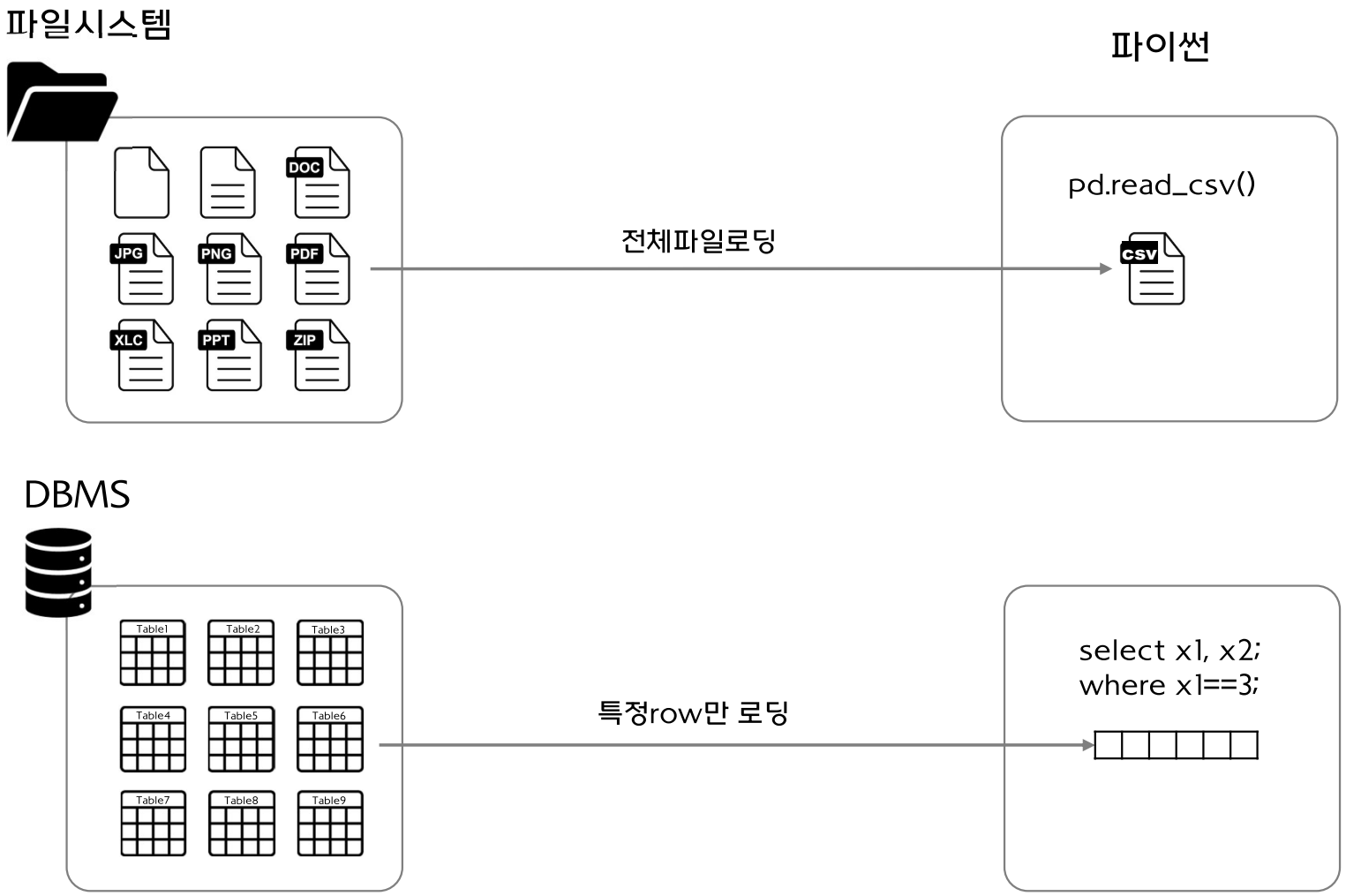
