

SQLITE + Python

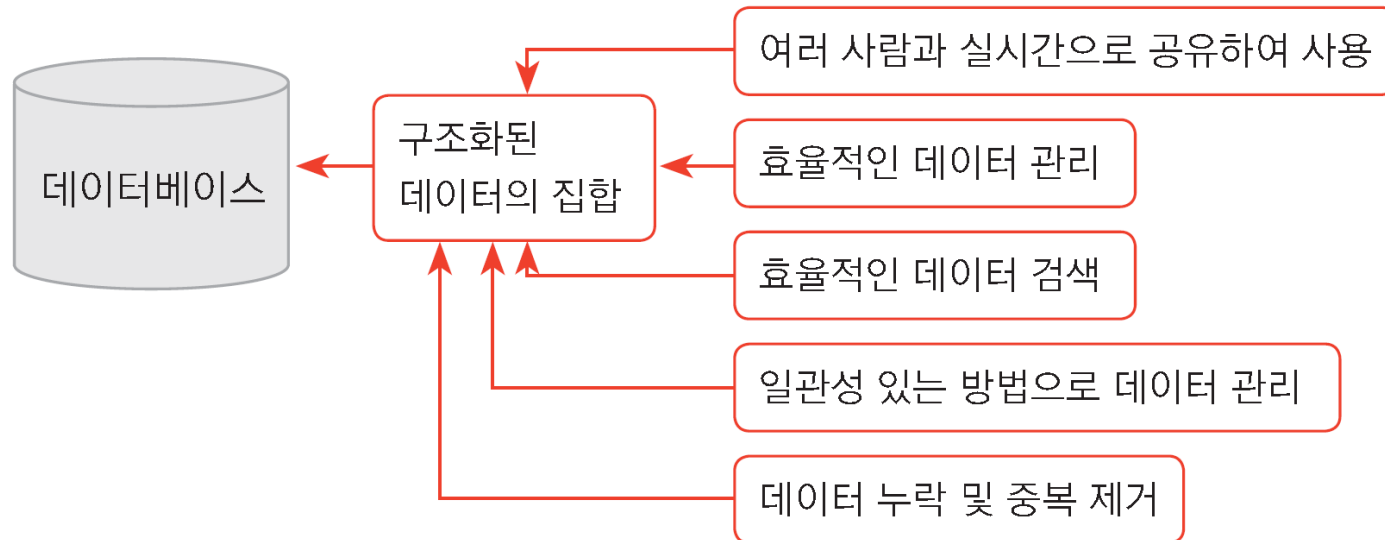
Update:2023-02

- 데이터베이스 개요
- SQLITE3
- SQL
- SQLITE + Python 연동

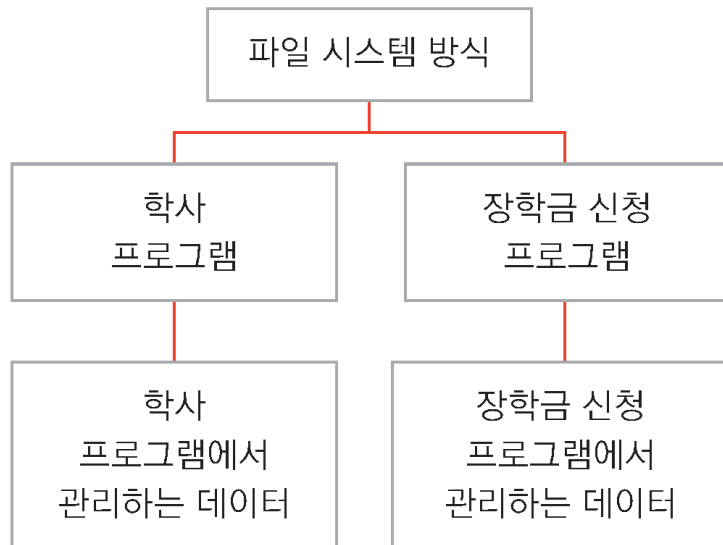
데이터베이스 개요

데이터베이스(DBMS)

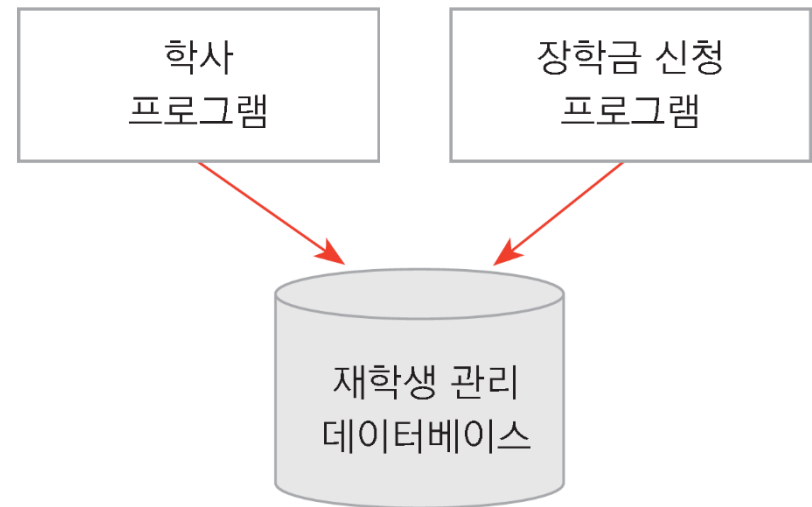
- 데이터베이스 : 효율적인 관리와 검색을 위해 구조화한 데이터 집합



파일 시스템 VS DBMS



- 데이터 중복
- 데이터가 특정 응용 프로그램에 종속
- 응용 프로그램을 변경하면 기존 데이터 사용 불가능



- 하나의 소프트웨어가 데이터를 관리하므로 데이터 중복을 피할 수 있음
- 여러 응용 프로그램에서 하나의 DBMS를 이용하여 데이터 사용 가능
- 응용 프로그램의 업데이트 또는 변경과 관계없이 데이터를 사용할 수 있음

