

■ 데이터베이스 (Database)

- MySQL / Oracle / MSSQL / MongoDB / SQLite 등 데이터베이스 사용 가능
- 데이터베이스 제품에 맞는 라이브러리(모듈) 사용
- 일반적인 사용 절차
 - 1) 데이터베이스에 맞는 모듈 import
 - 2) 데이터베이스 접속
 - 3) Cursor 객체 생성
 - 4) Cursor 객체의 SQL 실행 메소드를 사용하여 CRUD 실행
 - 조회의 경우 실행 결과를 추출
 - 5) 데이터베이스 작업 완료 후 commit (저장)
 - 6) Cursor 객체 close
 - 7) 접속 객체 close

■ MySQL

● 테이블 생성

```
import pymysql # 모듈 import

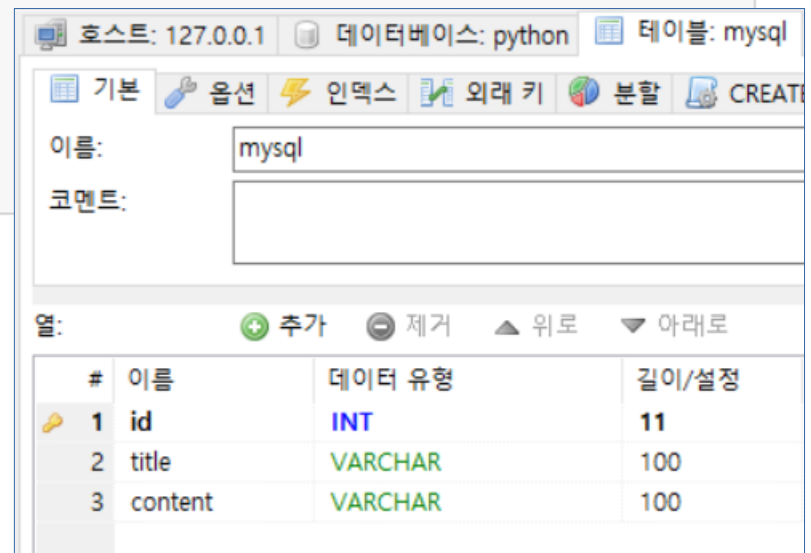
conn = pymysql.connect(
    host='localhost', user='root', password='mysql',
    db='python', charset='utf8') # 데이터베이스 접속

cursor = conn.cursor() # Cursor 객체 생성

sql = '''CREATE TABLE mysql (
        id INTEGER PRIMARY KEY AUTO_INCREMENT
        , title VARCHAR(100), content VARCHAR(100))'''
cursor.execute(sql) # SQL 실행

conn.commit() # 저장

cursor.close() # close
conn.close() # close
```



The screenshot shows a MySQL database management interface. The top bar indicates the host is 127.0.0.1, the database is 'python', and the table is 'mysql'. Below the bar, there are tabs for '기본' (Basic), '옵션' (Options), '인덱스' (Indexes), '외래 키' (Foreign Keys), '분할' (Partitions), and 'CREATE'. The '기본' tab is active, showing the table's structure. The '이름' (Name) field is set to 'mysql', and the '코멘트' (Comment) field is empty. Below the fields, there is a table showing the columns of the 'mysql' table.

#	이름	데이터 유형	길이/설정
1	id	INT	11
2	title	VARCHAR	100
3	content	VARCHAR	100

■ MySQL

● 예외 처리

```
import pymysql # 모듈 import

try:
    conn = pymysql.connect(
        host='localhost', user='root', password='mysql',
        db='pythom', charset='utf8') # 데이터베이스 접속

    cursor = conn.cursor() # Cursor 객체 생성

    sql = '''CREATE TABLE mysql (
        id INTEGER PRIMARY KEY AUTO_INCREMENT
        , title VARCHAR(100), content VARCHAR(100))'''
    cursor.execute(sql) # SQL 실행

    conn.commit() # 저장

    cursor.close() # close
    conn.close() # close

except pymysql.InternalError as e :
    print('예외 발생', e)
```

예외 발생 (1049, "Unknown database 'pythom'")

