

DB 테이블 만들기

이번 시간에서는 DB 테이블 만들기에 대해서 알아보겠습니다.

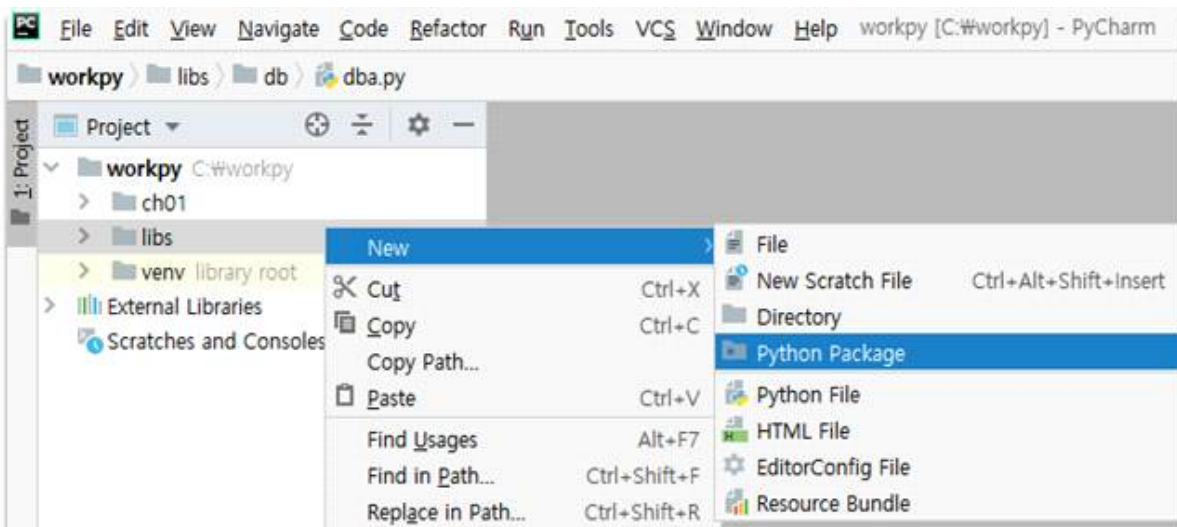
1. 테이블 만들기

테이블을 사용하기 위해서는 커넥션을 먼저 생성해야 합니다.

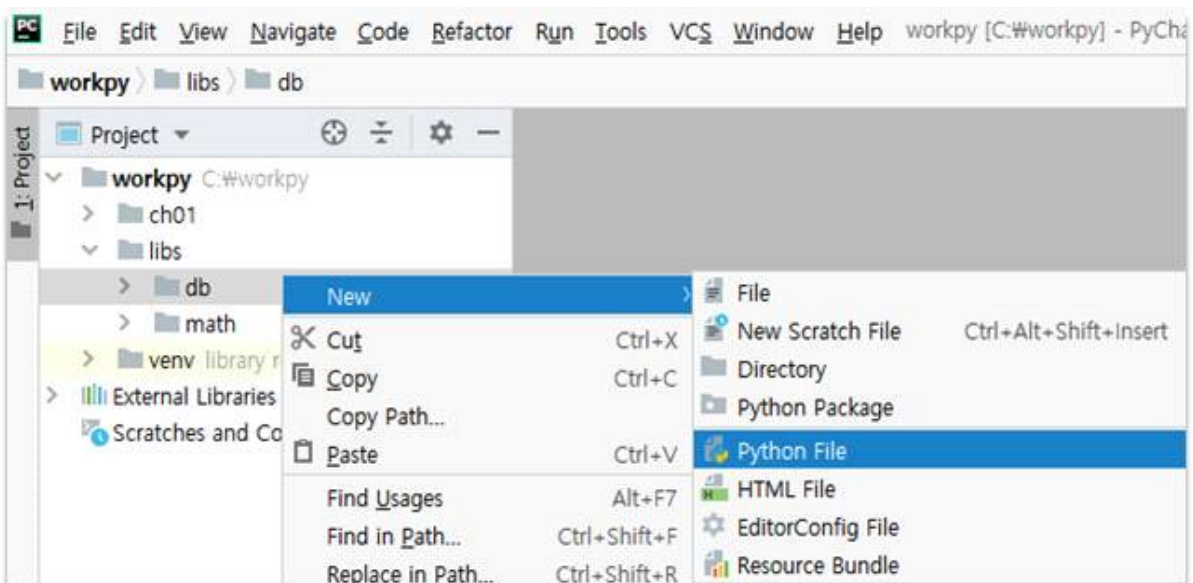
테이블에 값을 넣거나, 수정, 삭제 등을 할 때 모두 필요한 것이 커넥션 입니다.

커넥션은 자주 사용하기 때문에 라이브러리 형태 즉 모듈로 만들어 놓고 사용하는 것이 좋습니다.

libs 마우스 오른쪽 클릭. New → Python Package 에서 이름이 db인 패키지를 생성합니다.



db 패키지에 dba.py 파일을 생성합니다.



db 패키지 안의 dba.py 에 다음과 같이 작성합니다.

```
import sqlite3

def getConn():
    conn = sqlite3.connect('c:/test/abc.db')

    return conn
```

→ 'sqlite3' 모듈을 import

데이터가 저장될 데이터 베이스 파일에 연결. 커넥션을 생성하고 리턴 한다.

dba.py

```
import sqlite3
```

```
def getConn():
```

```
    conn = sqlite3.connect('c:/test/abc.db') # db 접속과 관련 된 정보를 가진 커넥션 객체생성
    return conn
```

만든 모듈을 가져와서 테이블을 생성하는 p15.py 에는 다음과 같이 작성합니다.

```
from libs.db.dba import getConn

def create_table():
    conn = getConn()
    cur = conn.cursor()

    cur.execute('''
        create table test(name text,
        kor int,
        eng int,
        math int)
        ''')

    conn.commit()
    conn.close()

if __name__ == '__main__':
    create_table()
```

→ 'dba' 모듈을 import

'dba' 모듈의 getConn()을 호출해서 커넥션을 가져온다. 커넥션을 이용해서 커서를 생성한다.

