

14주차 실습

python에서 데이터베이스 (SQLite3)의 이용

2019. 12. 03.

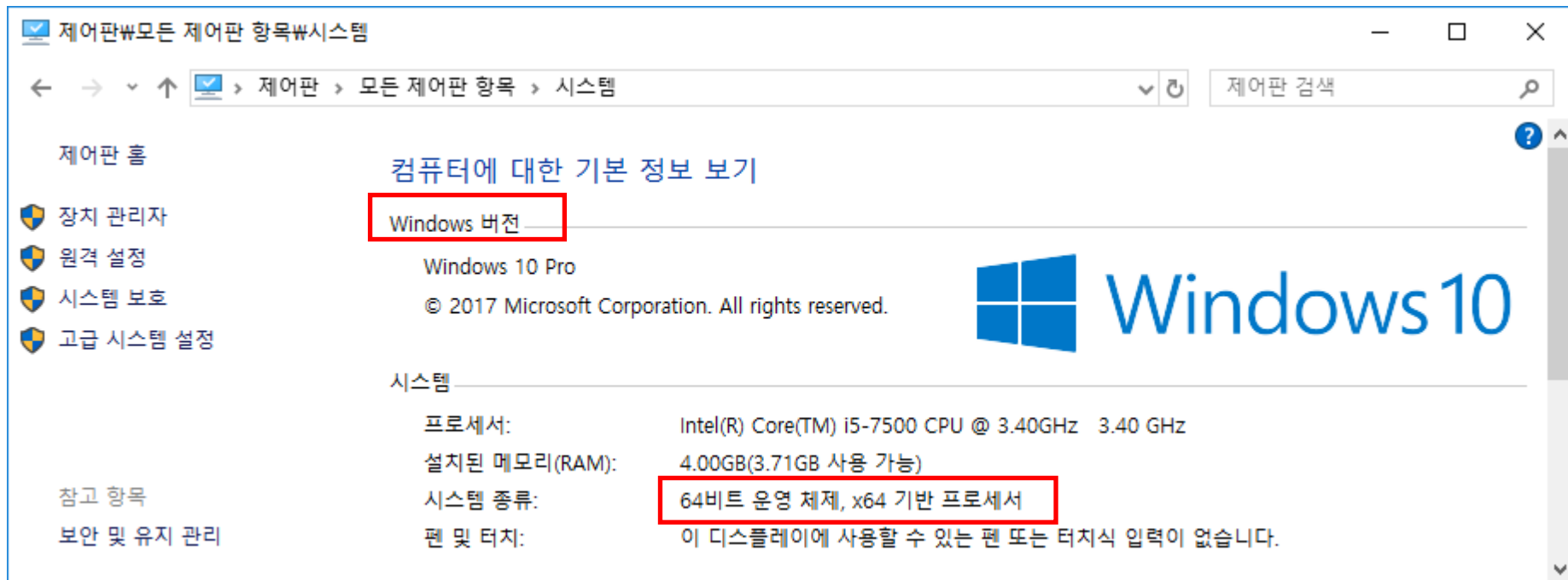
오늘의 학습내용 및 실습

1. SQLite3 설치방법
2. SQLite3의 간단 사용법
3. 파이썬에서 SQLite3를 이용하는 프로그래밍 방법
 1. 데이터베이스에 연결
 2. 커서를 오픈
 3. 테이블을 생성
 4. 예제 데이터를 입력
 5. 커서를 실행하는 작업 (입력/수정/삭제/조회) 수행
 6. 데이터를 확정(commit)
 7. 데이터베이스를 닫기
4. 소스코드
5. 오늘의 실습