■ MySQL

● 데이터 입력

```
import pymysql

try:
    conn = pymysql.connect(
        host='localhost', user='root', password='mysql',
        db='python', charset='utf8')

    cursor = conn.cursor()

    sql = '''INSERT INTO mysql (id, title, content) VALUES (null, %s, %s)'''
    cursor.execute(sql, ('제목', '내용'))

    conn.commit()

    cursor.close()
    conn.close()
except pymysql.InternalError as e:
    print('InternalError', e)
except pymysql.ProgrammingError as e:
    print('ProgrammingError', e)
```

호스트: 127.0.0.1 데이터베이스: python 테이블: mysql 데이터			
python.mysql: 1 행 (중) (대략적) ▶ 다음 ⏮ 모두 보기			
id	title	content	
1	제목	내용	

■ MySQL

● with / finally (close 및 예외처리)

```
import pymysql

try:
    conn = pymysql.connect(
        host='localhost', user='root', password='mysql',
        db='python', charset='utf8')

    with conn.cursor() as cursor:
        sql = '''INSERT INTO mysql (id, title, content) VALUES (null, %s, %s)'''
        cursor.execute(sql, ('제목', '내용'))

    conn.commit()
except pymysql.InternalError as e:
    print('InternalError', e)
except pymysql.ProgrammingError as e:
    print('ProgrammingError', e)
finally:
    conn.close()
```

호스트: 127.0.0.1 데이터베이스: python 테이블: mysql 데이터			
python.mysql: 2 행 (총) (대략적) 다음 모두 보기			
id	title	content	
1	제목	내용	
2	제목	내용	

■ MySQL

● 데이터 수정

```
import pymysql

try:
    conn = pymysql.connect(
        host='localhost', user='root', password='mysql',
        db='python', charset='utf8')

    with conn.cursor() as cursor:
        sql = '''UPDATE mysql SET title = %s WHERE id = %s''' # 모든 형식을 문자열로
        cursor.execute(sql, ('제목수정', 1)) # 대입되는 값은 실제 데이터 형식

    conn.commit()
except pymysql.InternalError as e:
    print('InternalError', e)
except pymysql.ProgrammingError as e:
    print('ProgrammingError', e)
finally:
    conn.close()
```

호스트: 127.0.0.1 데이터베이스: python 테이블: mysql 데이터			
python.mysql: 2 행 (총) (대략적) ▶ 다음 ◀◀ 모두 보기 ▼			
id	title	content	
1	제목수정	내용	
2	제목	내용	

■ MySQL

● 데이터 삭제

```
import pymysql

try:
    conn = pymysql.connect(
        host='localhost', user='root', password='mysql',
        db='python', charset='utf8')

    with conn.cursor() as cursor:
        sql = '''DELETE FROM mysql WHERE id = %s'''
        cursor.execute(sql, (2))

    conn.commit()
except pymysql.InternalError as e:
    print('InternalError', e)
except pymysql.ProgrammingError as e:
    print('ProgrammingError', e)
finally:
    conn.close()
```

호스트: 127.0.0.1 데이터베이스: python 테이블: mysql 데이터			
python.mysql: 1 행 (총) (대략적) ▶ 다음 ⏮ 모두 보기 ▼			
id	title	content	
1	제목수정	내용	

■ MySQL

● 데이터 조회

```
import pymysql

try:
    conn = pymysql.connect(
        host='localhost', user='root', password='mysql',
        db='python', charset='utf8')

    with conn.cursor() as cursor:
        sql = '''SELECT id, title, content FROM mysql'''
        cursor.execute(sql)

        result = cursor.fetchall() # SQL SELECT 결과 모두 조회
        for row in result:
            print(row, type(row))
except pymysql.InternalError as e:
    print('InternalError', e)
except pymysql.ProgrammingError as e:
    print('ProgrammingError', e)
finally:
    conn.close()
```

(1, '제목수정', '내용') <class 'tuple'>

■ MySQL

● 데이터 조회

– fetchall() : 모두 조회

```
result = cursor.fetchall()
for row in result:
    print(row, type(row))
```

```
(1, '제목수정', '내용') <class 'tuple'>
(3, '제목', '내용') <class 'tuple'>
(4, '제목', '내용') <class 'tuple'>
(5, '제목', '내용') <class 'tuple'>
(6, '제목', '내용') <class 'tuple'>
(7, '제목', '내용') <class 'tuple'>
(8, '제목', '내용') <class 'tuple'>
(9, '제목', '내용') <class 'tuple'>
(10, '제목', '내용') <class 'tuple'>
(11, '제목', '내용') <class 'tuple'>
(12, '제목', '내용') <class 'tuple'>
```

