===========20191218

데이터베이스 처리

I. 데이터베이스 개념

II. 데이터베이스 구축

III. 데이터 입력과 조회

## I. 데이터베이스 개념

=>데이터베이스 이해

대량의 데이터가 발생하는 현대 사회에서 데이터베이스를 사용하면 수십억, 수백억 건의 데이터를 저장하고 무리 없이 운영할 수 있음

데이터베이스 소프트웨어(DBMS) - Oracle, SQL Server, MySQL, Access, SQLite 등

=>데이터베이스 기본 기념

데이터베이스 정의:

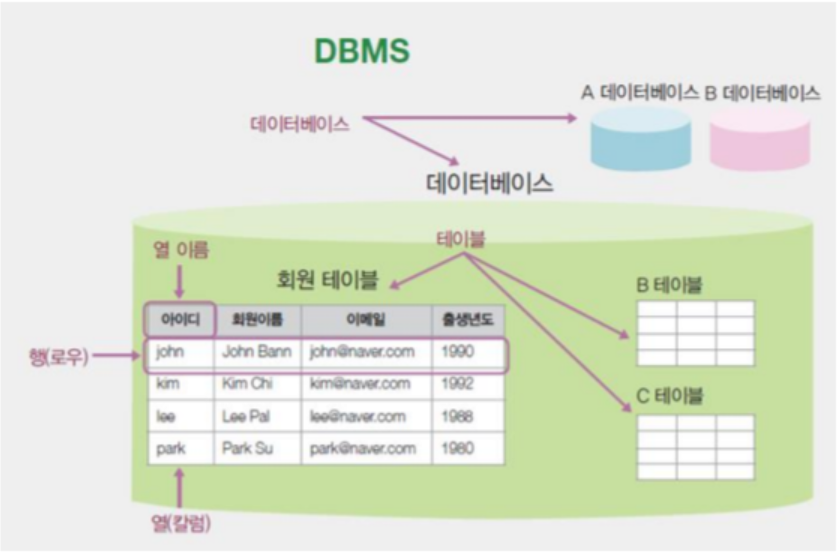
대량의 데이터를 체계적으로 저장해 여러 사용자가 서로 공유할 수 있는 시스템

관계형 데이터베이스:

DBMS는 계층형(Hierarchical), 망형(Network), 관계형(Relational), 객체지향형(Object-Oriented), 객체관계형(Object-Relational) 등의 유형으로 나눔

그 중 관계형(Relational) DBMS가 가장 많이 사용됨

=>DBMS 구성도

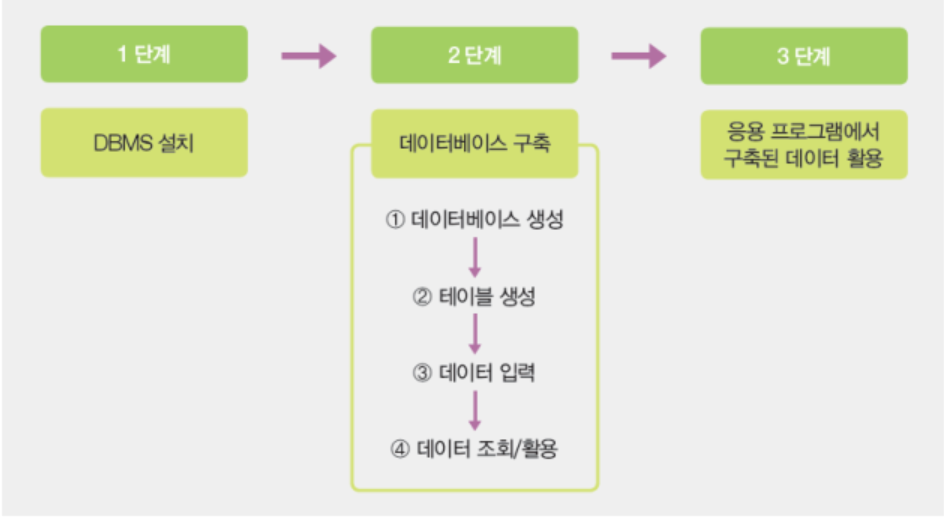


데이터베이스 객체지향으로 만들면 중복이 될수 있다.

rdb관계형 database\

## II. 데이터베이스 구축

=>데이터베이스 구축 및 운영 절차



=> 파이썬에서 데이터 입력하는 코딩 순서



## III. 데이터 입력과 조회

오라클 db 연동을 위한 모듈 다운로드 및 설치 (윈도우 키 + R)

python -m pip install cx\_Oracle --upgrade

pip install cx\_Oracle

1. 데이터베이스 연결

오라클을 사용하기 위해 관련 모듈인 cx\_Oracle를 임포트한 후, cx\_Oracle.connect(‘유저’, ‘비밀번호‘, ‘데이터베이스 서버 주소)으로 데이터베이스와 연결

1. 커서 생성

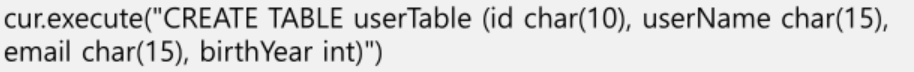
커서(Cursor)는 데이터베이스에 SQL문을 실행 또는 실행 결과를 돌려받는 통로

①에서 연결한 연결자에 커서를 만들어야 함

cur = con.cursor( )