1. SQLite3의 설치

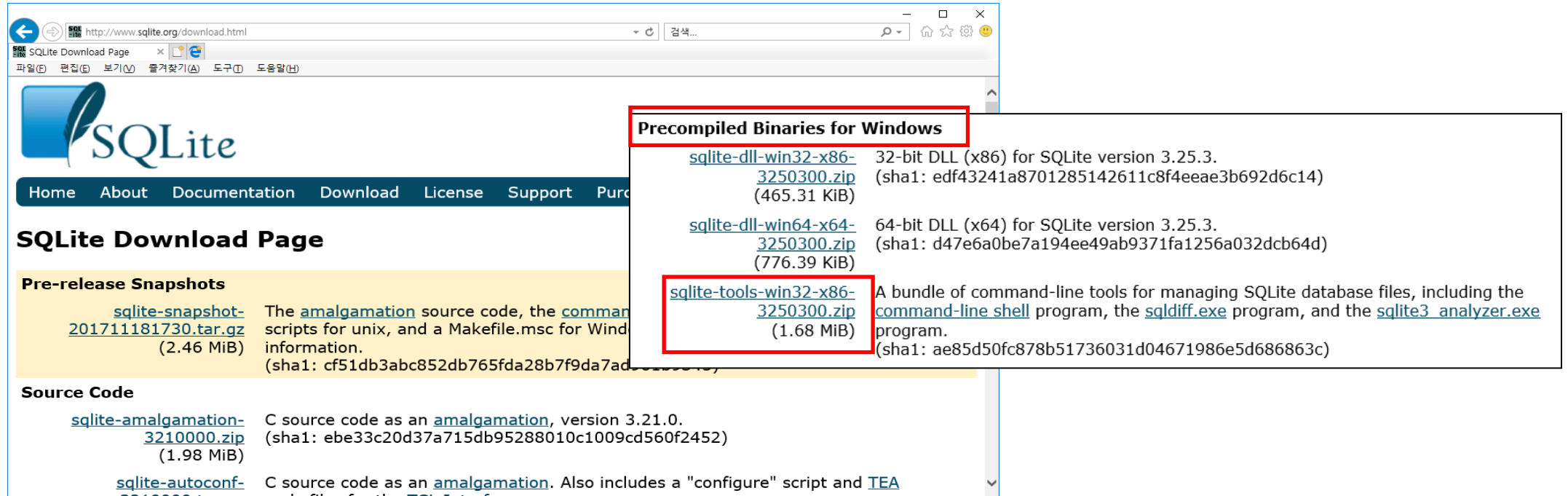
내 컴퓨터의 설치환경 점검

- 설치환경 점검
 - 시작 > 제어판 > 시스템 에서 시스템 종류를 점검



SQLite3의 다운로드 사이트에서 다운로드

- 다운로드
 - 사이트 : <http://www.sqlite.org/download.html>
 - Precompiled Binaries for Windows 검색



The screenshot shows the SQLite Download Page in a web browser. The page has a navigation bar with links: Home, About, Documentation, Download, License, Support, and Purchase. The main heading is "SQLite Download Page". Below this, there are sections for "Pre-release Snapshots", "Source Code", and "Precompiled Binaries for Windows". The "Precompiled Binaries for Windows" section is highlighted with a red box, and a search overlay is shown, listing three items: "sqlite-dll-win32-x86-3250300.zip" (465.31 KiB), "sqlite-dll-win64-x64-3250300.zip" (776.39 KiB), and "sqlite-tools-win32-x86-3250300.zip" (1.68 MiB). The search overlay also includes a description for the tools bundle: "A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff.exe program, and the sqlite3_analyzer.exe program." The search overlay is also highlighted with a red box.

SQLite Download Page

Home About Documentation Download License Support Purchase

SQLite Download Page

Pre-release Snapshots

[sqlite-snapshot-201711181730.tar.gz](#) (2.46 MiB) The [amalgamation](#) source code, the [command-line shell](#) scripts for unix, and a [Makefile.msc](#) for Windows information. (sha1: cf51db3abc852db765fda28b7f9da7ad...)

Source Code

[sqlite-amalgamation-3210000.zip](#) (1.98 MiB) C source code as an [amalgamation](#), version 3.21.0. (sha1: ebe33c20d37a715db95288010c1009cd560f2452)

[sqlite-autoconf-3210000.tar.gz](#) C source code as an [amalgamation](#). Also includes a "configure" script and [TEA](#) makefiles for the [TCL](#) interface.