👉 id	title	content
1	제목수정	내용
3	제목	내용
4	제목	내용
5	제목	내용
6	제목	내용
7	제목	내용
8	제목	내용
9	제목	내용
10	제목	내용
11	제목	내용
12	제목	내용

– fetchone() : 한개 데이터 조회

```
result = cursor.fetchone()
print(result, type(result))
```

```
(1, '제목수정', '내용') <class 'tuple'>
```

– fetchmany() : 여러개 데이터 조회

```
result = cursor.fetchmany()
print(result, type(result))
```

```
(1, '제목수정', '내용') <class 'tuple'>
(3, '제목', '내용') <class 'tuple'>
(4, '제목', '내용') <class 'tuple'>
(5, '제목', '내용') <class 'tuple'>
(6, '제목', '내용') <class 'tuple'>
```

■ MySQL

● 데이터 조회 (컬럼별 데이터 확인)

```
import pymysql

try:
    conn = pymysql.connect(
        host='localhost', user='root', password='mysql',
        db='python', charset='utf8')

    with conn.cursor() as cursor:
        sql = '''SELECT id, title, content FROM mysql'''
        cursor.execute(sql)

        result = cursor.fetchmany(3)
        for row in result:
            print(row[0], row[1], row[2])
except pymysql.InternalError as e:
    print('InternalError', e)
except pymysql.ProgrammingError as e:
    print('ProgrammingError', e)
finally:
    conn.close()
```

Array Based (Tuple)

- 1 제목수정 내용
- 3 제목 내용
- 4 제목 내용

■ MySQL

● 데이터 조회 (컬럼별 데이터 확인)

```
import pymysql

try:
    conn = pymysql.connect(
        host='localhost', user='root', password='mysql',
        db='python', charset='utf8')

    with conn.cursor(pymysql.cursors.DictCursor) as cursor:
        sql = '''SELECT id, title, content FROM mysql'''
        cursor.execute(sql)

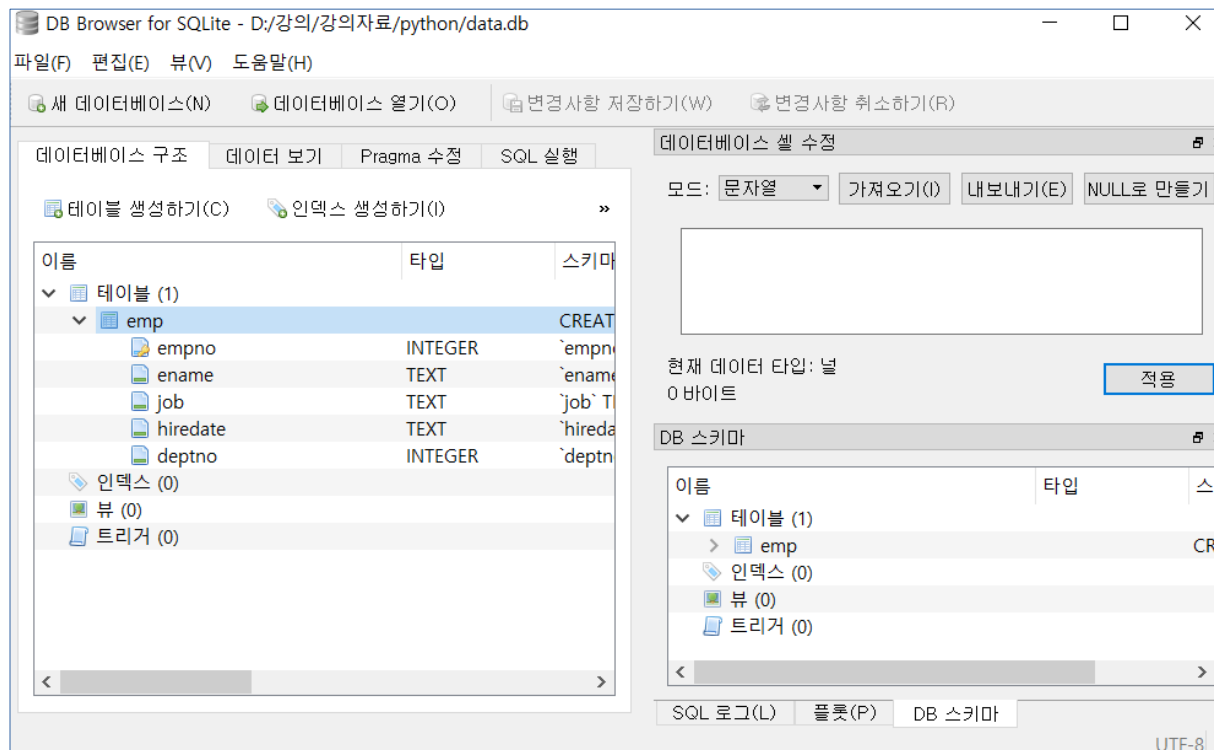
        result = cursor.fetchmany(3)
        for row in result:
            print(row['id'], row['title'], row['content'])
except pymysql.InternalError as e:
    print('InternalError', e)
except pymysql.ProgrammingError as e:
    print('ProgrammingError', e)
finally:
    conn.close()
```