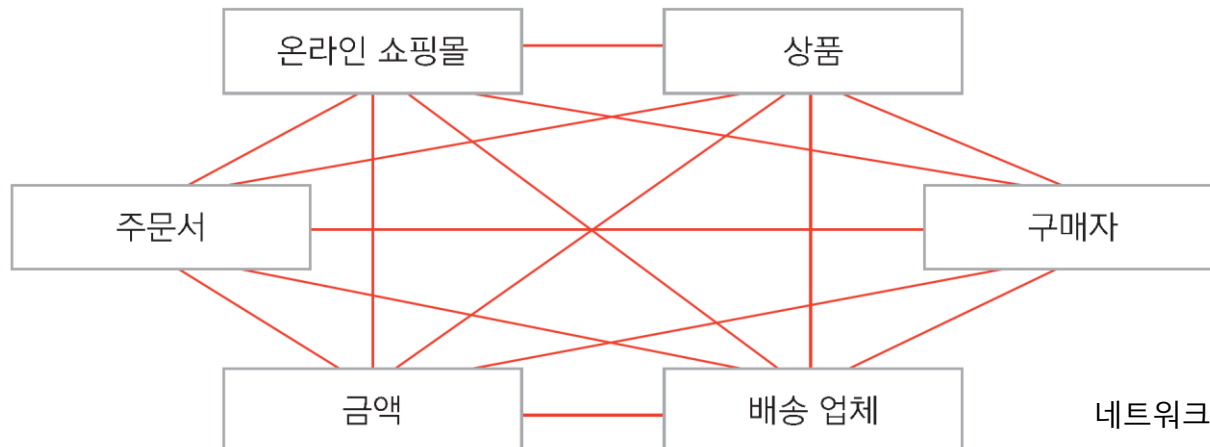
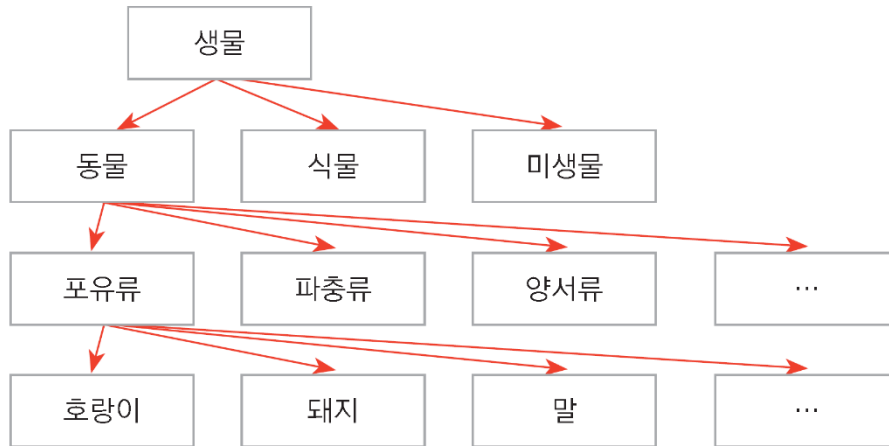
DBMS를 통한 데이터 관리



- DBMS : 데이터베이스관리시스템
- 데이터의 접근 • 관리 등의 업무를 DBMS가 전담

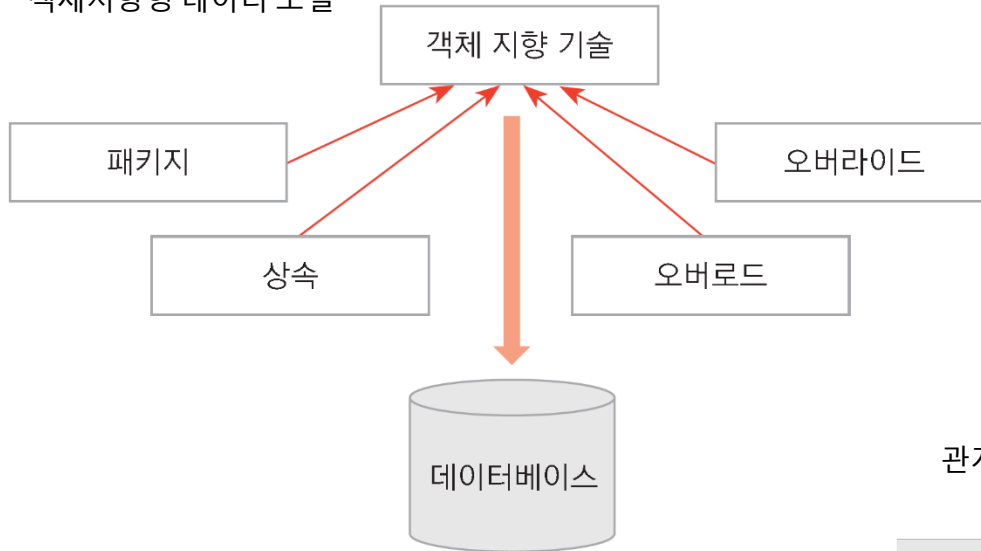
데이터 모델

- 데이터모델이란? 컴퓨터에 데이터를 저장하는 방식



데이터 모델

객체지향형 데이터 모델



관계형 데이터 모델

사원 정보	사원 번호	사원 이름	사원 직급	부서 코드
사원 번호 사원 이름 사원 직급 부서 코드	0001	홍길동	과장	10
	0002	성춘향	대리	20
	0003	박문수	사원	30
	0004	심청이	사원	10
부서 정보	부서 코드	부서 이름	위치	
부서 코드 부서 이름 위치	10	회계	서울	
	20	연구	인천	
	30	운영	분당	

관계형 데이터베이스

- 관계형 데이터베이스란?
 - 관계형 데이터 모델 개념 바탕
 - RDBMS(Relational Database Management System)
 - Oracle, MS-SQL, MySQL, MariaDB, PostgreSQL, DB2, etc...
 - SQL(Structured Query Language) : 데이터베이스 질의 언어