Precompiled Binaries for Windows

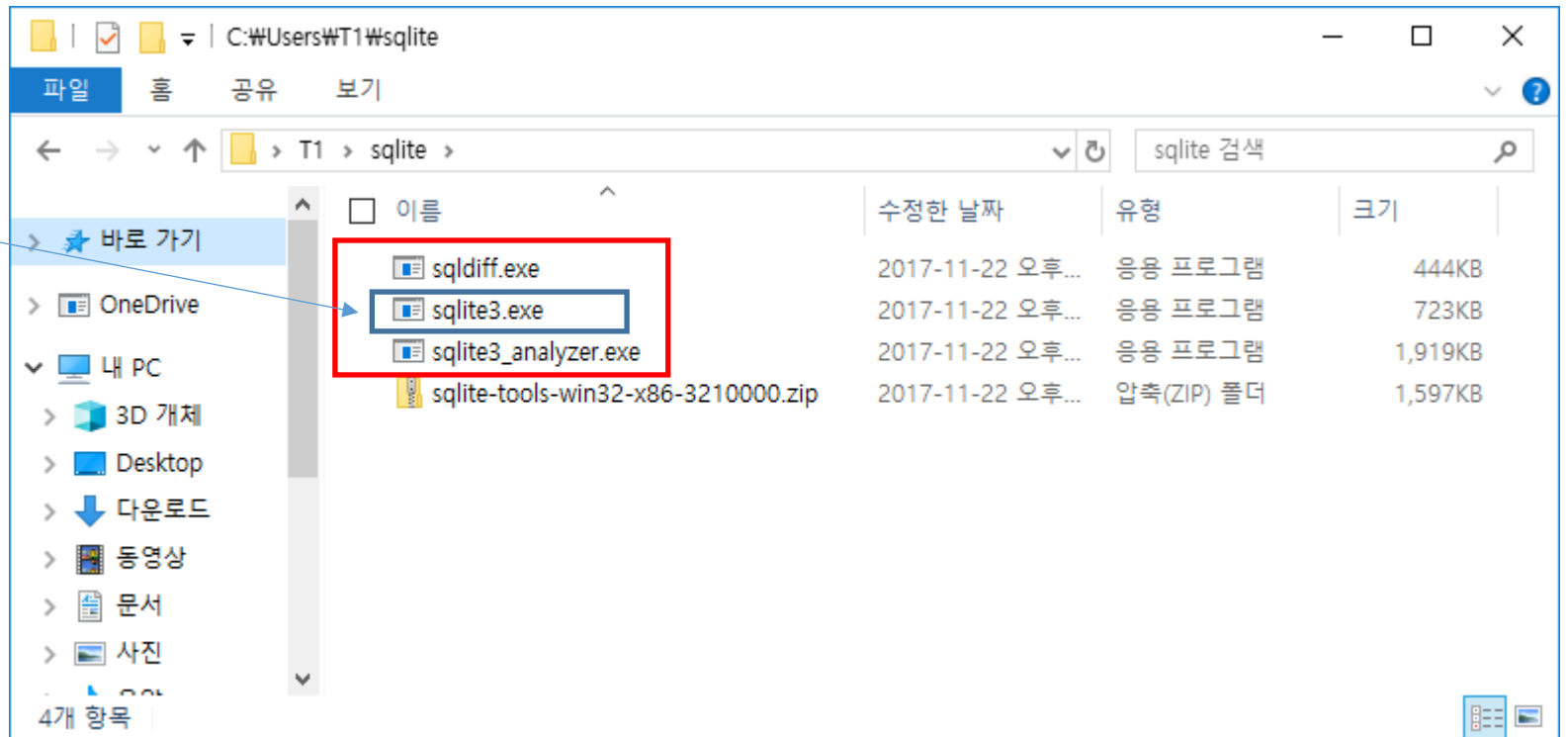
[sqlite-dll-win32-x86-3250300.zip](#) (465.31 KiB) 32-bit DLL (x86) for SQLite version 3.25.3. (sha1: edf43241a8701285142611c8f4eeae3b692d6c14)

[sqlite-dll-win64-x64-3250300.zip](#) (776.39 KiB) 64-bit DLL (x64) for SQLite version 3.25.3. (sha1: d47e6a0be7a194ee49ab9371fa1256a032dcb64d)

[sqlite-tools-win32-x86-3250300.zip](#) (1.68 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff.exe](#) program, and the [sqlite3_analyzer.exe](#) program. (sha1: ae85d50fc878b51736031d04671986e5d686863c)