Dictionary Based (Dictionary)

- 1 제목수정 내용
- 3 제목 내용
- 4 제목 내용

■ SQLite

- 별도의 DB서버 없이 파일을 이용하여 데이터 처리 (Embedded SQL DB 엔진)
- Mac OS / Linux 에서는 기본 내장
- 다운로드 / 설치
 - Command Line : <http://www.sqlite.org>
 - GUI : <https://sqlitebrowser.org> (DB Browser for SQLite)

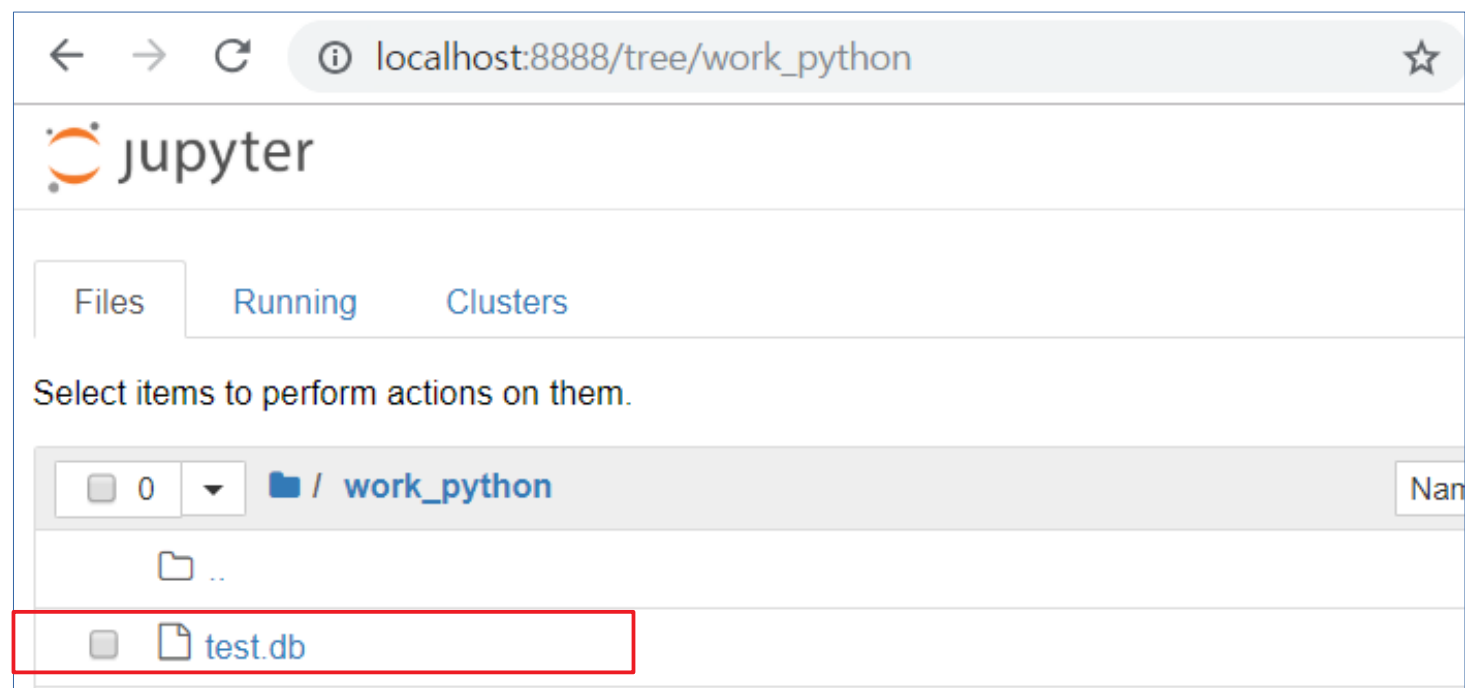


■ SQLite

● 데이터베이스 파일 생성

```
import sqlite3
```

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



■ SQLite

● 테이블 생성

```
import sqlite3

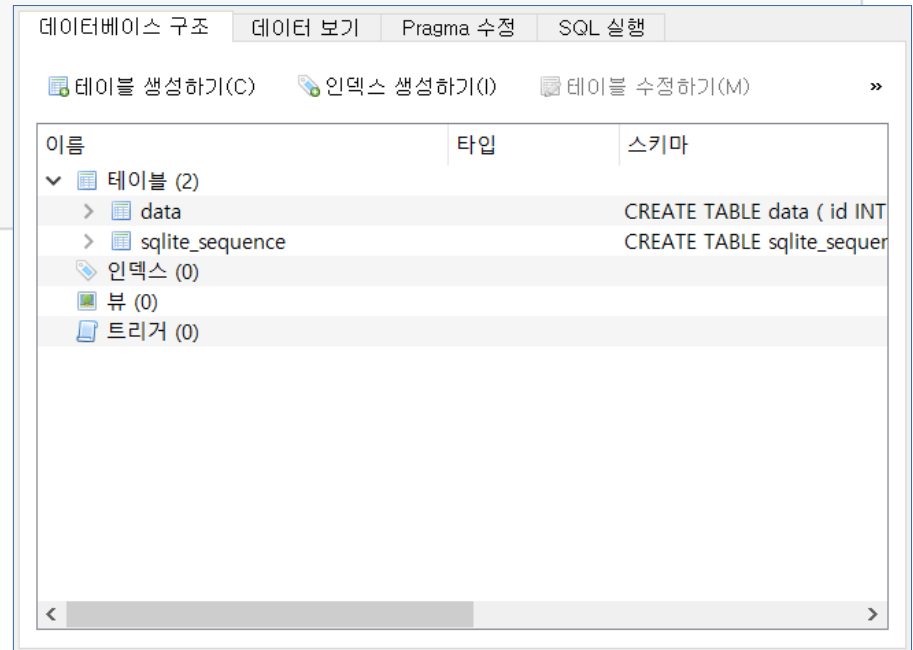
conn = sqlite3.connect('test.db')

cursor = conn.cursor()

sql = '''CREATE TABLE data (
        id INTEGER PRIMARY KEY AUTOINCREMENT
        , title VARCHAR(100), content VARCHAR(100))'''
cursor.execute(sql)

conn.commit()

cursor.close()
conn.close()
```



■ SQLite

● 데이터 입력

```
import sqlite3

conn = sqlite3.connect('test.db')

cursor = conn.cursor()

sql = '''INSERT INTO data (id, title, content) VALUES (null, ?, ?)'''
cursor.execute(sql, ('제목', '내용'))

conn.commit()

cursor.close()
conn.close()
```

The screenshot shows a SQLite database viewer interface. At the top, there are tabs for '데이터베이스 구조' (Database Structure), '데이터 보기' (View Data), 'Pragma 수정' (Modify Pragma), and 'SQL 실행' (Execute