관계형 데이터베이스

- 관계형 데이터베이스의 구성 요소

학번	이름	생년월일	전화번호	집 주소	학년	학기	학과 코드	졸업 여부	...
16031055	홍길동	971210	010-1111-1111	서울시...	1	2	COM		
12071632	성춘향	940424	010-2222-2222	부산시...	4	2	BNS	졸업	
			...						
15022655	박문수	960605	010-3333-3333	광주시...	2	2	MTH		

테이블

열

행

관계형 데이터베이스

- 기본키(PK:Primary Key) - 특별한 의미를 지닌 열, 키
- 보조키(alternate key) 또는 대체키 : 후보키의 일종



관계형 데이터베이스

- 외래키(FK : Foreign Key)

: 특정 테이블에 포함되어 있으면서 다른 테이블의 기본키로 지정된 키

학생 정보 테이블

학번	이름	...	학과 코드	졸업 여부	...
16031055	홍길동	...	COM		
12071632	성춘향	...	BNS	졸업	
		...			
15022655	박문수	...	MTH		

학과 정보 테이블

학과 코드	학과 이름	개설 날짜	대학 구분	대표 전화번호	...
COM	컴퓨터공학과	19861021	공과대학	070-0000-0000	
BNS	경영학과	19730201	경상대학	070-1111-1111	
		...			
MTH	수학과	19730201	과학대학	070-2222-2222	

SQLITE3

SQLite 소개

SQLite란 오라클, MS-SQL, My-SQL과 달리 소규모 DB에 사용되는 관계 형 크기가 매우 작고, 작동하기가 쉽다는 장점이 있고 라이선스 제약에서 벗어날 수 있다. 모든 휴대폰과 대부분의 컴퓨터에 내장되어 있으며 사람들이 매일 사용하는 수많은 다른 응용 프로그램에 번들로 제공되어지고 있다.

특징

- 단일 파일에 데이터베이스의 모든 것을 포함한다.
- TCL, Perl, PHP, .Net, Java, Python, SmallTalk, Ruby등의 다양한 언어를 지원한다.
- Public Domain 라이선스를 가진다.
- 효율적인 메모리 사용 : 25k라인정도의 C코드로 이루어졌다.
- 표준 SQL 언어를 지원한다.



<https://www.sqlite.org>

설치 및 작업환경 설정

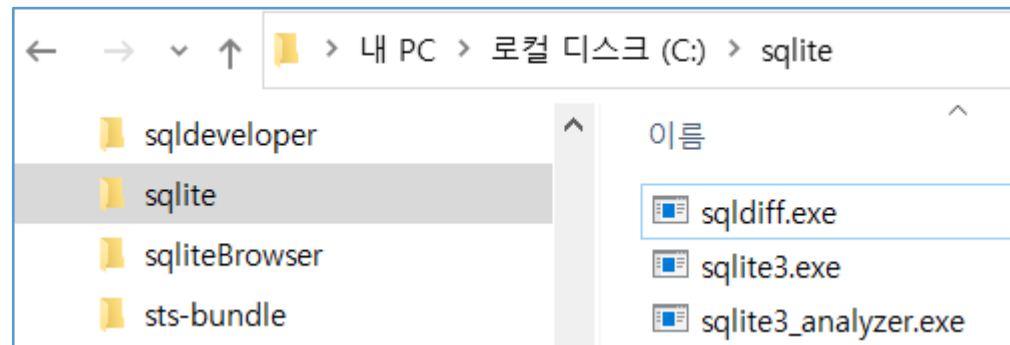
- CLI 개발환경

<https://www.sqlite.org/download.html>

Precompiled Binaries for Windows

sqlite-dll-win32-x86-3400100.zip (559.78 KiB)	32-bit DLL (x86) for SQLite version 3.40.1. (sha3-256: 419652569c65fd2a8cc8fef0cf2894cf64060e79226240030bda67bd21312099)
sqlite-dll-win64-x64-3400100.zip (896.43 KiB)	64-bit DLL (x64) for SQLite version 3.40.1. (sha3-256: 0087fa491604ddcb01356d20d271eaf209c97431836fb72b599fec7eaa14a640)

다운로드 후 별도 폴더(예:sqlite)에 압축 해제한다.



설치 및 작업환경 설정

- GUI 개발환경

<https://sqlitebrowser.org/dl>

- 윈도우 64비트 기준

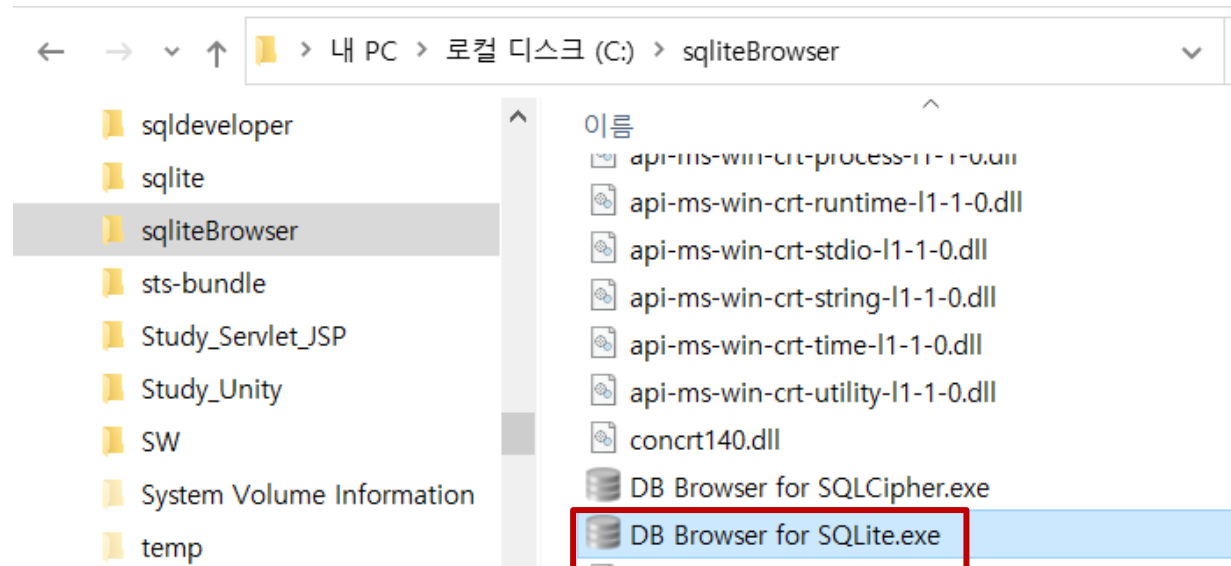
DB Browser for SQLite - .zip (no installer)
for 64-bit Windows