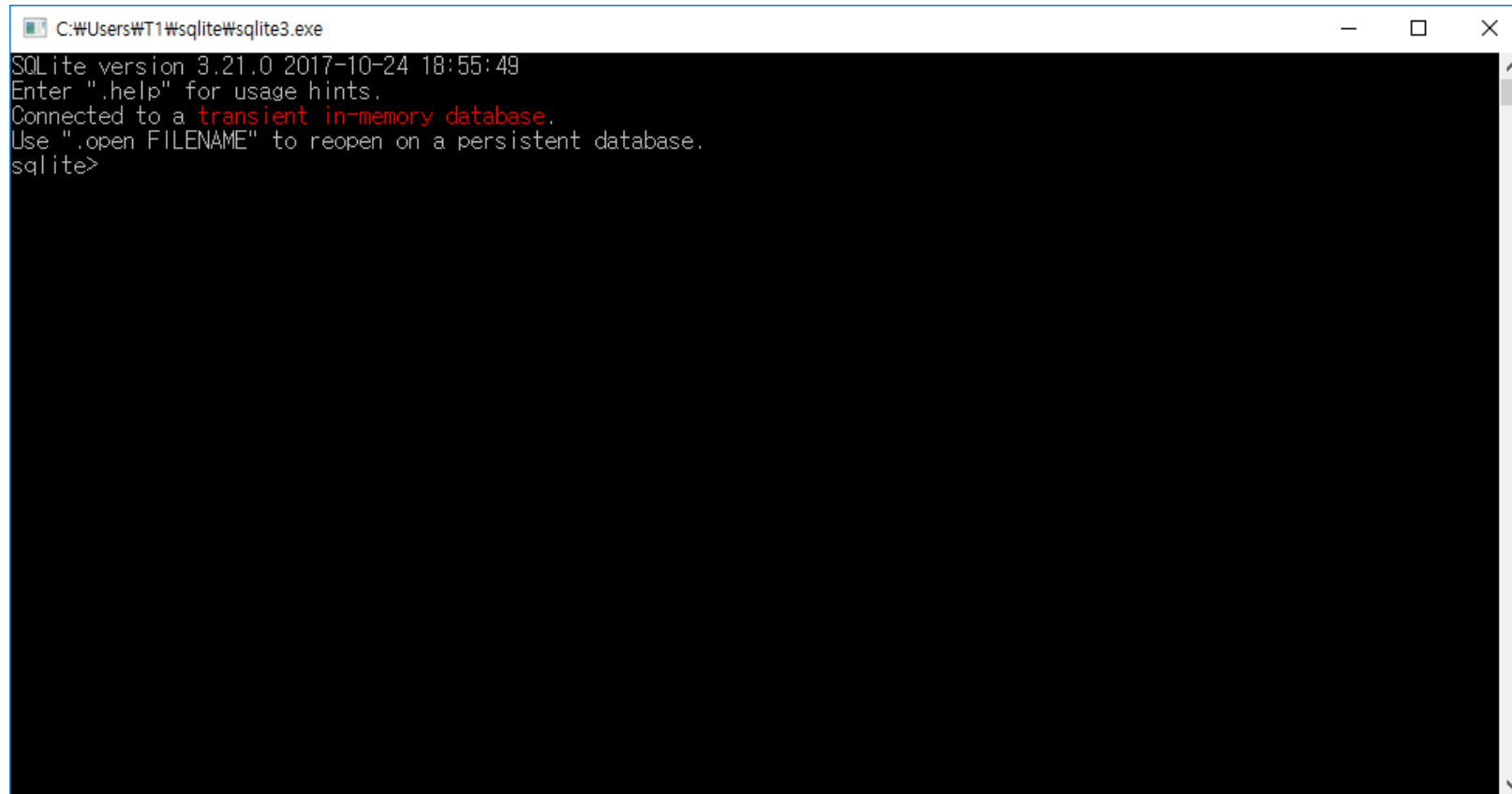
다운로드한 압축파일을 압축해제

- 다운로드한 파일을 하드디스크에 저장
 - 예) C:\T1\sqlite\
- 압축을 푼다.
 - 3개의 파일



sqlite3.exe를 실행하여 설치 여부 확인

- sqlite3.exe을 실행



```
C:\Users\WT1\sqlite\sqlite3.exe
SQLite version 3.21.0 2017-10-24 18:55:49
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

2. SQLite3의 간단 사용법

SQLite3 관련 참고자료

- SQLite3의 명령어라인 shell : <http://sqlite.org/cli.html>
- SQLite에서 사용할 수 있는 SQL(구조적질의어) <http://sqlite.org/lang.html>
- SQLite에서 사용할 수 있는 데이터 타입 : <http://sqlite.org/datatype3.html>
- SQLite에서 사용할 수 있는 함수
 - 일반함수 : http://sqlite.org/lang_corefunc.html
 - 그룹함수 : http://sqlite.org/lang_aggfunc.html
 - 날짜와 시간 함수 : http://sqlite.org/lang_datefunc.html
- SQLite 결과와 오류 코드 : <http://sqlite.org/rescode.html>
- 인덱스 : http://sqlite.org/keyword_index.html

SQLite3의 주요 dot 명령어

- .help
 - 사용할 수 있는 command들과 내용을 보여줌