SQL). The '데이터 보기' tab is selected. Below the tabs, there is a dropdown menu for '테이블:' (Table:) with 'data' selected. To the right of the dropdown are icons for refreshing and saving, and buttons for '새 레코드' (New Record) and '레코드 삭제' (Delete Record). The main area displays a table with three columns: 'id', 'title', and 'content'. The first row has the values '1', '제목', and '내용'. At the bottom, there are navigation buttons (first, previous, next, last) and a text input field labeled '특정 레코드 행으로 가기:' (Go to specific record row:) with the value '1' entered.

id	title	content
1	제목	내용

■ SQLite

● 데이터 수정

```
import sqlite3

conn = sqlite3.connect('test.db')

cursor = conn.cursor()

sql = '''UPDATE data SET title = ? WHERE id = ?'''
cursor.execute(sql, ('제목수정', 1))

conn.commit()

cursor.close()
conn.close()
```

The screenshot shows a SQLite database viewer window with the following components:

- Tab bar: 데이터베이스 구조, 데이터 보기 (selected), Pragma 수정, SQL 실행
- Database: data
- Buttons: 새 레코드, 레코드 삭제
- Table structure and data:

	id	title	content
필터	필터	필터	
1	1	제목수정	내용

At the bottom, there are navigation controls: |< < 1 - 1 of 1 > >| and a field "특정 레코드 행으로 가기:" with the value "1".