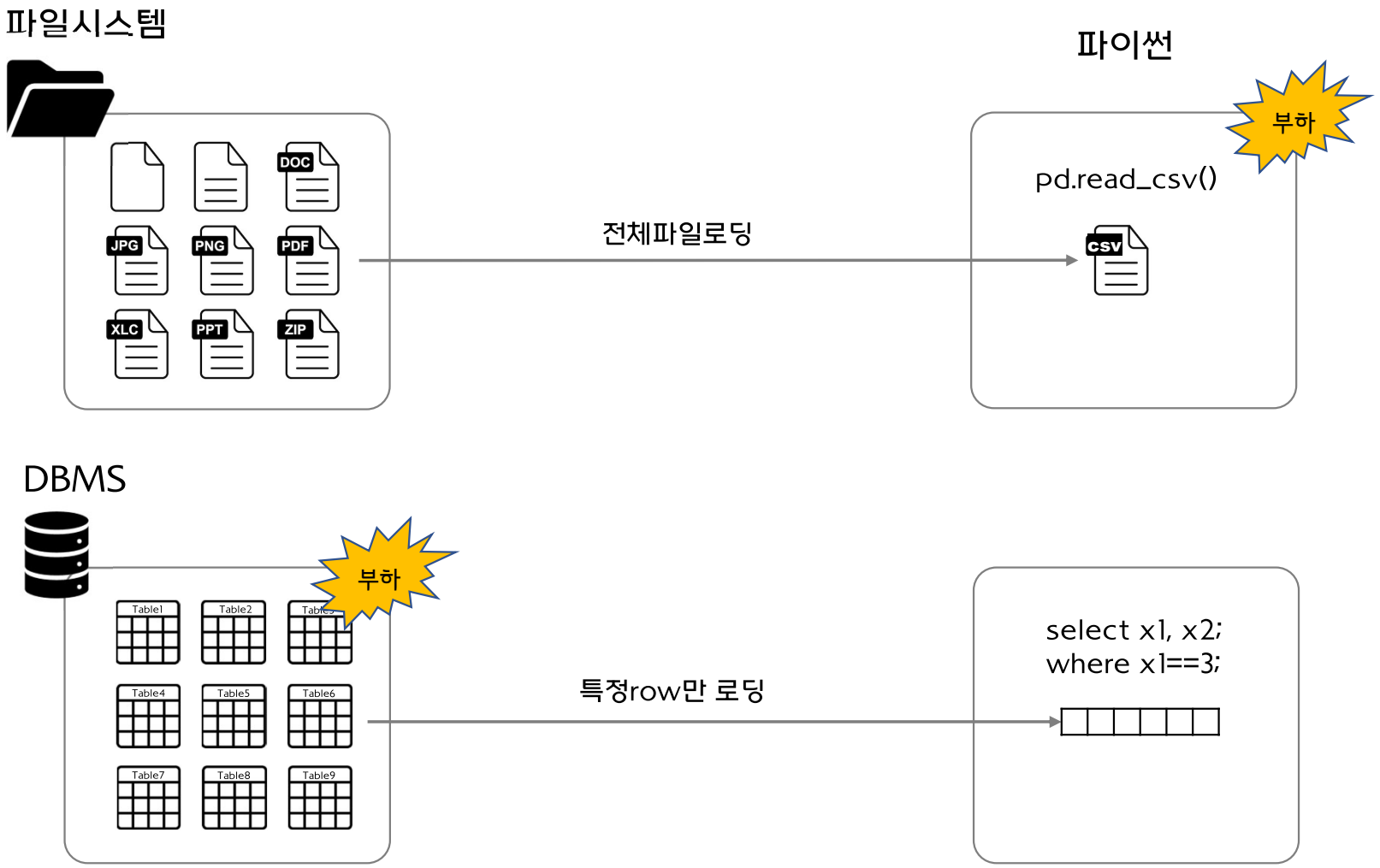
Machine Learning Operation(MLOps)