```
C:\Users\WT1\sqlite\sqlite3.exe
SQLite version 3.21.0 2017-10-24 18:55:49
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .help
.auth ON|OFF          Show authorizer callbacks
.backup ?DB? FILE      Backup DB (default "main") to FILE
.bail on|off           Stop after hitting an error. Default OFF
.binary on|off         Turn binary output on or off. Default OFF
.cd DIRECTORY          Change the working directory to DIRECTORY
.changes on|off        Show number of rows changed by SQL
.check GLOB            Fail if output since .testcase does not match
.clone NEWDB           Clone data into NEWDB from the existing database
.databases             List names and files of attached databases
.dbinfo ?DB?          Show status information about the database
.dump ?TABLE? ...     Dump the database in an SQL text format
                      If TABLE specified, only dump tables matching
                      LIKE pattern TABLE.
.echo on|off           Turn command echo on or off
.eqp on|off|full       Enable or disable automatic EXPLAIN QUERY PLAN
.exit                 Exit this program
.fullschema ?--indent? Show schema and the content of sqlite_stat tables
.headers on|off        Turn display of headers on or off
.help                 Show this message
.import FILE TABLE    Import data from FILE into TABLE
.imposter INDEX TABLE Create imposter table TABLE on index INDEX
.indexes ?TABLE?       Show names of all indexes
                      If TABLE specified, only show indexes for tables
                      matching LIKE pattern TABLE.
.limit ?LIMIT? ?VAL?  Display or change the value of an SQLITE_LIMIT
```

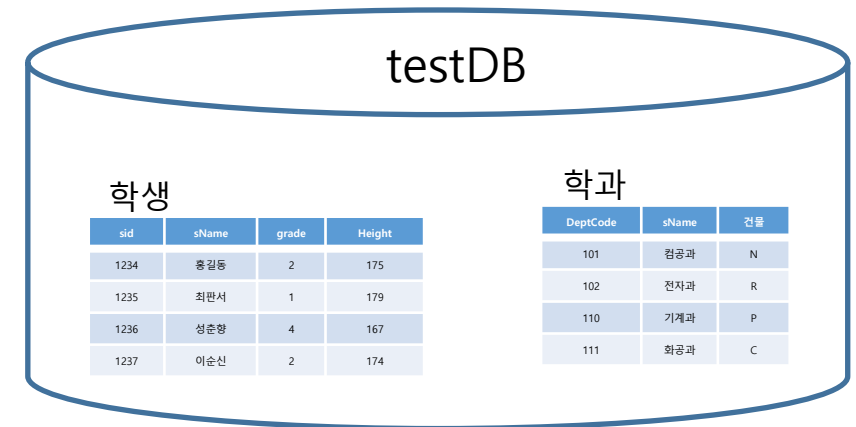
주요 SQLite3 dot 명령어

<http://sqlite.org/cli.html#dotcmd>

.open	: 기존데이터베이스가 있으면 열어주고, 없으면 생성해 줌
.table	: 현재 데이터베이스의 테이블들의 리스트를 보여줌
.schema 테이블명	: 테이블명의 컬럼과 데이터 형식 등의 정보를 보여줌
.header on	: SELECT문으로 출력할 때 header(제목줄)을 보여줌
.mode column	: SELECT문으로 출력할 때 컬럼 모드로 해줌
.exit	: SQLite를 종료함

command line shell에서 SQLite의 사용법

- 데이터베이스의 생성 및 사용
 - 1단계 사용할 데이터베이스의 생성
 - .open 사용할테이블이름
 - 2단계 테이블 생성
 - CREATE TABLE 테이블이름 (컬럼이름1 데이터형식, 컬럼이름2, 데이터형식2, ...);
 - 3단계 데이터 입력
 - INSERT INTO 테이블이름 VALUES (값1, 값2, ...);
 - 4단계 데이터 조회
 - SELECT * FROM 테이블 이름



예제 데이터베이스

Student

튜플

sid	sName	grade	Height
1234	홍길동	2	175
1235	최판서	1	179
1236	성춘향	4	167
1237	이순신	2	174
1238	강감찬	3	169
1239	최무선	1	181

테이블
(또는 릴레이션)

컬럼

command line shell에서 사용 예제

- 예제 데이터 1

.open testDB

```
CREATE TABLE Student ( sId char(4), sName char(15), grade int, Height int );
```

```
INSERT INTO Student Values ('1234', '홍길동', 2, 175) ;
```

```
INSERT INTO Student Values ('1235', '최판서', 1, 179) ;
```

```
INSERT INTO Student Values ('1236', '성춘향', 4, 167) ;
```

```
INSERT INTO Student Values ('1237', '이순신', 2, 174) ;
```

```
INSERT INTO Student Values ('1238', '강감찬', 3, 169) ;
```

```
INSERT INTO Student Values ('1239', '최무선', 1, 181) ;
```

```
SELECT * FROM Student;
```

.header on

.mode tab

```
SELECT * FROM Student;
```

.exit

.mode 옵션

-옵션종류 : quote, line, column, tab

2. 다시 SQLite3을 실행 후,
다음은 한 줄씩 입력

.table

.open testDB

.table

.mode column

```
SELECT * FROM Student;
```

.header on

```
SELECT * FROM Student;
```

command line shell에서 예제 실행 화면

```
C:\Users\WT1\sqlite>sqlite3.exe
.tables ?TABLE?      List names of tables
                      If TABLE specified, only list tables matching
                      LIKE pattern TABLE
.testcase NAME        Begin redirecting
.timeout MS           Try opening lock
.timer on|off         Turn SQL timer on|off
.trace FILE|off       Output each SQL statement to FILE|off
.vfsinfo ?AUX?        Information about the top-level VFS
.vfslist              List all available VFSes
.vfsname ?AUX?        Print the name of the VFS stack
.width NUM1 NUM2 ...  Set column widths for "column" mode
                      Negative values for "column" mode mean "reset"

