

데이터 저장

데이터 저장 방법 선택

■ 파일

- 기계가 읽을 수 있는 파일 : csv, json, xml
- 그 중 csv 형식으로 파일 저장 실습

■ 데이터베이스

- 관계형 데이터베이스 : MySQL, PostgreSQL, SQLite
- NoSQL (비관계형 데이터베이스)
- 그 중 SQLite로 DB저장 실습

■ API (서비스 관점)

데이터셋 소개

- 공공데이터포털, 서울 열린데이터광장, 국가통계포털 등

- <https://kbig.kr/portal/kbig/datacube/dataset/info>

The screenshot shows the K-ICT Big Data Center website. The header includes the K-ICT logo and navigation links such as '로그인', '회원가입', '사이트맵', and '검색'. The main navigation bar contains '데이터공유', '분석인프라 이용 신청', '데이터결합지원', '지식공유', '주요사업', and '센터소개'. The page title is '데이터셋 소개'. A breadcrumb trail shows 'HOME > 데이터공유 > 데이터셋 > 데이터셋 소개'. A sidebar on the left lists '데이터공유' and '데이터셋' with sub-items like '데이터셋재정보', '교육 동영상', '교육실습콘텐츠', and '형태소사전'. The main content area is titled '데이터셋 관련 서비스 바로가기' and contains a table of services.

이름	바로가기
통합데이터지도	
AI 허브	
공공데이터포털	
국가통계포털	
마이크로데이터	
지역데이터개방	
서울 열린데이터광장	

CSV (Comma Separated Values)

■ 형식

- 레코드 내의 각 필드가 콤마(쉼표)로 구분되어 있는 파일
- 각 레코드는 줄 바꿈 문자로 구분
- 확장자: .csv

■ 실험 데이터^{Experiment Data} [1] ref. chap3

- github : <https://github.com/jackiekazil/data-wrangling>
- data-text.csv (github) : 국가별 기대수명 데이터

① csv.writer()

■ csv module로 일부 행들만 쓰기

```
import csv

rows = []
with open('data-text.csv') as f:
    reader = csv.reader(f)
    for row in reader:
        rows.append(row)

#csv write
with open('data-text20.csv', 'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerows(rows[:20])
```

코드 설명

■ `open(newline='')`

```
csv.writer(csvfile, dialect='excel', **fmtparams)
```

Return a writer object responsible for converting the user's data into delimited strings on the given file-like object. `csvfile` can be any object with a `write()` method. If `csvfile` is a file object, it should be opened with `newline=''` [1]. An optional `dialect` parameter can be given which is used to define a set of parameters

Footnotes

1(1,2) If `newline=''` is not specified, newlines embedded inside quoted fields will not be interpreted correctly, and on platforms that use `\r\n` linendings on write an extra `\r` will be added. It should always be safe to specify `newline=''`, since the csv module does its own (**universal**) newline handling.

■ `.writerows(iterable)`

- 예를들어, list를 파라미터로 주면 list의 모든 행을 저장

```
csvwriter.writerows(rows)
```

Write all elements in `rows` (an iterable of `row` objects as described above) to the writer's file object, formatted according to the current dialect.

② pandas to_csv()

- pandas module로 일부 행들만 쓰기

```
import csv

file_path = 'data-text.csv'
df = pd.read_csv(file_path, encoding='utf-8')

#iloc인덱서 & slicing
df100 = df.iloc[:100]

#csv write
df100.to_csv('data-text100.csv')
```

데이터 베이스

■ DB vs. DBMS (DB Management System)

- user (query) - DBMS (S/W) - H/W : DB
- 분류 : Relational DB vs. NoSQL (Non-relational DB)
- RDBMS의 예
: MySQL, PostgreSQL, SQL Server (M/S), Oracle

■ SQL (Structured Query Language)

- SQL은 RDBMS에서 사용하는 query 언어
- 발음 SQL or SEQUEL (Structured English QUery Language - 초기 IBM)
- SQL vs. NoSQL