- Ch2. Level0 MLOps(2), SQLite3

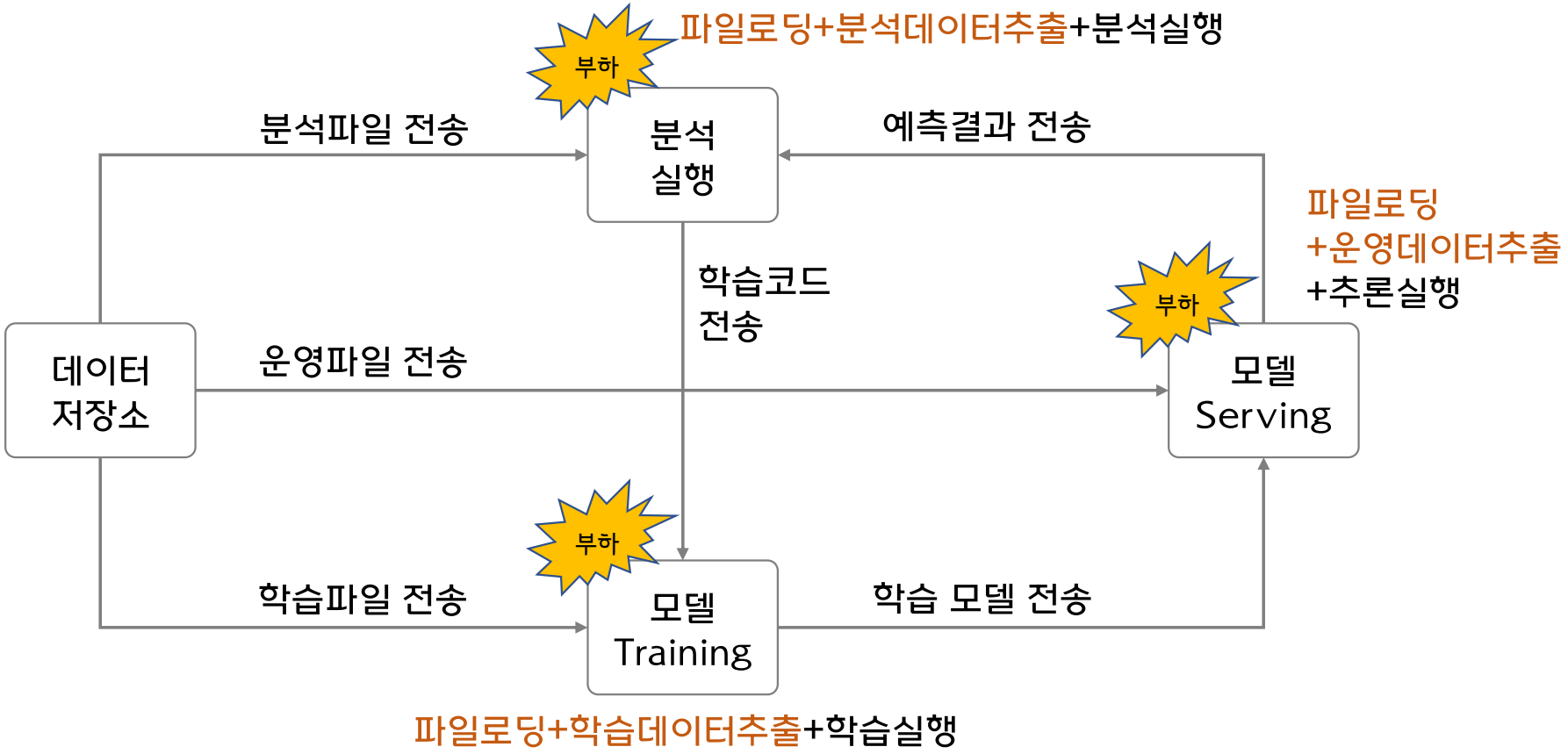
DB와 파일 사용시 차이점



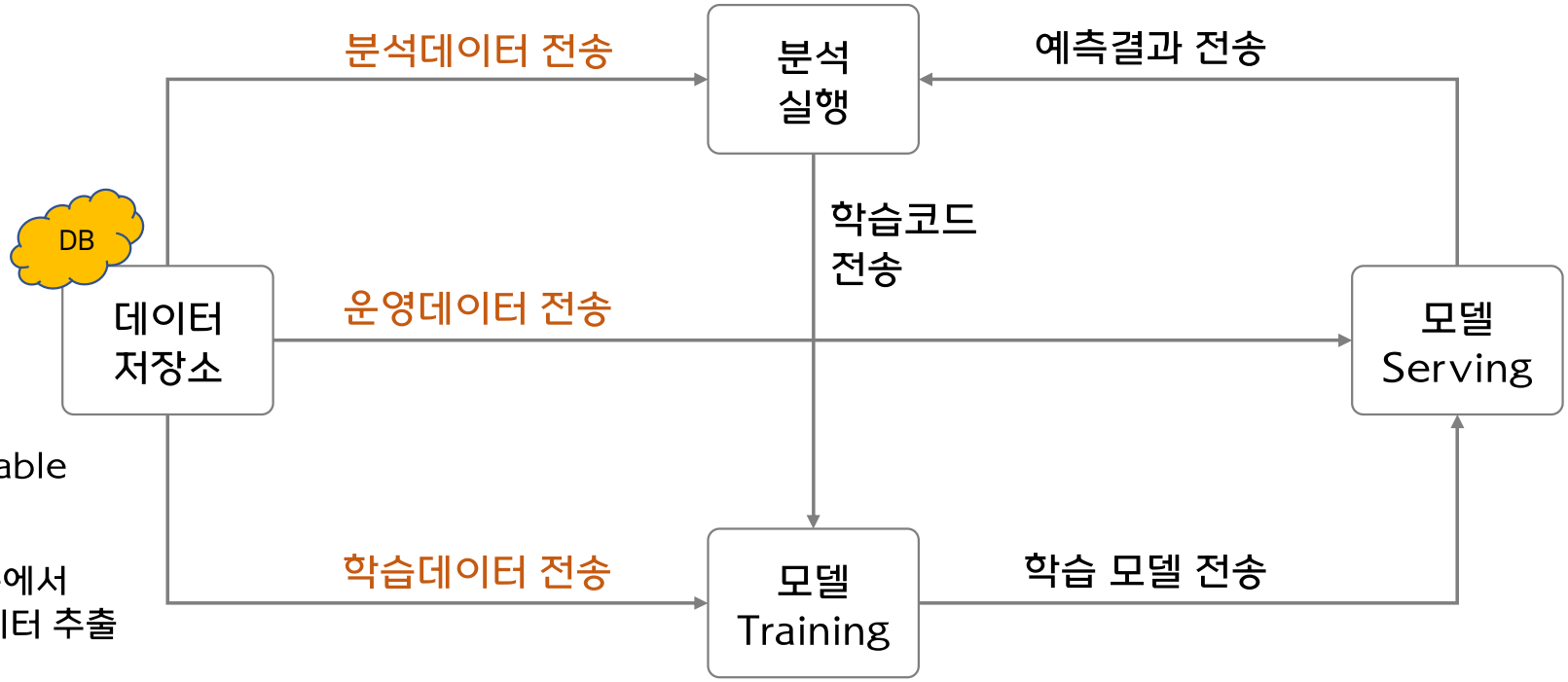
DB와 파일 사용시 차이점



역할별 환경분리

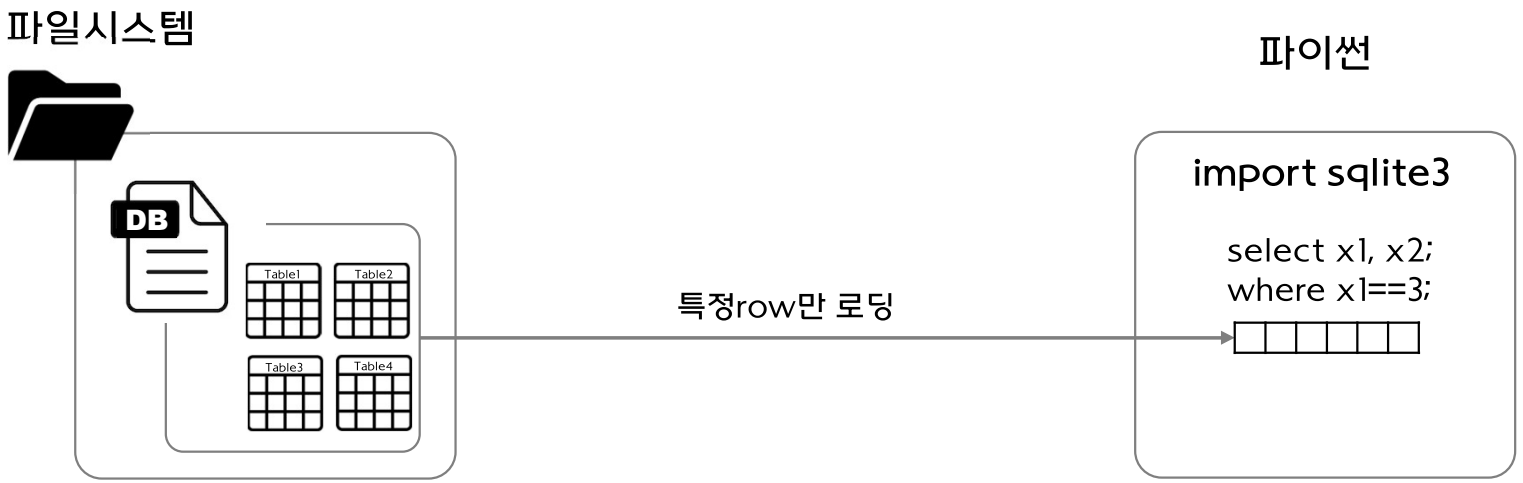

파일로딩+분석데이터추출+분석실행
부하
분석
실행
분석파일 전송
예측결과 전송
파일로딩
+운영데이터추출
+추론실행
부하
학습코드
전송
데이터
저장소
운영파일 전송
모델
Serving
부하
학습파일 전송
학습 모델 전송
모델
Training
파일로딩+학습데이터추출+학습실행

역할별 환경분리



- 용도별 table 사전생성
- 각 table에서 필요 데이터 추출

SQLite3


파일시스템
DB
Table1
Table2
Table3
Table4
특정row만 로딩
파이썬
import sqlite3
select x1, x2;
where x1==3;

SQLite3 사용법

※출처: https://www.youtube.com/watch?v=byHcYRpMgI4&t=1656s

SQLite3 기본문법

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("""  SQL명령어      """)

conn.commit()


conn.close()
```

DB 이름 (불러올 파일명)

커서 생성

SQL 명령어 입력

Commit command

DB 접속 종료

Create Table – database1.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("""CREATE TABLE test_table(
        first_name text,
        last_name text,
        email text
        )""")

# Datatypes:
# NULL
# INTEGER
# REAL
# TEXT
# BLOB

## commit our command
conn.commit()

## close our connection
conn.close()
```

1개 행 추가하기 – database2.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()
c.execute(""" INSERT INTO test_table VALUES ('John','Elder','john@korea.ac.kr') """)
