

Python 的逆襲

- Python 與  SQLite



段維瀚 老師





SQLite



- 嵌入式資料庫

- 遵守ACID，實現了大多數SQL標準
- 它使用動態的、弱類型的SQL語法
- 在用戶端儲存資料的常見選擇
 - 它可能是最廣泛部署的資料庫引擎，因為它正在被一些流行的瀏覽器、作業系統、嵌入式系統所使用
- 廣受許多程式語言的支持
 - Python 3 已內建





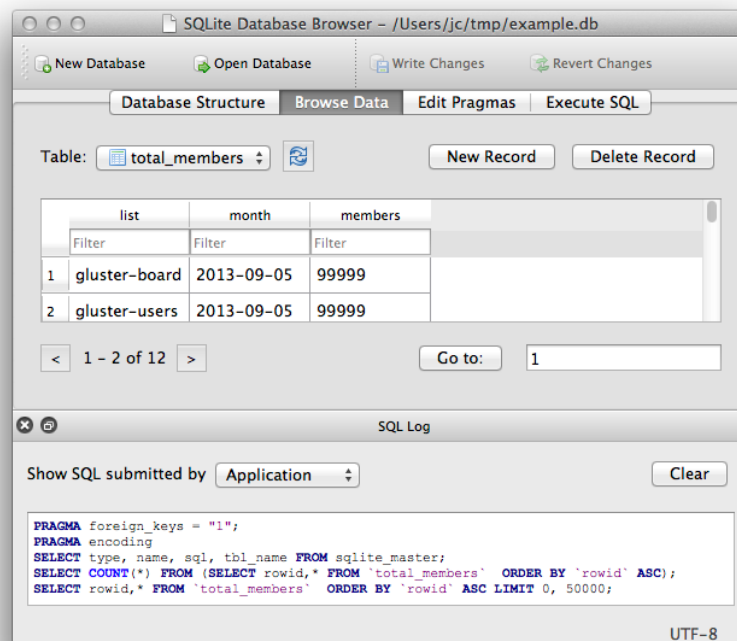
ACID

- ACID，是指資料庫管理系統（DBMS）在寫入或更新資料的過程中，為保證事務（transaction）是正確可靠的，所必須具備的四個特性：
 - 原子性（Atomicity，或稱不可分割性）
 - 一致性（Consistency）
 - 隔離性（Isolation，又稱獨立性）
 - 持久性（Durability）。



DB Browser for SQLite

- 請下載 SQLite 視窗管理介面
 - <https://sqlitebrowser.org/dl/>





connection

方法	說明
<code>cursor()</code>	建立一個 <code>cursor</code> 物件， <code>cursor</code> 物件的 <code>execute()</code> 方法可以完成資料表建立、CRUD 等動作
<code>execute(SQL命令)</code>	執行SQL命令 (CRUD)
<code>commit()</code>	執行資料庫更新
<code>close()</code>	關閉資料庫連線





cursor

方法	說明
<code>execute(SQL命令)</code> <code>execute(SQL命令, (args...))</code>	執行SQL命令 (CRUD)
<code>executemany(SQL命令, data[])</code>	執行多筆SQL命令 (CRUD)
<code>fetchall()</code>	以list的方式回傳所有資料或者是空list(無資料)
<code>fetchone()</code>	一次取出一筆，若沒有資料便會回傳None
<code>rowcount</code>	提供上次操作所影響的筆數，此屬性僅限用於UPDATE和DELETE命令
<code>lastrowid</code>	提供最後一個插入值的ID





建立 db

```
import sqlite3
```

```
conn = sqlite3.connect('demo.db')
```

```
print(conn)
```

```
conn.close()
```





建立 *table*

```
import sqlite3

sql = 'create table if not exists lotto (' \
      'id INTEGER PRIMARY KEY AUTOINCREMENT, ' \
      'n1 INTEGER, ' \
      'n2 INTEGER, ' \
      'n3 INTEGER, ' \
      'n4 INTEGER, ' \
      'n5 INTEGER, ' \
      'n6 INTEGER, ' \
      'ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP)'

conn = sqlite3.connect('demo.db')
cursor = conn.cursor() # 建立 cursor
cursor.execute(sql) # 建立 lotto 資料表

conn.commit() # 執行資料庫更新
conn.close()
print('建立成功')
```





新增紀錄- 單筆

```
import sqlite3
import random
```

```
conn = sqlite3.connect('demo.db')
cursor = conn.cursor() # 建立 cursor
```

```
cursor.execute('INSERT INTO lotto(n1, n2, n3, n4, n5, n6) VALUES '
               '(%d,%d,%d,%d,%d,%d)' % (1, 2, 3, 4, 5, 6))
```

```
print(cursor.lastrowid) # 取得最新一筆 id
conn.commit() # 執行資料庫更新
conn.close()
print('新增成功')
```





新增紀錄- 多筆

```
import sqlite3
import random

conn = sqlite3.connect('demo.db')
cursor = conn.cursor() # 建立 cursor

lottos = []

for i in range(100):
    # 取出 1~46 不重複的數字六個
    nums = set()
    while len(nums) < 6:
        nums.add(random.randint(1, 46))

    lottos.append(tuple(nums)) # 要轉元組

print(lottos)

cursor.executemany('INSERT INTO lotto(n1, n2, n3, n4, n5, n6) VALUES (?,?,?,?,?,?)', lottos)

conn.commit() # 執行資料庫更新
conn.close()
print('新增成功')
```





查詢 *table I*

```
import sqlite3
import random

conn = sqlite3.connect('demo.db')
cursor = conn.cursor() # 建立 cursor

# 查詢 Table META-INFO
cursor.execute('PRAGMA TABLE_INFO({})'.format('lotto'))
names = [t[1] for t in cursor.fetchall()]
for name in names:
    print(name, end='\t')

print("\n-----")

# 查詢資料列 SQL
sql = 'SELECT id, n1, n2, n3, n4, n5, n6, ts FROM lotto'

cursor.execute(sql)
rows = cursor.fetchall()
for row in rows:
    print('{0}\t{1}\t{2}\t{3}\t{4}\t{5}\t{6}\t{7}'.format(row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7]))
```





查詢 *table II*

查詢資料列 SQL

```
sql = 'SELECT id, n1, n2, n3, n4, n5, n6, ts FROM lotto'
```

```
cursor.execute(sql)
```

印出每列全部欄位的資料

```
lottos = [lo for lo in cursor.fetchall()]
```

```
print(lottos)
```

```
cursor.execute(sql)
```

印出每列第 2 欄位的資料 (欄位索引 0 開始)

```
lottos = [lo[1] for lo in cursor.fetchall()]
```

```
print(lottos)
```