데이터 베이스 선택

■ 관계형 데이터베이스

- 상용화 시에는 MySQL이나 PostgreSQL등이 많이 쓰임
- SQLite는 파이썬에서 기본으로 지원하는 데이터베이스이고 파일 기반이기 때문에 별도의 설치나 설정이 필요 없어 데이터베이스 조작을 연습하거나 빠른 프로토타입을 개발할 때 유용 cf. Data Migration
- SQLite vs. MySQL
<http://www.itworld.co.kr/news/117213#csidx8f2059f6f107c3a8c56b0f4faeba392>
- MySQL vs. Maria DB
<http://www.itdaily.kr/news/articleView.html?idxno=53340>

■ NoSQL

- 비관계형 데이터베이스
- 모든 관계에 대한 맵핑없이 플랫폼 형식으로 데이터 저장
-  mongoDB. <https://www.mongodb.com/>

관계형 데이터베이스 기초

■ RDB vs. Excel

- 관계형 데이터베이스(RDB)의 동작을 Excel과 비교해서 이해하기
- RDB는 Excel과 비슷하게 생겼는데, RDB의 Table은 Excel의 sheet로 생각하면 됨
- Table은 sheet와 같이 행(row)과 열(column)로 구성되어 있음

■ SQL -- RDB

- SQL 명령어 (command) 로 SQL 문 (statement) 을 만들어 RDB의 데이터를 처리 (CRUD)

RDB Table

■ RDB Table 예제

- 열(column) 3개와 행(row) 4개인 Table의 예를 보면,

학번	이름	학점
2017001	홍길동	4.2
2020005	장길산	2.8
2019010	고주몽	3.5
2016007	신윤복	3.2

- 학번, 이름, 학점의 3개 열이 있고, 홍길동, 장길산, 고주몽, 신윤복을 포함하는 4개의 행이 있음

SQL Query

■ SQL Query

- 데이터베이스 생성(Create), 조회(Read), 갱신(Update), 삭제 (Delete) 작업 : CRUD
- SQL명령어 : SELECT, INSERT, INTO ...
(SQL은 프로그래밍 언어는 아니고, Query언어임)

■ SQL 예시

- 이전 페이지의 테이블명을 GRADE라고 하고, 학생을 추가하는 SQL 명령문을 작성해보면,
INSERT INTO GRADE VALUES (2018002, “ 김삿갓“, 2.7)

SQLite

■ SQLite란

- SQLite는 내장 가능한 오픈소스 데이터베이스로, C로 작성됐으며 일반적인 SQL로 쿼리가 가능
- Anaconda를 설치하면 파이썬 내부에 기본적으로 설치가 되는 sqlite3모듈을 사용해서 데이터베이스를 생성, 조회, 갱신, 삭제 (CRUD) 작업 가능
즉, 모듈 install이 필요 없음

■ 파이썬으로 sqlite3 활용하는 2가지 방법

- 하나는 DB에 연결하고 커서를 만든 다음에 SQL문을 보내서 DB에 저장
- 또 하나는 판다스 데이터프레임을 이용하면 함수를 통해 쉽게 저장

sqlite3 module

■ Python Standard Library

- <https://docs.python.org/3/library/index.html> Data Persistence

sqlite3 — DB-API 2.0 interface for SQLite databases

Source code: [Lib/sqlite3/](#)

SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

The sqlite3 module was written by Gerhard Häring. It provides a SQL interface compliant with the DB-API 2.0 specification described by [PEP 249](#).

To use the module, you must first create a `Connection` object that represents the database. Here the data will be stored in the `example.db` file:

```
import sqlite3
conn = sqlite3.connect('example.db')
```

코드 설명1

■ Tutorial in Python Standard Library

First, we need to create a new database and open a database connection to allow `sqlite3` to work with it. Call `sqlite3.connect()` to create a connection to the database `tutorial.db` in the current working directory, implicitly creating it if it does not exist:

```
import sqlite3
con = sqlite3.connect("tutorial.db")
```

The returned `Connection` object `con` represents the connection to the on-disk database.