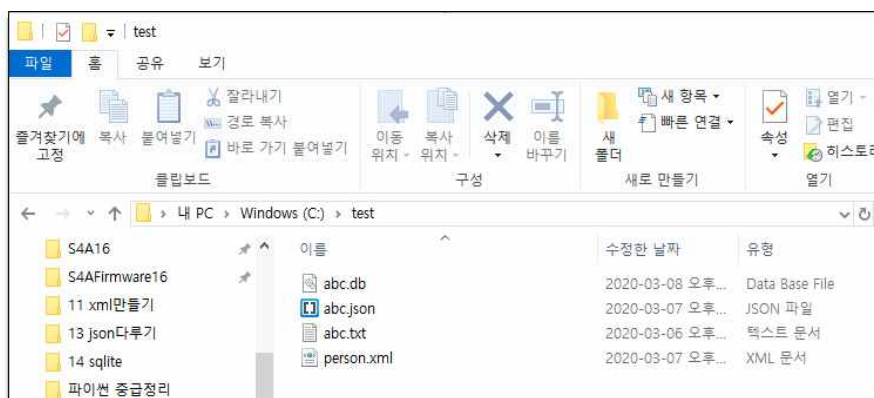
테이블 생성하는 쿼리를 작성하고, 쿼리 실행.

테이블 이름은 test.
필드는 이름,국어,영어, 수학.
text, int는 자료형

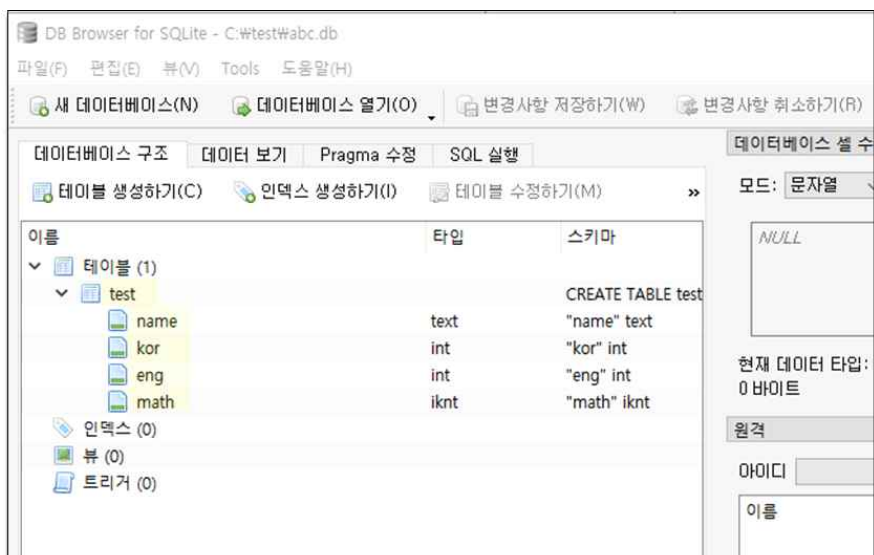
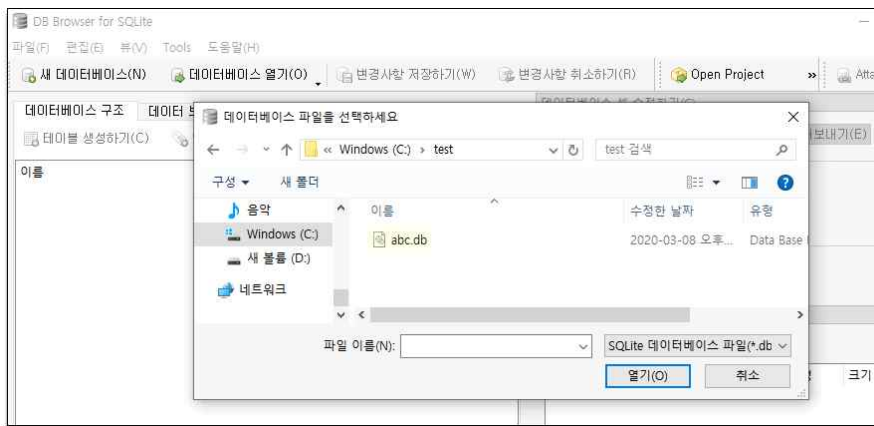
최종적으로 반영한다는 의미로 commit() 커넥션을 닫는다.

이 파일이 main으로서 동작할 때만 if 안의 코드를 실행하라

다음과 같이 데이터베이스 파일이 생성되었습니다.



DB Browser를 열어서 확인해보겠습니다.



p15.py

```
from libs.db.dba import getConn
```

```
def create_table():
    conn = getConn()          # 커넥션을 가져옵니다.
    cur = conn.cursor()       # 커넥션을 이용해서 커서를 생성합니다.

    # 쿼리를 작성하고 실행
    cur.execute('''
create table test(name text,
kor int,
eng int,
math iknt)
''')
    conn.commit()             # 최종적으로 반영하겠다
    conn.close()              # 커넥션을 닫는다

# 이 파일이 main으로 동작할 때만 아래의 구문을 수행하라
if __name__ == '__main__':
    create_table()
```