# c.execute(""" INSERT INTO test_table VALUES ('Tim','Smith','tim@korea.ac.kr') """)
# c.execute(""" INSERT INTO test_table VALUES ('Mary','Brown','mary@korea.ac.kr') """)

print("Command executed sucessfully...")

## commit our command
conn.commit()

## close our connection
conn.close()
```

여러 개 행 추가하기 – databse3.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

many_customers = [
                ('Wes', 'Brown', 'wes@brown.com'),
                ('Steph', 'Kuewa', 'steph@kuewa.com'),
                ('Dan', 'Pas', 'dan@pas.com')
            ]

c.executemany("INSERT INTO test_table VALUES (?,?,?)",
many_customers)


print("Command executed successfully...")
```

```
## commit our command
conn.commit()

## close our connection
conn.close()
```

Query and Fetchall – database4.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("SELECT * FROM test_table")

print(c.fetchone())
# print(c.fetchmany(3))
# print(c.fetchall())


# print("Command executed successfully...")
## commit our command
conn.commit()

## close our connection
conn.close()
```

Format result – database5.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("SELECT * FROM test_table")

items = c.fetchall()

print("NAME " + "₩t₩tEMAIL")
print("-----" + "₩t₩t--------")
for item in items:
    print(item[0] + " " + item[1] + "₩t" + item[2])