In order to execute SQL statements and fetch results from SQL queries, we will need to use a database cursor. Call `con.cursor()` to create the `Cursor`:

```
cur = con.cursor()
```

Now that we've got a database connection and a cursor, we can create a database table `movie` with columns for title, release year, and review score. For simplicity, we can just use column names in the table declaration – thanks to the `flexible typing` feature of SQLite, specifying the data types is optional. Execute the `CREATE TABLE` statement by calling `cur.execute(...)`:

코드 설명2

■ execute() & commit()

```
cur.execute("CREATE TABLE movie(title, year, score)")
```

Now, add two rows of data supplied as SQL literals by executing an `INSERT` statement, once again by calling `cur.execute(...)`:

```
cur.execute("""
    INSERT INTO movie VALUES
        ('Monty Python and the Holy Grail', 1975, 8.2),
        ('And Now for Something Completely Different', 1971, 7.5)
    """)
```

The `INSERT` statement implicitly opens a transaction, which needs to be committed before changes are saved in the database (see [Transaction control](#) for details). Call `con.commit()` on the connection object to commit the transaction:

```
con.commit()
```

① sqlite3 준비

■ SQLite DB준비

```
import sqlite3
```

```
#1. DB 연결 (파일이 없으면 만들고 있으면 연결)  
con = sqlite3.connect("C:/Temp/sqlite3/test.db")
```

```
#2. cursor 객체 만들기  
cursor = con.cursor()
```

```
#3. 커서객체.execute()로 SQL 명령  
    ( 다음 페이지에 )
```

```
#4. DB에 반영  
con.commit()
```

```
#5. DB 연결 종료  
con.close()
```

① SQL query

■ SQL query 보내기

- 데이터베이스 생성, 조회, 갱신, 삭제 (CRUD) 작업
- 커서객체.execute(“ SQL query “)

#3-1. table만들기

```
cursor.execute("CREATE TABLE school(name, number, math_score)")
```

#3-2. 값 입력

```
cursor.execute('INSERT INTO school VALUES("James", 1, 80)')
```

#3-3. 변수 입력

```
for row in rows:
```

```
    cursor.execute("INSERT INTO school VALUES(?, ?)", (row[0],  
row[5]))
```

② pandas to_sql()

■ pandas로 SQL 활용

```
#1. DB 연결
con = sqlite3.connect("C:/Temp/sqlite3/test.db")

# DataFrame 생성 예
data = {'name' : ['Jerry', 'Riah', 'Paul'],
        'algo1' : ['A', 'A+', 'B'],
        'basic' : ['C', 'B', 'B+'],
        'c++' : ['B+', 'C', 'C+']}
df = pd.DataFrame(data)

#2. DB 저장
df.to_sql('score', con, if_exists='append', index=False)

#3. DB 연결 종료 (이전 예제 참고)
```

SQLite Expert

■ GUI기반 SQLite DB 관리 및 개발 프로그램



[HOME](#) [FEATURES](#) [SCREENSHOTS](#) [DOWNLOAD](#) [PRICING](#) [SUPPORT](#) [HISTORY](#) [FAQ](#)



Visual SQL Query Builder

Build complex SQL queries with ease.
[Formatted SQL query text layout](#).
Powerful means of SQL query parsing and analysis.
Advanced SQL editor with syntax highlighting and code completion.



Powerful restructure capabilities

Visual editors for table columns, indexes, foreign keys, triggers, unique and check constraints.
Restructure any complex table without losing data.
Any restructure operation is wrapped in a nested transaction which is rolled back if any errors occur when applying changes.



Import and Export data

Import data from ADO data source, CSV files, SQL script or SQLite.
Export data to CSV files, SQL script, Excel or SQLite.
Copy tables between SQLite databases using drag and drop operations.
Export data to Excel via clipboard.



Data editing

Edit data in the grid using powerful in-place editors.
[Image editor](#) currently supporting JPEG, PNG, BMP, GIF and ICO image formats.
Visualize and modify BLOB fields using the integrated [Hex editor](#).



Other features

Full Unicode support.
Support for attached databases.
Supports SQLite extensions and third-party SQLite libraries.
Lua and Pascal scripting support.



Recent features

View all the functions and collations installed by each SQLite extension!
New in 3.5: Added support for partial indexes.
New in 3.5: Added support for tables without rowid.