■ SQLite

● 데이터 삭제

```
import sqlite3

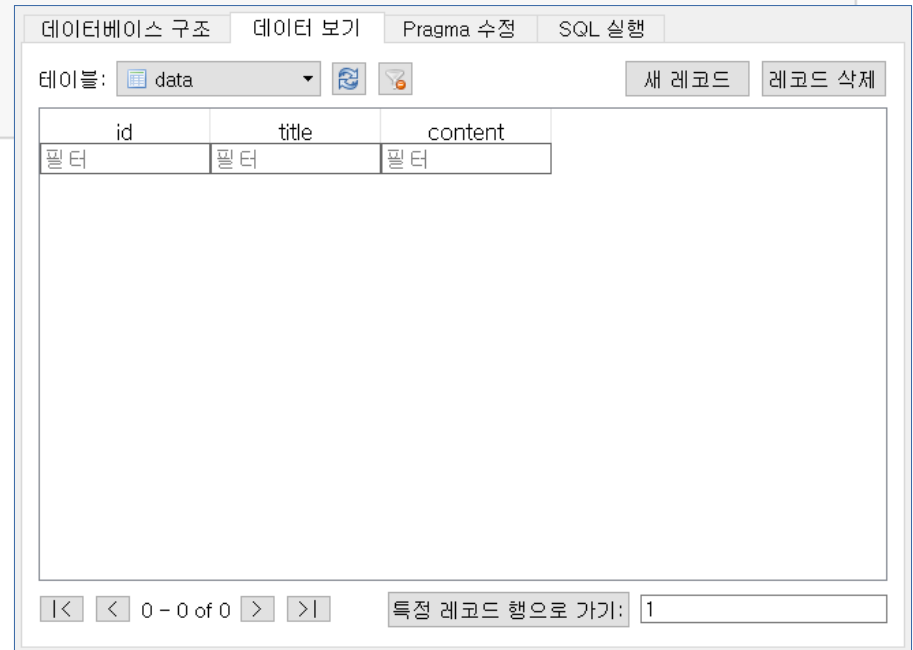
conn = sqlite3.connect('test.db')

cursor = conn.cursor()

sql = '''DELETE FROM data WHERE id = ?'''
cursor.execute(sql, (1,))

conn.commit()

cursor.close()
conn.close()
```



■ SQLite

● 데이터 조회

```
import sqlite3

conn = sqlite3.connect('test.db')

cursor = conn.cursor()

sql = '''SELECT id, title, content FROM data'''
cursor.execute(sql)

result = cursor.fetchall() # fetchone() / fetchmany()
for row in result:
    print(row, type(row))

conn.commit()

cursor.close()
conn.close()
```

```
(2, '제목', '내용') <class 'tuple'>
(3, '제목', '내용') <class 'tuple'>
(4, '제목', '내용') <class 'tuple'>
(5, '제목', '내용') <class 'tuple'>
(6, '제목', '내용') <class 'tuple'>
(7, '제목', '내용') <class 'tuple'>
```

■ SQLite

● 데이터 조회 (컬럼별 데이터 확인)

```
import sqlite3

conn = sqlite3.connect('test.db')

cursor = conn.cursor()

sql = '''SELECT id, title, content FROM data'''
cursor.execute(sql)

result = cursor.fetchmany(3)
for row in result:
    print(row[0], row[1], row[2])

conn.commit()

cursor.close()
conn.close()
```

2 제목 내용

3 제목 내용

4 제목 내용

■ SQLite

● 데이터 조회 (컬럼별 데이터 확인)

```
import sqlite3

conn = sqlite3.connect('test.db')
conn.row_factory = sqlite3.Row

cursor = conn.cursor()

sql = '''SELECT id, title, content FROM data'''
cursor.execute(sql)

result = cursor.fetchmany(3)
print(row.keys())
for row in result:
    print(row['id'], row['title'], row['content'])

conn.commit()

cursor.close()
conn.close()
```

['id', 'title', 'content']

2 제목 내용

3 제목 내용

4 제목 내용