: 별도 폴더 생성후 압축 해제
DB Browser for SQLite.exe 실행

Windows

Our latest release (3.12.1) for Windows:

- DB Browser for SQLite - Standard installer for 32-bit Windows
- DB Browser for SQLite - .zip (no installer) for 32-bit Windows
- DB Browser for SQLite - Standard installer for 64-bit Windows
- DB Browser for SQLite - .zip (no installer) for 64-bit Windows



SQL

SQL 개요

- SQL이란?



- Structured Query Language => SQL

구조적 데이터 질의 언어. 에스큐엘, 혹은 시퀄이라고 읽는다.

관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어이다.

관계형 데이터 베이스 관리 시스템에서 자료의 검색과 관리, 데이터베이스 스키마 생성과 수정, 데이터 베이스 객체 접근 조정 관리를 위해 고안되었다.

많은 수의 데이터베이스 관련 프로그램들이 SQL을 표준으로 채택하고 있다.

- 로고는 SQL을 사용하는 회사별로 색상을 다르게 해 표현하지만 심볼 자체는 원통을 쌓은 모양이 공용으로 쓰인다.

Sqlite Shell Command

-- 터미널 창 실행 (윈도우+R, cmd)

Sqlite 설치 폴더로 이동

>> cd c:/sqlite

-- 데이터베이스 생성

sqlite3 데이터베이스명

예) C:\sqlite>sqlite3 book.db

-- 데이터베이스명

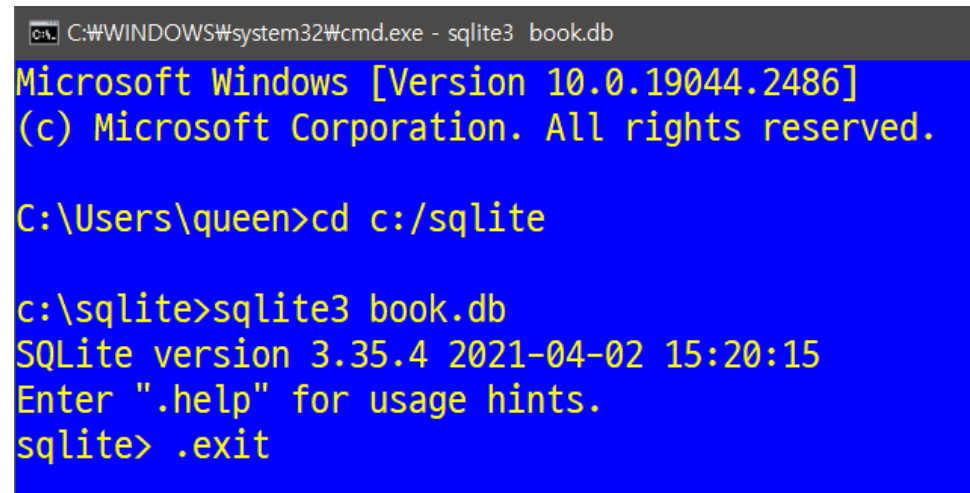
.db

.sqlite

.sqlite3

-- 데이터베이스와의 연결 종료

.exit



```
C:\WINDOWS\system32\cmd.exe - sqlite3 book.db
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\queen>cd c:/sqlite

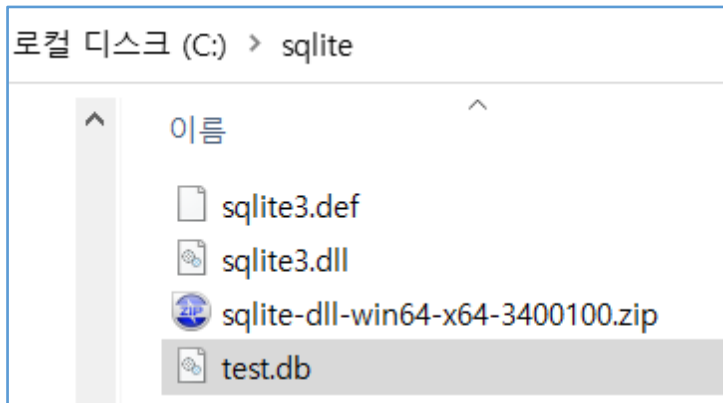
c:\sqlite>sqlite3 book.db
SQLite version 3.35.4 2021-04-02 15:20:15
Enter ".help" for usage hints.
sqlite> .exit
```

Sqlite Shell Command

-- 데이터베이스 연결

sqlite3 데이터베이스명.확장자

예) C:\sqlite>sqlite3 test.db



test.db 파일을
sqlite 폴더에 복사한다

```
c:\sqlite>sqlite3 test.db
SQLite version 3.35.4 2021-04-02 15:20:15
Enter ".help" for usage hints.
sqlite> _
```

Sqlite Shell Command

-- 테이블 목록 보기

`.tables`

-- 현재 연결된 목록보기

`.databases`

-- 테이블 구조 확인

`.schema` 테이블명

```
c:\sqlite>sqlite3 test.db
SQLite version 3.35.4 2021-04-02 15:20:15
Enter ".help" for usage hints.
sqlite> .tables
albums          employees       invoices        playlists
artists         genres         media_types     tracks
customers       invoice_items  playlist_track
sqlite> .databases
main: c:\sqlite\test.db r/w
```

