

SQLite3

2019.04



1. SQLite 란?

■ 경량 DBMS

- 별도의 서버가 필요 없음
- 모바일 기기에서 많이 활용되고 있음
- 파이썬3에 기본 내장되어 있음
- 파일 또는 메모리에 DB 생성
- 참고자료: SQLite로 가볍게 배우는 데이터베이스 (WikiDocs)

■ 데이터 타입

- 동적 데이터 타입
- Null, Integer, Real, Text, Blob 유형이 있음 (Boolean, Date, Time 없음)
- 다른 유형 데이터를 삽입해도 컬럼에 맞게 알아서 들어감.
- 다른 DB에서 사용하는 데이터유형 이름 그대로 사용해도 무방

■ DBMS 관리 툴

- SQLite Expert (<http://www.sqliteexpert.com/download.html>)
- Personal version은 freeware 이므로 사용할 수 있음.

2. 파이썬에서 사용하는 방법

■ 데이터베이스 접속

```
import sqlite3
conn = sqlite3.connect(':memory:')      # 메모리 DB 접속(일회성)
conn = sqlite3.connect('./test.db')    # 파일 DB 접속(일회성)
...
데이터 쿼리 수행
...
conn.commit()                          # 변경사항 저장
conn.close()
```

■ with 문 이용

```
import sqlite3
conn = sqlite3.connect('./test.db')

with conn:
    cur = conn.cursor()
    cur.execute("SELECT * FROM test_table")
    rows = cur.fetchall()
    for row in rows:
        print(row)
```

3. Data Definition Language(DDL)

■ 테이블 생성

```
cur = conn.cursor()
cur.execute('CREATE TABLE IF NOT EXISTS Eagles
            (back_no INT NOT NULL,
             name TEXT,
             position TEXT,
             PRIMARY KEY(back_no));')
```

■ 테이블 변경

```
cur.execute('ALTER TABLE Eagles ADD COLUMN birth INTEGER')
```

■ 테이블 삭제

```
cur.execute('DROP TABLE Eagles')
```

4. 데이터 조작 언어(Data Manipulation Language, DML)

■ 데이터 삽입

```
# 기본 스트링 쿼리
cur = conn.cursor()
cur.execute("INSERT INTO Eagles VALUES(1, '하주석', '내야수');")
cur.execute("INSERT INTO Eagles VALUES
            (57, '정우람', '투수'), (8, '정근우', '내야수');")
```

```
# 파라미터: 튜플 사용
back_no = 50
name = '이성열'
position = '외야수'
cur = conn.cursor()
sql = 'INSERT INTO Eagles VALUES (?, ?, ?);'
cur.execute(sql, (back_no, name, position))
```

```
# 튜플 리스트 사용
players = ((22, '이태양', '투수'), (13, '최재훈', '포수'))
cur = conn.cursor()
sql = 'INSERT INTO Eagles VALUES (?, ?, ?);'
cur.executemany(sql, players)
```

4. 데이터 조작 언어(Data Manipulation Language , DML)

■ 데이터 조회

순회 조회

```
cur = conn.cursor()
cur.execute('SELECT * FROM Eagles')
for row in cur:
    print(row)
```

단건 조회

```
cur = conn.cursor()
cur.execute('SELECT * FROM Eagles')
row = cur.fetchone()
```

다건 조회

```
rows = cur.fetchmany(2)
```

모두 조회

```
rows = cur.fetchall()
for row in rows:
    print(row)
```

4. 데이터 조작 언어(Data Manipulation Language , DML)

■ 데이터 조회

필요한 column만 조회

```
cur = conn.cursor()
```

```
cur.execute("SELECT name FROM Eagles WHERE back_no > 10")
```

```
rows = cur.fetchall();
```

```
for row in rows:
```

```
    print(row)
```

원하는 순서 및 갯수

```
cur.execute('SELECT * FROM Eagles ORDER BY name')
```

```
cur.execute('SELECT * FROM Eagles ORDER BY name DESC')
```

```
cur.execute('SELECT * FROM Eagles ORDER BY name DESC LIMIT 1')
```

```
row = cur.fetchone()
```

```
print(row[1])          # '하주석'
```

집계 함수

```
cur.execute('SELECT count(*) FROM Eagles')
```

```
count = cur.fetchone()
```

```
max(column), min(column), sum(column), avg(column)
```

4. 데이터 조작 언어(Data Manipulation Language , DML)

■ 데이터 검색

```
# 기본 스트링 쿼리
cur = conn.cursor()
cur.execute("SELECT * FROM Eagles WHERE position='내야수';")
rows = cur.fetchall();
for row in rows:
    print(row)

# Placeholder
cur = con.cursor()
back_no = 50
cur.execute('SELECT * FROM Eagles WHERE back_no=?;', (back_no,))
player = cur.fetchone()
print(player[0])                                # 50

# Grouping
sql = 'SELECT position, count(*) FROM Eagles GROUP BY position'
```


4. 데이터 조작 언어(Data Manipulation Language , DML)

■ 데이터 변경

```
position = '외야수'
back_no = 8
cur.execute('UPDATE Eagles SET position=? WHERE back_no=?;',
            (position, back_no))
cur.execute('SELECT * FROM Eagles WHERE back_no=?', (back_no,))
cur.fetchone()

data = ((1995,1), (1986,57))
sql = 'UPDATE Eagles SET position=? WHERE back_no=?'
cur.executedmany(sql, data)
```

■ 데이터 삭제

```
cur = con.cursor()
cur.execute('DELETE FROM Eagles WHERE back_no=1);')
```