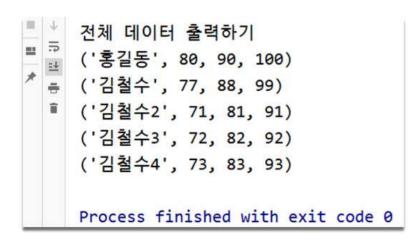
SELECT 사용하기

이번 시간에서는 테이블에 데이터를 조회하는 SELECT에 대해서 알아보겠습니다.

1. 테이블의 데이터조회 하기

```
import sqlite3
                                   'sqlite3' 'dba' 모듈을 import
from libs.db.dba import getConn
def select a():
                                    'dba' 모듈의 getConn()을 호출해서 커넥션을 가져온다.
   conn = getConn()
                                   커넥션을 이용해서 커서를 생성한다.
   cur = conn.cursor()
                                   데이터를 조회하는 쿼리를 작성하고, 쿼리 실행.
   cur.execute('select * from test')
                                   test 테이블의 데이터를 모두 가지고 오겠다.
   print('전체 데이터 출력하기')
                                   가져온 결과를 담을 rs 객체를 만듭니다.
   rs = cur.fetchall()
                                   커서를 통해서 데이터를 모두 가져와서 리스트로 반환.
   for i in rs:
                                   리스트(rs)에 있는 데이터를 반복을 통해서 출력.
      print(i)
   conn.close()
if __name__ == '__main__':
                                    이 파일이 main으로서 동작할 때만
   select a()
                                   if 안의 코드를 실행하라
```



```
р17.ру
```

```
import sqlite3
from libs.db.dba import getConn

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

    cur.execute('select * from test')

    print('전체 데이터 출력하기')

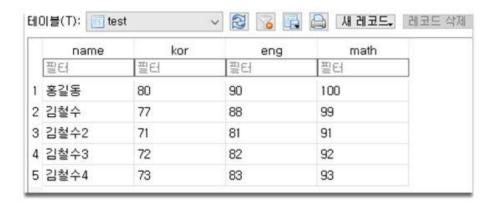
    rs = cur.fetchall()

    for i in rs:
        print(i)

    conn.close()

if __name__ == '__main___':
```

select_a()



2. 테이블에서 조회한 데이터 중 일부만 가져오기

이번에는 select_b() 함수의 인자에 가져올 데이터의 개수를 넣습니다. cur 커서의 메서드 fetchmany() 의 인자 num은 가져올 데이터의 행수를 의미합니다.

```
import sqlite3
                                    'salite3' 'dba' 모듈을 import
from libs.db.dba import getConn
def select_b(num):
                                    'dba' 모듈의 getConn()을 호출해서 커넥션을 가져온다.
   conn = getConn()
                                    커넥션을 이용해서 커서를 생성한다.
   cur = conn.cursor()
   cur.execute('''
                                    데이터를 조회하는 쿼리를 작성하고, 쿼리 실행.
   select * from test
                                    test 테이블에서 name이 홍길동인 사람을 가져오겠다.
   where name="홍길동"
   111)
   print('전체 데이터 출력하기')
                                    가져온 결과를 담을 rs 객체를 생성.
   rs = cur.fetchmany(num)
                                    매개변수로 받은 num 만큼 데이터를 가져온다.
                                    커서를 통해서 데이터를 모두 가져와서 리스트로 반환.
   for i in rs:
                                   리스트(rs)에 있는 데이터를 반복을 통해서 출력.
      print(i)
   conn.close()
if __name__ == '__main__':
                                    이 파일이 main으로서 동작할 때만 if 안의 코드를 실행
   select b(1)
```

C:\workpy\venv\Scripts\python.exe C:/workpy/ch01/p17_1.py 전체 데이터 출력하기 ('홍길동', 80, 90, 100)

```
import sqlite3
from libs.db.dba import getConn
def select_b(num):
   conn = getConn()
                   # 커넥션을 가져옵니다.
   cur = conn.cursor() # 커넥션을 이용해서 커서를 생성합니다.
   # name이 홍길동인 데이터를 가지고 와라
   cur.execute('''
   select * from test where name="홍길동"
   ''')
   print('전체 데이터 출력하기')
   # 개수만큼 가지고 와라
   rs = cur.fetchmany(num)
   # print(rs)
   for i in rs:
      print(i)
                  # 커넥션을 닫는다
   conn.close()
# 이 파일이 main으로 동작할 때만 아래의 구문을 수행하라
if __name__=='__main__':
   select_b(1)
```

3. 테이블에서 조회한 데이터 중 일부만 가져오기(동적 쿼리)

이번에는 select_c() 함수의 인자에 가져올 데이터의 개수와 이름을 넣습니다. 쿼리는 다음과 같이 동적 쿼리로 작성합니다. 동적 쿼리로 작성할 때 커서 cur의 execute 메서드의 두 번째 인자는 튜플이어야 합니다.

```
import sqlite3
                                   'sqlite3' 'dba' 모듈을 import
from libs.db.dba import getConn
def select c(num, name):
                                   'dba' 모듈의 getConn()을 호출해서 커넥션을 가져온다.
   conn = getConn()
                                   커넥션을 이용해서 커서를 생성한다.
   cur = conn.cursor()
   cur.execute('''
                                   데이터를 조회하는 쿼리를 작성하고, 쿼리 실행.
   select * from test
                                   test 테이블에서 name이 홍길동인 사람을 가져오겠다.
   where name=?''', (name,))
                                   동적쿼리로 작성.
                    두 튜플
   print('전체 데이터 출력하기')
                                   가져온 결과를 담을 rs 객체를 생성.
   rs = cur.fetchmany(num)
                                   매개변수로 받은 num 만큼 데이터를 가져온다.
                                   커서를 통해서 데이터를 모두 가져와서 리스트로 반환.
   for i in rs:
                                   리스트(rs)에 있는 데이터를 반복을 통해서 출력.
      print(i)
   conn.close()
if name == ' main ':
                                   이 파일이 main으로서 동작할 때만 if 안의 코드를 실행
   select c(1, '홍길동')
```

C:\workpy\venv\Scripts\python.exe C:/workpy/ch01/p17_1.py 전체 데이터 출력하기 ('홍길동', 80, 90, 100)

```
import sqlite3
from libs.db.dba import getConn
def select_c(num, name):
   conn = getConn()
                   # 커넥션을 가져옵니다.
   cur = conn.cursor() # 커넥션을 이용해서 커서를 생성합니다.
   # 동적쿼리 적용
   cur.execute('''
   select * from test where name=?
   ''', (name,) ) # 튜플이어야 한다.
   print('전체 데이터 출력하기')
   # 개수 만큼 가지고 와라
   rs = cur.fetchmany(num)
   # print(rs)
   for i in rs:
      print(i)
   conn.close()
                  # 커넥션을 닫는다
# 이 파일이 main으로 동작할 때만 아래의 구문을 수행하라
if __name__=='__main__':
   select_c(1,'홍길동')
```