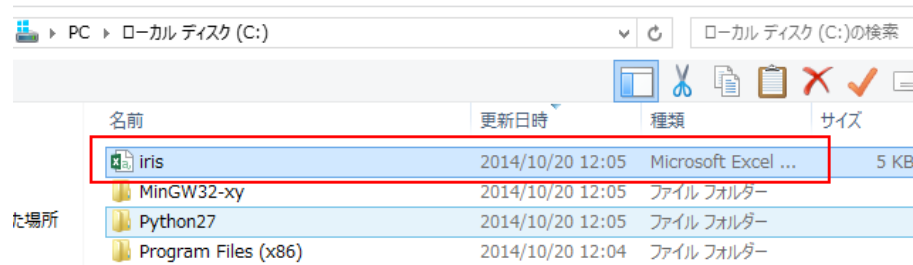
Windows のコマンドプロンプトを管理者として開き, 次のコマンドを実行.

pip install -U pandas

CSV ファイル iris.csv のダウンロード

① Web ブラウザで <https://www.kkaneko.jp/sample/iris.csv> を開き、ダウンロードする

② ダウンロードされたファイル **iris.csv** を **C:¥iris.csv にコピー**



③ iris.csv は次のようなファイルである.

	A	B	C	D	E	F	G
1	id	sepal_leng	sepal_wid	petal_leng	petal_wid	species	
2	0	5.1	3.5	1.4	0.2	setosa	
3	1	4.9	3	1.4	0.2	setosa	
4	2	4.7	3.2	1.3	0.2	setosa	
5	3	4.6	3.1	1.5	0.2	setosa	
6	4	5	3.6	1.4	0.2	setosa	
7	5	5.4	3.9	1.7	0.4	setosa	
8	6	4.6	3.4	1.4	0.3	setosa	
9	7	5	3.4	1.5	0.2	setosa	
10	8	4.4	2.9	1.4	0.2	setosa	
11	9	4.9	3.1	1.5	0.1	setosa	
12	10	5.4	3.7	1.5	0.2	setosa	
13	11	4.8	3.4	1.6	0.2	setosa	
14	12	4.8	3	1.4	0.1	setosa	
15	13	4.3	3	1.1	0.1	setosa	
16	14	5.8	4	1.2	0.2	setosa	
17	15	5.7	4.4	1.5	0.4	setosa	

テーブル定義

① Python を起動する

`python`

② `os.getcwd()` により カレントディレクトリを確認.

`import os`
`os.getcwd()`

```
C:\Users\user>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.getcwd()
'C:\Users\user'
>>>
```

③ **テーブル定義**を行う, 次の Python プログラムを実行

データ型の指定は, 各列で, `integer`, `real` などのデータ型を指定している.

`id` は主キーであるので「`primary key`」を指定している.

※ 複数行にわたるユニコード文字列を使いたいところでは、「`u'''' . . . ''''`」のように書く.

```
import pandas as pd
import sqlite3
c = sqlite3.connect('hoge.sqlite')
sql = u''''
create table iris (
    id integer      primary key,
    sepal_length    real,
    sepal_width     real,
    petal_length    real,
    petal_width     real,
    species         integer );
''''
c.execute(sql)
```

```
C:\Users\user>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.getcwd()
'C:\Users\user'
>>> import pandas as pd
>>> import sqlite3
>>> c = sqlite3.connect('hoge.sqlite')
>>> sql = u''''
... create table iris (
...     id integer      primary key,
...     sepal_length    real,
...     sepal_width     real,
...     petal_length    real,
...     petal_width     real,
...     species         integer );
... ''''
>>> c.execute(sql)
<sqlite3.Cursor object at 0x000001F00C7131F0>
>>>
```

- ④ **テーブル生成** (空のテーブルにレコードを挿入) を行う, 次の Python プログラムを実行
プログラム中の「?」は **SQL プレースホルダー** である.

```
x = pd.read_csv('c:¥¥iris.csv')
for index, r in x.iterrows():
    sql = u"insert into iris values (?, ?, ?, ?, ?, ?)"
    c.execute(sql, (r[0], r[1], r[2], r[3], r[4], r[5]))
c.commit()
```

```
<sqlite3.Cursor object at 0x000001F00C7131F0>
<sqlite3.Cursor object at 0x000001F00C713570>
<sqlite3.Cursor object at 0x000001F00C7132D0>
<sqlite3.Cursor object at 0x000001F00C7131F0>
<sqlite3.Cursor object at 0x000001F00C713570>
<sqlite3.Cursor object at 0x000001F00C7132D0>
<sqlite3.Cursor object at 0x000001F00C7131F0>
>>> c.commit()
>>>
```