Sqlite Shell Command

-- 컬럼포매팅 모드

.mode columns

-- 상단 필드 표시

.headers on

-- 옵션 확인

.show

```
sqlite> .mode columns
sqlite> .headers on
sqlite> .show
      echo: off
      eqp: off
    explain: auto
    headers: on
      mode: column
  nullvalue: ""
    output: stdout
colseparator: "|"
rowseparator: "\n"
      stats: off
      width:
  filename: test.db
```

Select

-- SELECT => SQL 출력문
-- 필터링 - 조건에 맞는 레코드 표시

SELECT 필드명 FROM 테이블명 (WHERE 조건절 LIMIT 숫자);

-- 특정 테이블의 전체 레코드 확인
SELECT * FROM 테이블명

-- 특정 테이블의 특정 필드 확인
-- SELECT 필드명1, 필드명2 FROM 테이블명;

-- 특정 테이블에서 레코드수 제한하여 출력하기
SELECT * FROM 테이블명 LIMIT 숫자;
SELECT 필드명1, 필드명2 FROM 테이블명 LIMIT 숫자;

Select

```
sqlite> select * from albums limit 5;
```

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
4	Let There Be Rock	1
5	Big Ones	3

```
sqlite> select firstname, lastname from customers limit 10;
```

FirstName	LastName
Luis	Goncalves
Leonie	Kohler
Francois	Tremblay
Bjørn	Hansen
František	Wichterlova
Helena	Holy
Astrid	Gruber
Daan	Peeters
Kara	Nielsen
Eduardo	Martins

Select

-- WHERE 조건절

-- = , <=, >=, <, >

-- AND, OR, NOT Clause

SELECT 필드명 FROM 테이블명 WHERE 조건절 ;

```
sqlite> select firstname, lastname, customerid from customers where customerid = 10;
FirstName  LastName  CustomerId
-----
Eduardo    Martins   10
```

```
sqlite> select firstname, lastname, customerid
...> from customers
...> where customerid = 10 or customerid = 30;
FirstName  LastName  CustomerId
-----
Eduardo    Martins   10
Edward     Francis   30
```

Select

```
sqlite> select * from playlists where playlistid <= 5;
PlaylistId  Name
-----
1           Music
2           Movies
3           TV Shows
4           Audiobooks
5           90's Music
```

```
sqlite> select * from playlists
...> where playlistid <= 5 and playlistid > 2;
PlaylistId  Name
-----
3           TV Shows
4           Audiobooks
5           90's Music
```

Select

```
-- 특정 컬럼을 기준으로 정렬하기
-- SELECT * FROM 테이블명
    WHERE 조건절
    ORDER BY 컬럼명1, 컬럼명2 ASC/DESC
    LIMIT 숫자;
-- ASC : 오름차순 / DESC : 내림차순
```

```
sqlite> select * from albums order by title limit 10;
```

AlbumId	Title	ArtistId
156	...And Justice For All	50
257	20th Century Masters - The Millennium Collection: The Best of Scorpions	179
296	A Copland Celebration, Vol. I	230
94	A Matter of Life and Death	90
95	A Real Dead One	90
96	A Real Live One	90
285	A Soprano Inspired	219
139	A TempestadeTempestade Ou O Livro Dos Dias	99
203	A-Sides	132
160	Ace Of Spades	106

Select

```
sqlite> select * from albums order by title desc limit 10;
```

AlbumId	Title	ArtistId
208	[1997] Black Light Syndrome	136
240	Zooropa	150
267	Worlds	202
334	Weill: The Seven Deadly Sins	264
8	Warner 25 Anos	6
239	War	150
175	Walking Into Clarksdale	115
287	Wagner: Favourite Overtures	221
182	Vs.	118
53	Vozes do MPB	21

Select

-- WHERE 필드명 LIKE %서식 == *

예) 'a%'

-- WHERE 필드명 LIKE _서식 _글자수

예) 'U__'

```
sqlite> select * from albums
...> where title like 'w%';
```

AlbumId	Title	ArtistId
8	Warner 25 Anos	6
175	Walking Into Clarksdale	115
239	War	150
267	Worlds	202
287	Wagner: Favourite Overtures	221
334	Weill: The Seven Deadly Sins	264

```
sqlite> select * from albums
...> where title like 'w__';
```

AlbumId	Title	ArtistId
239	War	150

Select

-- 테이블의 레코드 갯수 반환하기

SELECT COUNT(*) FROM 테이블 WHERE 절;

```
sqlite> select count(*) from albums;
count(*)
-----
347
```

```
sqlite> select count(*) from albums
...> where title like 'a%';
count(*)
-----
32
```