## commit our command
conn.commit()

## close our connection
conn.close()
```

WHERE 절 – database6.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

# c.execute("SELECT rowid, * FROM test_table")
c.execute("SELECT * FROM test_table WHERE last_name = 'Elder'")
# c.execute("SELECT * FROM test_table WHERE last_name LIKE 'Br%'")
# c.execute("SELECT * FROM test_table WHERE email LIKE '%korea.ac.kr'")

items = c.fetchall()

for item in items:
    print(item)

## commit our command
conn.commit()

## close our connection
conn.close()
```

Update Records – database7.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("""UPDATE test_table SET first_name = 'Bob'
            WHERE last_name = 'Elder' """)
# c.execute("""UPDATE test_table SET first_name = 'John'
#             WHERE rowid = 1  """)
# c.execute("""UPDATE test_table SET first_name = 'Marty'
#             WHERE last_name = 'Brown' """)
# c.execute("""UPDATE test_table SET first_name = 'Mary'
#             WHERE last_name = 'Brown' """)
# c.execute("""UPDATE test_table SET first_name = 'Wes'
#             WHERE rowid = 4 """)

conn.commit()
```

```
c.execute("SELECT rowid,* FROM test_table")

items = c.fetchall()

for item in items:
    print(item)

## close our connection
conn.close()
```