sqlite> .tables
sqlite> .table
sqlite> .open testDB
sqlite> .table
sqlite> CREATE TABLE Student (
...>   sId char(4),
...>   sName char(15),
...>   grade int,
...>   Height int
...> );
sqlite> INSERT INTO Student Values ('1234','홍길동',2,175);
sqlite> INSERT INTO Student Values ('1235','최판서',1,179);
sqlite> INSERT INTO Student Values ('1236','성춘향',4,167);
sqlite> INSERT INTO Student Values ('1237','이순신',2,174);
sqlite> INSERT INTO Student Values ('1238','강감찬',3,169);
sqlite> INSERT INTO Student Values ('1239','최무선',1,181);
sqlite> SELECT * FROM Student;
```

```
C:\Users\WT1\sqlite>sqlite3.exe
SQLite version 3.21.0 2017-10-24 18:55:49
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .table
sqlite> .open testDB
sqlite> .table
Student
sqlite> .mode column
sqlite> SELECT * FROM Student;
sId      sName      grade      Height
-----
1234     홍길동      2           175
1235     최판서      1           179
1236     성춘향      4           167
1237     이순신      2           174
1238     강감찬      3           169
1239     최무선      1           181
sqlite>
```

다음은 command line shell에서 실습해 볼 것!

```
SELECT *  
FROM Student  
WHERE Height > 170;
```

```
SELECT sId, SName  
FROM Student  
WHERE Height > 170;
```

```
SELECT *  
FROM Student  
WHERE grade = 2;
```

```
SELECT *  
FROM Student  
WHERE sName like '최%';
```

```
SELECT *  
FROM Student  
WHERE sName like '_길_';
```

```
SELECT grade, count(*) AS '학생수'  
FROM Student  
GROUP BY grade;
```

다음은 command line shell에서 실습해 볼 것!

```
SELECT *  
FROM Student ;
```

```
INSERT INTO Student  
VALUES ('1240','김옥균',4, 176);
```

```
SELECT *  
FROM Student ;
```

```
DELETE FROM Student  
WHERE sName = '김옥균';
```

```
SELECT *  
FROM Student ;
```

```
UPDATE Student  
SET grade = 3  
WHERE sName = '홍길동';
```

```
SELECT *  
FROM Student ;
```


3. Python에서 SQLite의 사용

Python에서 SQLite3의 사용 절차

- SQLite3을 설치한다 (ok)
1. 데이터베이스에 연결
 2. 커서를 오픈
 3. 테이블을 생성
 4. 예제 데이터를 입력
 5. 커서를 실행하는 작업 (입력/수정/삭제/조회) 수행
 6. 데이터를 확정(commit)
 7. 데이터베이스를 닫기

1. 데이터베이스의 연결

```
import sqlite3  
conn = sqlite3.connect("testDB")
```

- sqlite3의 자원을 사용할 수 있게 import함
- 연결자(conn)을 이용하여 testDB라는 데이터베이스를 open
 - 기존에 testDB라는 데이터베이스가 없다면, create하여 open함

2. 커서 open

```
cur1 = conn.cursor()
```

- 문법 : 커서이름 = 연결자.cursor()
- 커서라는 자원을 이용하여 **데이터베이스와 python프로그램이 값을 주고 받음**

3. 테이블 만들기 / 삭제하기

```
cur1.execute("DROP TABLE Student;")  
sql = "CREATE TABLE Student (sId char(4),sName char(15), grade int, Height int );"  
cur1.execute(sql)
```

- 테이블 삭제하기
 - 기존에 Student 테이블이 있으면 지움
 - 문법 : DROP TABLE 테이블명 ;
- 테이블 생성하기
 - 이름이 cur1인 커서에 CREATE TABLE문을 전달하여, SQLite3이 이를 실행하게 함
 - 문법
 - CREATE TABLE 테이블이름 (컬럼이름1 데이터형식, 컬럼이름2, 데이터형식2, ...);
 - 대표적 데이터형식 : (문자 : CHAR(길이)), (정수 : INT)

4. 예제 데이터 입력 작업

sid	sName	grade	Height
1234	홍길동	2	175
1235	최판서	1	179
1236	성춘향	4	167
1237	이순신	2	174
1238	강감찬	3	169
1239	최무선	1	181

```
sql = "INSERT INTO Student Values ('1234', '홍길동', 2, 175);"  
cur1.execute(sql)  
sql = "INSERT INTO Student Values ('1235', '최판서', 1, 179);"  
cur1.execute(sql)  
sql = "INSERT INTO Student Values ('1236', '성춘향', 4, 167);"  
cur1.execute(sql)  
sql = "INSERT INTO Student Values ('1237', '이순신', 2, 174);"  
cur1.execute(sql)  
sql = "INSERT INTO Student Values ('1238', '강감찬', 3, 169);"  
cur1.execute(sql)  
sql = "INSERT INTO Student Values ('1239', '최무선', 1, 181);"  
cur1.execute(sql)
```

문법 :

INSERT INTO 테이블이름 VALUES (값1, 값2, ...);

주의사항

1. 입력할 데이터는 생성한 테이블의 컬럼 개수와 같아야 하고, 모든 컬럼의 데이터타입이 정의된 것과 같아야 함