Select

-- SQLite Browser 활용하기

데이터베이스 연결하기

SQL 명령 실행하기



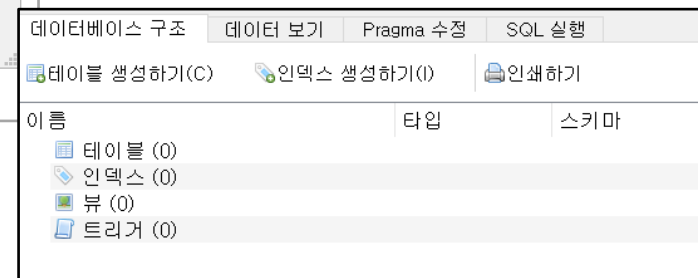
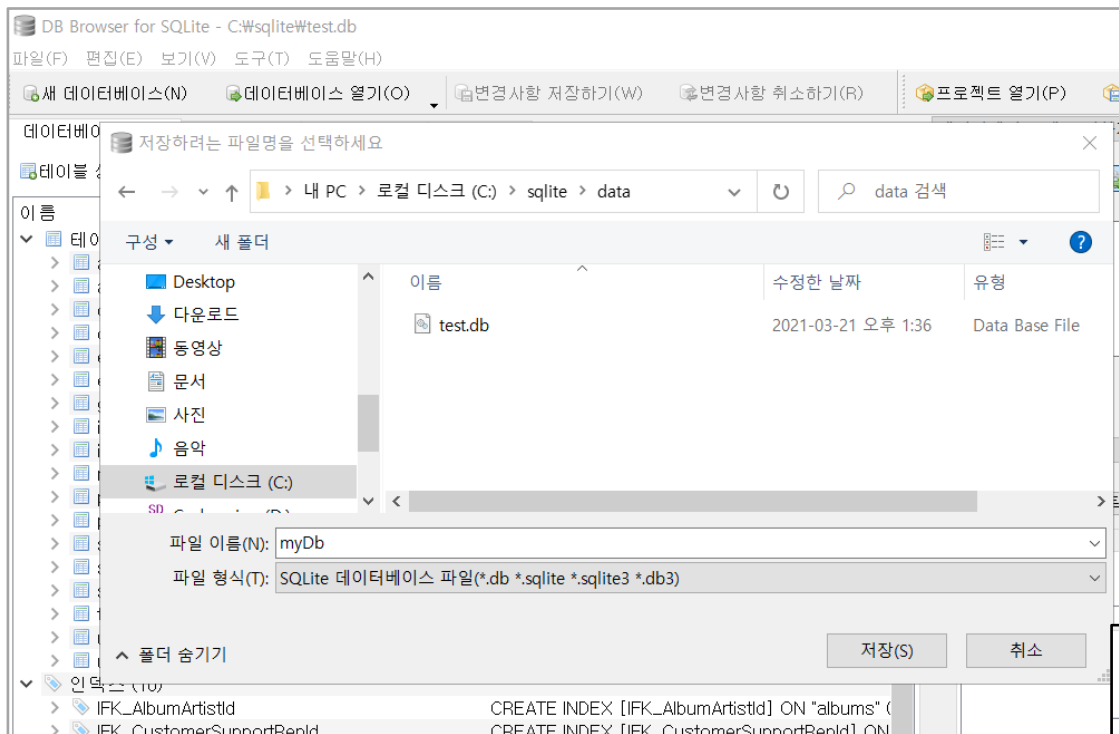
The screenshot shows the SQLite Browser application window. The title bar reads "DB Browser for SQLite - C:\sqlite\test.db". The menu bar includes "파일(E)", "편집(E)", "보기(V)", "도구(I)", and "도움말(H)". The toolbar contains icons for opening a new database, opening an existing database, saving, printing, and running SQL. The "데이터베이스 구조" (Database Structure) tab is active, showing a tree view with "데이터베이스" (Database) and "테이블" (Tables). The "SQL 실행" (SQL Execute) tab is also visible. The SQL editor shows a query: "SELECT * FROM albums;". The results are displayed in a table with columns "AlbumId", "Title", and "ArtistId".

	AlbumId	Title	ArtistId
1	1	For Those About To Roc...	1
2	2	Balls to the Wall	2
3	3	Restless and Wild	2
4	4	Let There Be Rock	1
5	5	Big Ones	3
6	6	Jagged Little Pill	4
7	7	Facelift	5

데이터베이스 생성

-- SQLite Browser 에서 데이터베이스 만들기

[새 데이터베이스] 클릭후 작업폴더 지정 => 파일이름 지정 후 [저장]



Create Table & Insert Row

-- 데이터 타입

NULL NULL 값

INTEGER 부호있는 정수. 1, 2, 3, 4, 6, or 8 바이트로 저장

REAL 부동 소수점 숫자. 8 바이트로 저장

TEXT 텍스트. UTF-8, UTF-16BE or UTF-16-LE 중 하나에 저장

BLOB Binary Large Object. 입력 데이터를 그대로 저장

-- 테이블 만들기

테이블 구조 => 스키마

필드명과 자료형 지정

NOT NULL(필수항목)

PRIMARY KEY(기준키)

AUTOINCREMENT(자동숫자증가)

DEFAULT

Create Table & Insert Row

-- 테이블 생성하기

CREATE TABLE 테이블명 (필드명 자료형 옵션);

```
CREATE TABLE student (  
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    name      TEXT NOT NULL,  
    age       INTEGER NOT NULL,  
    addr      TEXT DEFAULT "서울" NOT NULL  
);
```

-- 테이블에 레코드 추가하기

INSERT INTO 테이블명 (컬럼1, 컬럼2...) VALUES (값1, 값2 ...);

```
INSERT INTO student (name, age, addr) VALUES ("김영희", 23, "대구");  
INSERT INTO student (name, age) VALUES ("고길동", 29);
```

Delete Row & Update Row

-- 테이블에 레코드 삭제하기

DELETE FROM 테이블명 WHERE 조건절;

```
DELETE FROM student WHERE name="고길동";
```

-- 테이블에 레코드 수정하기

UPDATE 테이블명 SET 컬럼명 1 = 값1, 컬럼명2 = 값2, ... WHERE 조건식;

```
UPDATE student SET addr = "제주" WHERE name = "마동탁";
```

Table Delete & Rename & Add Column

- 기존 테이블 복사해서 새로운 테이블 생성하기

CREATE TABLE 테이블명 AS SELECT * FROM 기존테이블명;

```
CREATE TABLE customers2 AS  
SELECT  
    CustomerId, FirstName, LastName, Country, Email  
FROM customers LIMIT 12;
```

- 테이블 삭제하기

DROP TABLE 테이블명;

```
DROP TABLE students;
```

- 테이블명 변경하기

ALTER TABLE 현재테이블명 RENAME TO 새로운테이블명;

```
ALTER TABLE customers2 RENAME TO customers3;
```

Table Delete & Rename & Add Column

- 컬럼명 변경하기

ALTER TABLE 테이블명 RENAME COLUMN 컬럼명1 TO 컬럼명2;

```
ALTER TABLE customers2 RENAME COLUMN Country TO Nationality;
```

- 컬럼 추가하기

ALTER TABLE 테이블명 ADD COLUMN 컬럼명 자료형;

```
ALTER TABLE customers2 ADD COLUMN hobby TEXT;
```