Delete Records – database8.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("DELETE FROM test_table WHERE rowid = 6")
conn.commit()


c.execute("SELECT rowid, * FROM test_table")
items = c.fetchall()

for item in items:
    print(item)

## close our connection
conn.close()
```

결과 정렬하기 – database9.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

# c.execute("SELECT rowid, * FROM test_table ORDER BY rowid DESC")
c.execute("SELECT rowid, * FROM test_table ORDER BY last_name ASC")

items = c.fetchall()

for item in items:
    print(item)

## close our connection
conn.close()
```

AND / OR 문 – database10.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("SELECT rowid, * FROM test_table WHERE last_name LIKE 'Br%' ")
# c.execute("SELECT rowid, * FROM test_table WHERE last_name LIKE 'Br%' AND rowid = 3 ")
# c.execute("SELECT rowid, * FROM test_table WHERE last_name LIKE 'Br%' OR rowid = 3 ")

items = c.fetchall()

for item in items:
    print(item)

## close our connection
conn.close()
```

결과 길이 제한하기 – database11.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("SELECT rowid, * FROM test_table LIMIT 2")
# c.execute("SELECT rowid, * FROM test_table ORDER BY rowid DESC LIMIT 3 ")

items = c.fetchall()

for item in items:
    print(item)

## close our connection
conn.close()
```

Table 삭제하기 – database12.py

```
import sqlite3

db_name = "test.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

c.execute("DROP TABLE test_table")
c.commit()


c.execute("SELECT rowid, * FROM test_table")

items = c.fetchall()

for item in items:
    print(item)

## close our connection
conn.close()
```

SQL명령어 함수화#1 – show_all

app.py

```
import database

database.show_all()
```

database.py

```
import sqlite3

def show_all():
    db_name = "test.db"
    conn = sqlite3.connect(db_name)
    c = conn.cursor()

    c.execute("SELECT rowid, * from test_table")
    items = c.fetchall()

    for item in items:
        print(item)

    conn.commit()
    conn.close()
```

SQL명령어 함수화#2 – add_one

app.py

```
import database

database.add_one(“Laura”,”Smith”,”laura@smith.com” )
database.show_all()
```

database.py

```
import sqlite3

def add_one(first, last, email):
    db_name = "test.db"
    conn = sqlite3.connect(db_name)
    c = conn.cursor()

    c.execute("INSERT INTO test_table VALUES (?,?,?)", (first, last, email))

    conn.commit()
    conn.close()
```

SQL명령어 함수화#3 – delete_one

app.py

```
import database

database.delete_one("6")
database.show_all()
```

database.py

```
import sqlite3

def delete_one(id):
    db_name = "test.db"
    conn = sqlite3.connect(db_name)
    c = conn.cursor()

    c.execute("DELETE FROM test_table WHERE rowid = (?)", id)

    conn.commit()
    conn.close()
```

SQL명령어 함수화#4 – add_many

app.py

```
import database

stuff = [
    ('Branda','Smitherton','brenda@smitherton.com'),
    ('Joshua','Raintree','josh@raintree.com')
]

database.add_many(stuff)

database.show_all()
```

database.py

```
import sqlite3

def add_many(list):
    db_name = "test.db"
    conn = sqlite3.connect(db_name)
    c = conn.cursor()

    c.executemany("INSERT INTO test_table VALUES (?,?,?)", (list))

    conn.commit()
    conn.close()
```

SQL명령어 함수화#5 – lookup with where

app.py

```
import database

database.email_lookup('dan@korea.ac.kr')
database.show_all()
```

database.py

```
import sqlite3

def email_lookup(email):
    db_name = "test.db"
    conn = sqlite3.connect(db_name)
    c = conn.cursor()

    c.executemany("SELECT * FROM test_table WHERE email = (?)", (email,))

    items = c.fetchall()

    for item in items:
        print(item)

    conn.commit()
    conn.close()
```

End of Document