```
CREATE TABLE Student (  
    sid char(4),  
    sName char(15),  
    grade int,  
    Height int );
```

2. 문자 데이터타입에 해당하는 컬럼은 값은 따옴표 (' ' 또는 " ")로 묶어줘야 함

4. 커서를 실행하는 작업 (데이터 조회)

```
while (True) :  
    data1 = input("\n      학번 : ")  
    if data1 == "" :  
        break;  
    data2 = input("      학생 이름 : ")  
    data3 = input("      학년 : ")  
    data4 = input("      키 : ")  
    sql = "INSERT INTO Student VALUES('\" + data1 + \"', '\" + data2 + \"', '\" + data3 + \"', '\" + data4 + \"');"   
    cur1.execute(sql)
```

sid	sName	grade	Height
1234	홍길동	2	175
1235	최판서	1	179
1236	성춘향	4	167
1237	이순신	2	174
1238	강감찬	3	169
1239	최무선	1	181

- python의 input()함수를 이용하여 입력받은 데이터를 SQLite3에 전달하기 위해 변수에 저장
- cur1이름의 커서를 통해 SQLite에게 다음을 실행하게 함
 - INSERT INTO Student VALUES('\" + data1 + \"', '\" + data2 + \"', '\" + data3 + \"', '\" + data4 + \"');

4. 커서를 실행하는 작업 (데이터 출력)

```
cur1.execute("SELECT * FROM Student")
while (True) :
    row = cur1.fetchone()
    data1 = row[0]
    data2 = row[1]
    data3 = row[2]
    data4 = row[3]
    print("   %5s %15s %5d %7d" % (data1, data2, data3, data4) )
```

sid	sName	grade	Height
1234	홍길동	2	175
1235	최판서	1	179
1236	성춘향	4	167
1237	이순신	2	174
1238	강감찬	3	169
1239	최무선	1	181

학번	이름	학년	키
1234	홍길동	2	175
1235	최판서	1	179
1236	성춘향	4	167
1237	이순신	2	174
1238	강감찬	3	169
1239	최무선	1	181

- cur1이름의 커서를 통해 SQLite에게 다음을 실행하게 함
 - SELECT * FROM Student;
- SQLite의 출력결과를 1라인씩 cur1.fetchone()을 통해 python의 row라는 이름의 list로 가져옴
- row의 내용을 변수에 저장하고 이를 출력

5. 입력한 데이터를 확정

```
conn.commit()
```

- SQLite3에 python을 통해 입력하거나 수정한 데이터의 변경을 영구적으로 저장하게 하는 것.
- 문법 : 연결자.commit()

6. 데이터베이스 닫기

```
conn.close()
```

- open하여 **사용하였던 데이터베이스를 닫는 작업**
 - 예제에서는 testDB
- 문법 : 연결자.close()

4. 소스 코드

파이썬에서 SQLite3 사용하기 소스 (1/6)

#파이썬 이용 조회/출력 프로그램 작성 예제 - 1

```
import sqlite3
```

변수선언

```
conn, cur1 = None, None
```

```
data1, data2, data3, data4 = "", "", 0, 0
```

```
sql=""
```

메인 코드 부분

```
conn = sqlite3.connect("testDB")
```

```
print("1. testDB 데이터베이스 연결 성공")
```

```
cur1 = conn.cursor()
```

```
print("2. 커서 오픈 성공")
```

만일 이미 존재하는 Student 테이블이 있다면 삭제하는 부분

```
try:
```

```
    cur1.execute("DROP TABLE Student;")
```

```
    print("3. 기존 Student 테이블이 있음. Student 테이블 삭제 성공")
```

```
except:
```

```
    print("3. 기존 Student 테이블이 없음")
```

파이썬에서 SQLite3 사용하기 소스 (2/6)

#파이썬 이용 조회/출력 프로그램 작성 예제 -2

Student 테이블 생성 부분

```
sql = "CREATE TABLE Student (sid char(4),sName char(15), grade int, Height int );"
```