CSV Import

- Csv 파일을 이용한 테이블 생성

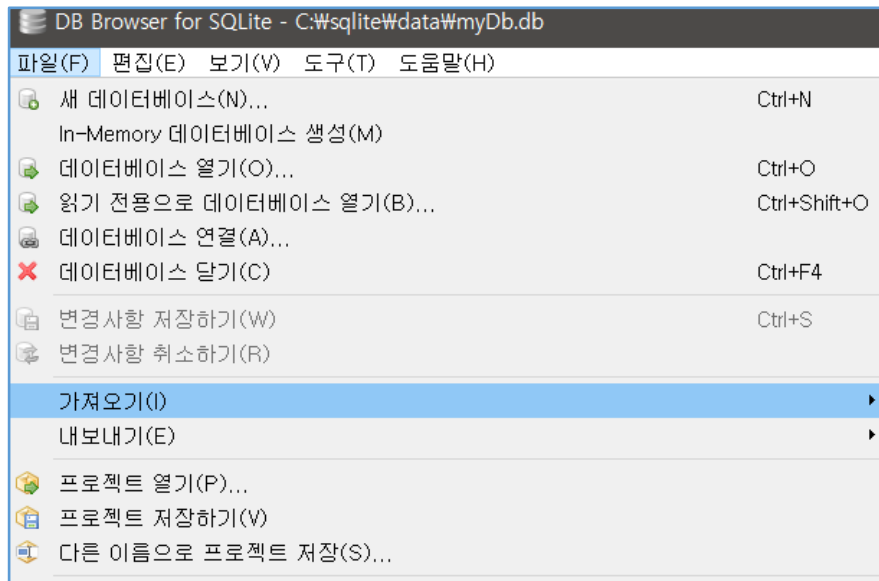
.mode csv

.import csv파일경로 테이블명

```
sqlite> .mode csv
```

```
sqlite>.import usertbl.csv usertbl
```

- [파일]-[가져오기]-[CSV]



CSV Export

- 테이블을 csv 파일로 저장하기

.mode csv

.output csv파일경로

SELECT * FROM 테이블명;

.quit

```
sqlite> .headers on
```

```
sqlite> .mode csv
```

```
sqlite> .output data.csv
```

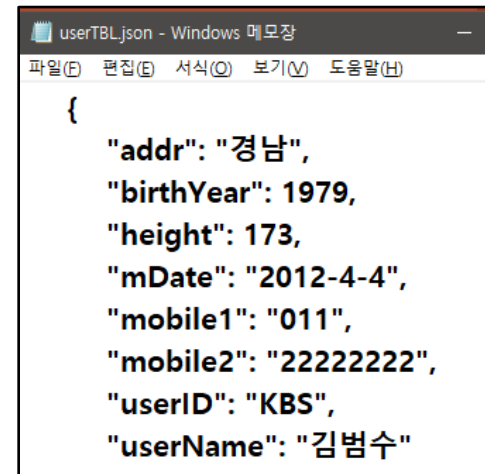
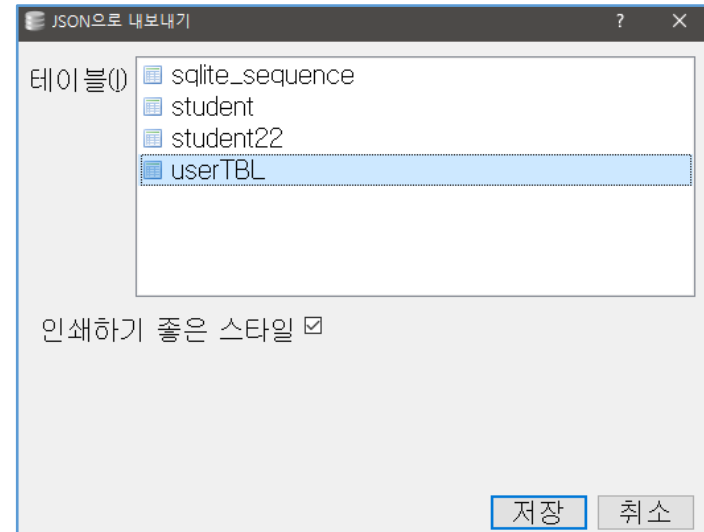
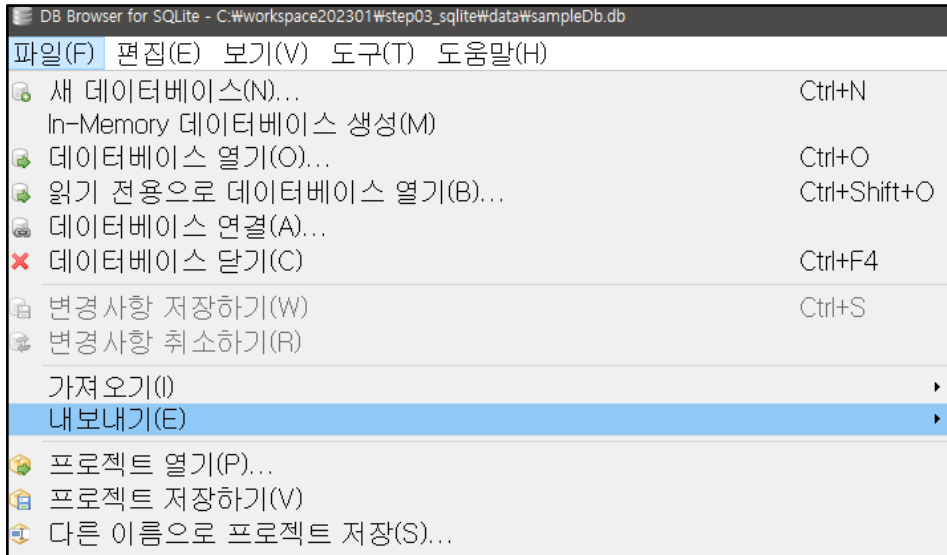
```
sqlite> SELECT customerid,firstname,lastname FROM customers;
```

```
sqlite> .quit
```

```
C:\sqlite>dir *.csv
```