SQLite Expert 설치

■ SQLite Expert 를 다운받아 설치

- <http://sqliteexpert.com/download.html>

: SQLite Expert Personal 64bit 5.x 버전 freeware 다운



HOME FEATURES SCREENSHOTS DOWNLOAD PRICING SUPPORT HISTORY FAQ

SQLite Expert Personal 5.x

License	Freeware
Version	5.3.5
Release date	February 2, 2020
Platform	Windows 2000/XP/2003/Vista/7/8/10
Size	35 MB

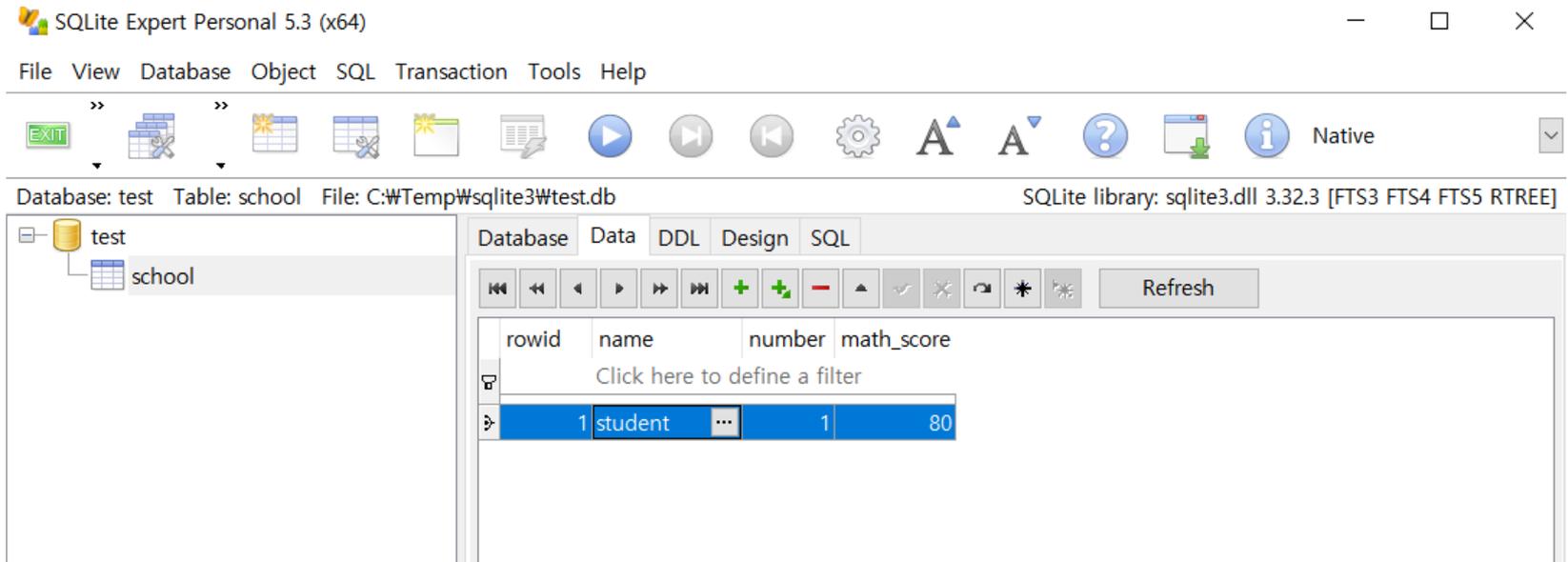
[SQLite Expert Personal 64bit](#) MD5

[SQLite Expert Personal 32bit](#) MD5

SQLite Expert Personal is freeware and does not have an expiration date.

SQLite Expert 활용

- **SQLite Expert** 로 프로그램에서 작성한 DB 결과 확인
 - File / Open database로 db파일을 열고, Data탭으로 데이터 확인



참고도서

➤ reference

[1] 파이썬을 활용한 데이터길들이기

- 프로그래밍인사이트

[2] 모두의 데이터분석

- 길벗

[3] 파이썬머신러닝 판다스데이터분석

- 정보문화사

End