- (1) オブジェクト **x** に CSV ファイルを読み込む

```
x = pd.read_csv('c:¥¥iris.csv')
```

- Windows でのファイル名「**C:¥¥iris.csv**」は、Python のプログラム中では「**'C:¥¥iris.csv'**」のように書く
- 読み込みたい CSV ファイルの先頭行に、「id, sepal_length, sepal_width, petal_length, petal_width, species」のようなヘッダーが**ない**場合には「header=None」を付ける
(今回は、ヘッダーがあるので、「header=None」を付けない)

- (2) オブジェクト **x** に格納されたデータを iris テーブルに挿入する

```
for index, r in x.iterrows():
    sql = u"insert into iris values (?, ?, ?, ?, ?, ?)"
    c.execute(sql, (r[0], r[1], r[2], r[3], r[4], r[5]))
```

- 「**for r in x:**」は x の各行について繰り返すという Python プログラム
- 「**insert into iris values**」は、テーブルに 1 行挿入するという SQL プログラム
- 「**?**」は、SQL プレースホルダー
- 「**r[0], r[1], r[2], r[3], r[4], r[5]**」は、もとの CSV データファイルの 0 列目、1 列目、2 列目、3 列目、4 列目、5 列目を使うという意味

- ⑤ テーブルをすべて読み出す。カーソルを使う。

```
cur = c.cursor()
cur.execute(u"select * from iris")
for t in cur:
    print (t)
```

```
>>> for t in cur:
...     print (t)
...
(0, 5.1, 3.5, 1.4, 0.2, 'setosa')
(1, 4.9, 3.0, 1.4, 0.2, 'setosa')
(2, 4.7, 3.2, 1.3, 0.2, 'setosa')
(2, 4.6, 2.1, 1.5, 0.2, 'setosa')
```

- 「`select * from iris`」は、SQL 問い合わせ
- 「`cur`」は、カーソルである。問い合わせ結果を得るのに使う。

⑥ テーブルから、条件に合致するレコードを得る。

```
cur = c.cursor()  
cur.execute(u"select * from iris where id = 2")  
for t in cur:  
    print (t)
```

```
>>> cur = c.cursor()  
>>> cur.execute(u"select * from iris where id = 2")  
<sqlite3.Cursor object at 0x000001C278071340>  
>>> for t in cur:  
...     print (t)  
...  
(2, 4.7, 3.2, 1.3, 0.2, 'setosa')  
>>> _
```

```
cur = c.cursor()  
cur.execute(u"select * from iris where sepal_length > 7")  
for t in cur:  
    print (t)
```

```
>>> cur = c.cursor()  
>>> cur.execute(u"select * from iris where sepal_length > 7")  
<sqlite3.Cursor object at 0x000001C2780712D0>  
>>> for t in cur:  
...     print (t)  
...  
(102, 7.1, 3.0, 5.9, 2.1, 'virginica')  
(105, 7.6, 3.0, 6.6, 2.1, 'virginica')  
(107, 7.3, 2.9, 6.3, 1.8, 'virginica')  
(109, 7.2, 3.6, 6.1, 2.5, 'virginica')  
(117, 7.7, 3.8, 6.7, 2.2, 'virginica')  
(118, 7.7, 2.6, 6.9, 2.3, 'virginica')  
(122, 7.7, 2.8, 6.7, 2.0, 'virginica')  
(125, 7.2, 3.2, 6.0, 1.8, 'virginica')  
(129, 7.2, 3.0, 5.8, 1.6, 'virginica')  
(130, 7.4, 2.8, 6.1, 1.9, 'virginica')  
(131, 7.9, 3.8, 6.4, 2.0, 'virginica')  
(135, 7.7, 3.0, 6.1, 2.3, 'virginica')  
>>> _
```

⑦ 終了処理として、データベース接続を切断する。

```
c.close()
```

```
>>> c.close()  
>>>
```