

파이썬 SQLite 연동

SQLite3 모듈

❖ 파이썬 SQLite3 지원 모듈

- sqlite3

```
import sqlite3
```

SQLite3 모듈

❖ 데이터베이스 연결 및 닫기

- 데이터베이스 연결 성공시 `sqlite3.Connection` 객체 리턴

```
import sqlite3
con = sqlite3.connect("파일경로")    # Connection 객체 리턴
```

- 파일이 존재하면 기존 파일 사용
- 파일이 존재하지 않으면 새로 생성

```
import sqlite3

try :
    con = sqlite3.connect("contact.db")
    print(con)
except Exception as err:
    print('데이터베이스 에러 : %s' % err)
finally:
    con.close()
```

SQLite3 모듈

❖ 데이터베이스 연결 및 닫기

```
import sqlite3

with sqlite3.connect("contact.db") as con:
    try:
        print(con)
    except Exception as err:
        print('데이터베이스 에러 : %s' % err)
```

SQLite3 모듈

❖ Connection 객체로 SQL문 실행하기

- `con.execute(sql [,parameters...])`

SQLite3 모듈

❖ Connection 객체로 SQL문 실행하기

- SQL 문장 실행 객체 : `cursor.execute()`

```
with sqlite3.connect("contact.db") as con:
    sql = """
        INSERT INTO CONTACTS(NAME, CELL_PHONE, EMAIL)
        VALUES('고길동', '010-1111-3333', 'gogd@naver.com')
    """;
    try:
        con.execute(sql)
        print('데이터 삽입 저장 완료')
    except Exception as err:
        print('데이터베이스 에러 : %s' % err)
    else:
        con.commit()
```

SQLite3 모듈

❖ Connection 객체로 SQL문 실행하기

- Parameterized Query : ? Placeholder
 - sql 문장의 구성 번거러움 → ? 매개변수로 처리

```
name = '이세돌'
cell_phone = '010-3333-3333'
email = 'lee@gmail.com'

with sqlite3.connect("contact.db") as con:
    sql = """
        INSERT INTO CONTACTS(NAME, CELL_PHONE, EMAIL)
        VALUES(?, ?, ?)
    """;
    try:
        con.execute(sql, (name, cell_phone, email))
        print('데이터 삽입 저장 완료')
    except Exception as err:
        print('데이터베이스 에러 : %s' % err)
    else:
        con.commit()
```

SQLite3 모듈

❖ Connection 객체로 SQL문 실행하기

- Parameterized Query : Named Placeholder Query
 - 매개변수 위치에 ':이름' 지정 → dict로 실제 값 전달

```
with sqlite3.connect("contact.db") as con:
    sql = """
        INSERT INTO CONTACTS(NAME, CELL_PHONE, EMAIL)
        VALUES(:name, :cell_phone, :email)
    """;
    try:
        con.execute(sql,
                    {name : name, cell_phone:cell_phone, email:email})
        print('데이터 삽입 저장 완료')
    except Exception as err:
        print('데이터베이스 에러 : %s' % err)
    else:
        con.commit()
```


SQLite3 모듈

❖ Cursor 객체(sqlite3.Cursor)

- 다중 입력 : executemany()로 실행

```
data = (  
    ('홍길동1', '010-1111-1111', 'h1.gmail.com'),  
    ('홍길동2', '010-2222-2222', 'h2.gmail.com'),  
    ('홍길동3', '010-3333-3333', 'h3.gmail.com')  
)  
  
with sqlite3.connect("contact.db") as con:  
    sql = """  
        INSERT INTO CONTACTS(NAME, CELL_PHONE, EMAIL)  
        VALUES(?, ?, ?)  
    """;  
    try:  
        con.executemany(sql, data)  
        print('데이터 삽입 저장 완료')  
    except Exception as err:  
        print('데이터베이스 에러 : %s' % err)  
    else:  
        con.commit()
```

SQLite3 모듈

❖ 데이터 추출하기

- Cursor 객체(sqlite3.Cursor)
 - SELECT 문을 실행하고 결과를 관리

```
cursor = con.cursor()
cursor.execute(sql_문자열) 또는 cursor.executemany(sql_문자열)

# cursor로 추출한 rows 처리
```

- 추출된 행은 튜플 구성
 - 개별 컬럼은 인덱스로 접근
 - 엘리먼트는 SELECT 절에 지정한 컬럼 순으로 생성

SQLite3 모듈

❖ 데이터 추출하기

- Cursor 객체(sqlite3.Cursor)

```
with sqlite3.connect("contact.db") as con:
    sql = """
        SELECT * FROM CONTACTS
    """;
    try:
        cursor = con.cursor()
        cursor.execute(sql)
        for row in cursor:
            print(row)

    except Exception as err:
        print('데이터베이스 에러 : %s' % err)
    else:
        con.commit()
```

SQLite3 모듈

❖ 데이터 추출하기

○ 모든 행 추출하기

```
rows = cursor.fetchall()    # 모든 행(튜플)으로 구성된 리스트 리턴
for row in rows:
    print(row)
```

○ n개 행 추출하기

```
for row in cursor.fetchmany(5) : # 5개의 행(튜플)로 구성된 리스트 리턴
    print(row)
```

○ 1개 행 추출하기 : 주로 키로 쿼리를 한 경우 사용

```
row = cursor.fetchone()     # 1개의 행(튜플) 리턴
print(row)
```

SQLite3 모듈

❖ 데이터 수정하기

```
with sqlite3.connect("contact.db") as con:
    sql = """
        UPDATE CONTACTS
        SET
            ADDRESS = ?,
            BIRTHDAY = ?
        WHERE NAME=?
    """;
    try:
        name = input('수정할 이름 : ')
        address = input('주소 : ')
        birthday = input('생일 : ')
        con.execute(sql, (address, birthday, name))
    except Exception as err:
        print('데이터베이스 에러 : %s' % err)
    else:
        con.commit()
```

SQLite3 모듈

❖ 데이터 삭제하기

- Connection 객체로 실행

```
with sqlite3.connect("contact.db") as con:
    sql = """
        DELETE FROM CONTACTS WHERE NAME=?
    """;
    try:
        con = sqlite3.connect("contact.db")

        name = input('삭제할 이름 : ')
        con.execute(sql, (name,))

    except Exception as err:
        print('데이터베이스 에러 : %s' % err)
    else:
        con.commit()
```

SQLite3 모듈

❖ 실행결과 개수 추출

- `cursor.rowcount`

- INSERT, UPDATE, DELETE 문 실행 결과로 `cursor`를 리턴
- 리턴 받은 `cursor`의 `rowcount` 속성으로 추출 가능
- SELECT문인 경우 `rowcount` 속성의 값은 항상 -1