CSV Export

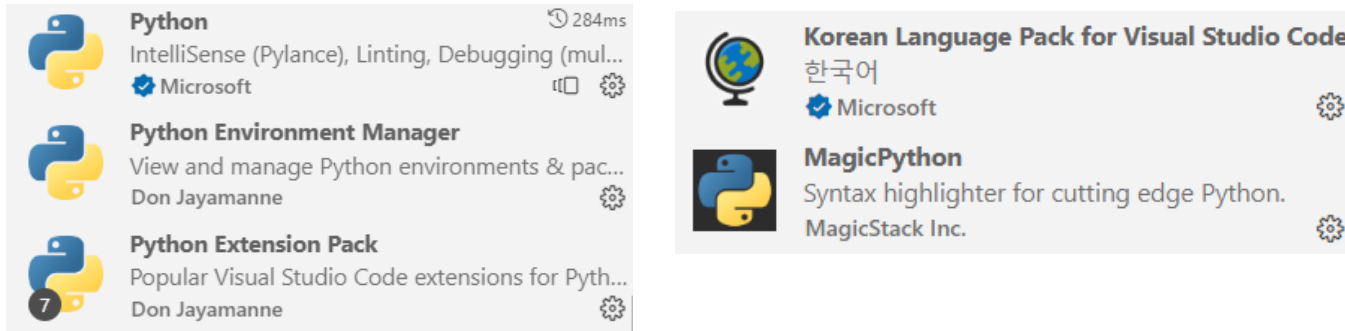
▪ [파일]-[내보내기]



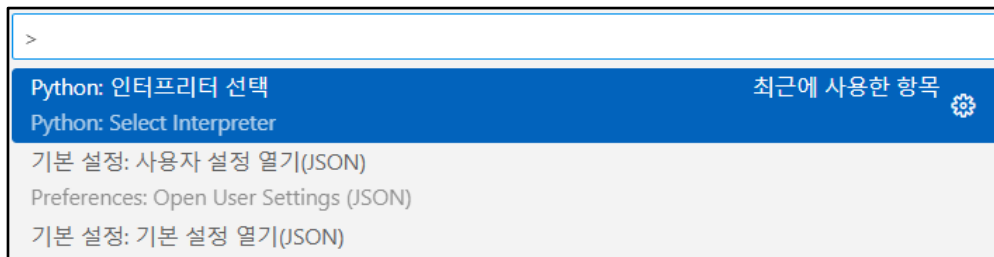
SQLITE + Python 연동

VSCODE+Python

■ Extention 설치



■ 인터프리터 설정 : ctrl+shift+p



Sqlite DB 연결

연결변수 생성

연결변수명 = sqlite3.connect(데이터베이스경로)

```
import sqlite3
```

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

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

```
print(cursor, type(cursor))
```

테이블 생성

테이블 생성 => book.db > book

필드와 자료형

아이디 id integer = > 기본키, 필수항목, 자동숫자증감

책제목 titiel text = > 필수항목

출판사 writer text = > 필수항목

페이지수 page int = > 필수항목

가격 price int = > 필수항목

create table 테이블명 (필드명 자료형 옵션);

execute(sql명령어)

테이블 생성

```
cursor.execute('''
    CREATE TABLE IF NOT EXISTS book1 (
        id integer not null primary key autoincrement,
        titiel text not null,
        writer text,
        page integer,
        price integer
    );
''')

conn.commit()

conn.close()
```

레코드 삽입

레코드 삽입 sql 명령 실행 후 데이터베이스 반영

```
cursor.execute('''
    insert into book1
    values(null, '파이썬완성', '홍길동', 300, 35000);
''')
conn.commit();
```

컬럼명이 있는 sql 명령어

```
cursor.execute('''
    insert into book1 (titiel, writer, page, price)
    values('장고로 웹사이트 제작', '김장고', 400, 40000);
''')

conn.commit();
```

레코드 삽입

executemany() 를 이용한 다중 데이터 삽입

```
sql = "insert into book1 (titiel, writer, page, price) values (?, ?, ?, ?);"  
data_list = [  
    ('해리포터1', '조앤 K. 롤링', 500, 25000),  
    ('해리포터2', '조앤 K. 롤링', 500, 25000),  
    ('해리포터3', '조앤 K. 롤링', 500, 25000)  
]  
cursor.executemany(sql, data_list)  
conn.commit()
```

레코드 조회

```
# fetchall() => 리스트로 반환  
# fetchone()  
# fetchmany(갯수) => 리스트로 반환
```

```
cursor.execute('select * from book1 order by id desc')  
result = cursor.fetchall()  
print('='*30)  
for row in result:  
    print(row)
```


레코드 수정

```
cursor.execute(' update book1 set price=10000 where id=2;')  
conn.commit();
```

```
sql = ' update book1 set price = ? where writer = ? ;'  
cursor.execute(sql, (12000, '홍길동'))  
conn.commit();
```

레코드 삭제

```
cursor.execute('delete from book1 where id = 1')  
conn.commit()
```

```
sql = 'delete from book1 where writer = ?;'  
cursor.execute(sql, ('조앤 K. 롤링',))  
conn.commit()
```

퀴즈

sqlite를 이용하여 주소록 프로그램을 완성하여라.

```
=====
주소록 메뉴
=====

1. 연락처 입력
2. 연락처 출력
3. 연락처 수정
4. 연락처 삭제
5. 연락처 초기화
6. 종료

메뉴선택: 2
-----
번호   이름           핸드폰           이메일           주소
-----
5      홍길동           010-7890-9999   hgd@gogo.com     대구광역시

메뉴선택: 3
수정할 code를 입력하세요.5
수정할 메뉴를 선택하세요.
1.이름 2.핸드폰 3.이메일 4.주소 :
이름 : 박길동

메뉴선택: 1
=====
레코드 삽입
이름 : 홍길동
핸드폰 : 010-7890-9999
이메일 : hgd@gogo.com
주소 : 대구광역시

메뉴선택: 4
삭제하실 code를 입력하세요: 6
해당 레코드를 삭제했습니다.

메뉴선택: 5
레코드가 모두 삭제되었습니다.

메뉴선택: 6
프로그램 종료
```