```
cur1.execute(sql)
```

```
print("4. Student테이블 생성 성공")
```

Student 테이블의 예제 데이터 입력 부분

```
sql = "INSERT INTO Student Values ('1234','홍길동', 2, 175);"
```

```
cur1.execute(sql)
```

```
sql = "INSERT INTO Student Values ('1235','최판서', 1, 179);"
```

```
cur1.execute(sql)
```

```
sql = "INSERT INTO Student Values ('1236','성춘향', 4, 167);"
```

```
cur1.execute(sql)
```

```
sql = "INSERT INTO Student Values ('1237','이순신', 2, 174);"
```

```
cur1.execute(sql)
```

```
sql = "INSERT INTO Student Values ('1238','강감찬', 3, 169);"
```

```
cur1.execute(sql)
```

```
sql = "INSERT INTO Student Values ('1239','최무선', 1, 181);"
```

```
cur1.execute(sql)
```

```
print("5. 학생데이터 6명을 Student테이블에 입력 성공")
```

sid	sName	grade	Height
1234	홍길동	2	175
1235	최판서	1	179
1236	성춘향	4	167
1237	이순신	2	174
1238	강감찬	3	169
1239	최무선	1	181

파이썬에서 SQLite3 사용하기 소스 (3/6)

#파이썬 이용 조회/출력 프로그램 작성 예제 - 3

Student 테이블의 예제 데이터 출력 부분

```
print("\n -----")
print("Wt   학번      Wt   이름 Wt   학년   키")
print(" -----")

cur1.execute("SELECT * FROM Student;")
while (True) :
    row = cur1.fetchone()
    if row == None:
        break;
    data1 = row[0]
    data2 = row[1]
    data3 = row[2]
    data4 = row[3]
    print("   Wt%5s Wt%15s Wt%5d Wt%7d" % (data1, data2, data3, data4))
print(" -----Wn")
print("6. 6명의 학생데이터를 출력 성공")
```

파이썬에서 SQLite3 사용하기 소스 (4/6)

#파이썬 이용 조회/출력 프로그램 작성 예제 - 4

새로운 학생 입력 부분

```
print("\n *****")
print(" 새로운 학생을 입력하시오~")
print(" 학번 입력에 아무것도 입력하지 않으면 ==> 입력 끝")
print("\n *****")
while (True) :
    data1 = input("\n 학번 : ")
    if data1 == "" :
        break;
    data2 = input(" 학생 이름 : ")
    data3 = input(" 학년 : ")
    data4 = input(" 키 : ")
    sql = "INSERT INTO Student VALUES('" + data1 + "', '" + data2 + "', '" + data3 + "', '" + data4 + "');"
    cur1.execute(sql)
```

파이썬에서 SQLite3 사용하기 소스 (5/6)

#파이썬 이용 조회/출력 프로그램 작성 예제 - 5

추가로 변경된 학생 테이블 출력 부분

```
print("\n *****")
print("          추가로 변경된 학생테이블")
print(" *****")
print("\n -----")
print("\t 학번 \t\t \t 이름 \t\t 학년 \t\t 키")
print(" -----")
cur1.execute("SELECT * FROM Student;")
while (True) :
    row = cur1.fetchone()
    if row == None:
        break;
    data1 = row[0]
    data2 = row[1]
    data3 = row[2]
    data4 = row[3]
    print("\t\t %5s \t %15s \t %5d \t %7d" % (data1, data2, data3, data4))
print(" -----")
```


파이썬에서 SQLite3 사용하기 소스 (6/6)

#파이썬 이용 조회/출력 프로그램 작성 예제 - 6

변경된 내용을 Student테이블에 확정하고 끝내는 부분

```
conn.commit()
```

```
conn.close()
```

```
print("7. Student테이블의 변경내용을 확정하고 끝내기 성공")
```

오늘의 실습

2019. 12. 03.

오늘의 실습 문제 (1/2)

- 다음을 처리하는 파이썬 프로그램을 작성하라.
 - 제출 : 12월 8일(일) 자정까지
- 1. testDB라는 이름의 데이터베이스를 생성하라.
- 2. testDB안에 Dept라는 테이블을 생성하라.
 - Dept 는 학과테이블이다. 다음의 컬럼을 가지고 있다.
 - deptCode : char (3) ---- 학과코드
 - deptName : char (10) ---- 학과이름
 - loc : char (1) ---- 건물위치
- 3. 다음의 데이터를 입력하라.

deptCode	deptName	loc
101	컴공과	N
102	전자과	N
110	경영학과	P
111	화공과	N

오늘의 실습 문제 (2/2)

4. 새로운 학과를 python의 input함수를 이용하여 입력받아서 Dept테이블에 입력한다.

deptCode	deptName	loc
120	영어학과	S

5. 건물 이름 (예 : 'N'동)을 python의 input함수를 이용하여 입력 받아, 해당 건물(예: 'N'이 입력되었다면 'N'동)에 있는 모든 학과명을 출력하라.
6. 학과이름을 입력받아(예:영어학과), 해당 학과의 학과정보를 Dept테이블에서 삭제하라.