1. 테이블 만들기

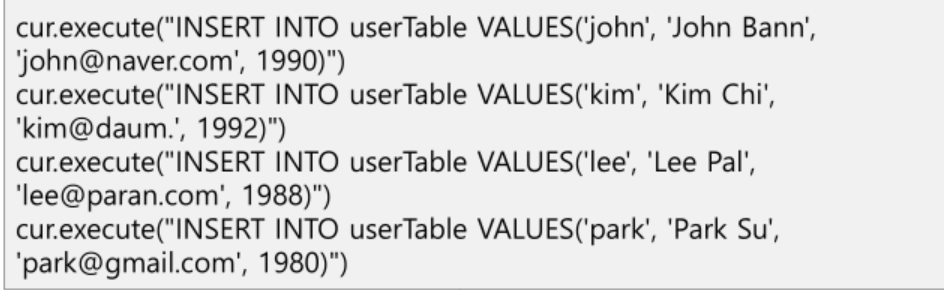
테이블을 만드는 SQL문을 커서이름.execute() 함수의 매개변수로 넘겨주면, SQL문이 데이터베이스에 실행



1. 데이터 입력

데이터 입력도 SQL문을 사용해야 하므로 커서이름.execute() 함수를 사용

데이터 입력은 필요한 만큼 반복함



1. 입력한 데이터 저장

④에서 입력한 4건의 데이터는 임시로 저장된 상태임

이를 확실하게 저장하는 것을 커밋(Commit)이라고 함

con.commit( )

1. 데이터베이스 닫기

con.close( )

=>데이터 입력 프로그램 구현



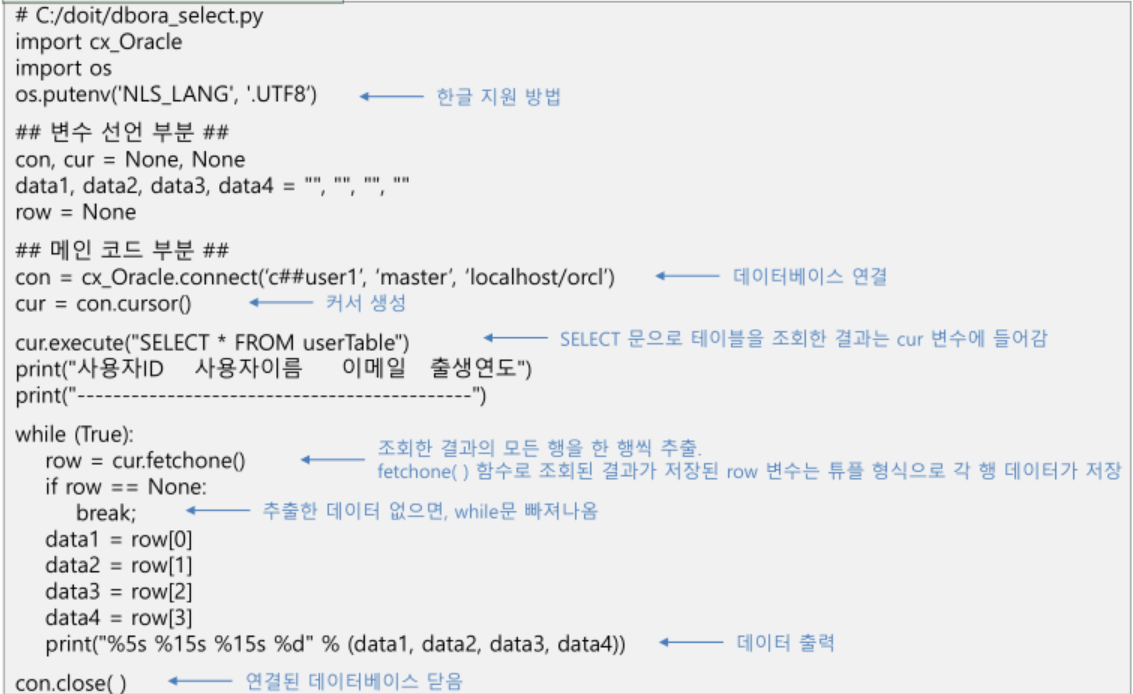
=>파이썬에서 데이터 조회하는 코딩 순서



③에서는 커서에 SELECT로 조회한 결과를 한꺼번에 저장

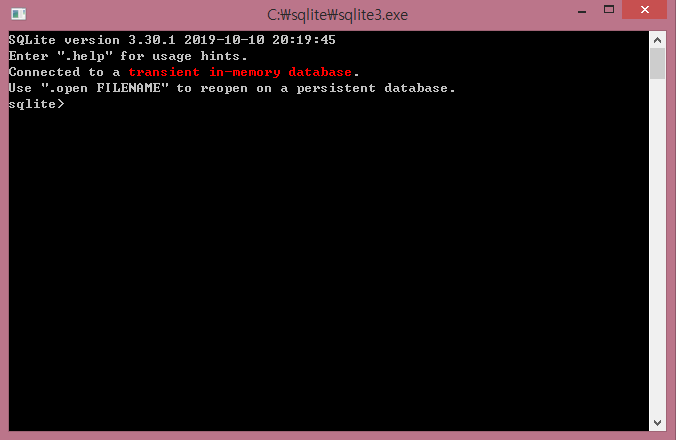
④에서 조회한 데이터를 한 행씩 fetchone( ) 함수로 접근한 다음 출력

=>데이터 조회 프로그램 구현



## 2 단계 : 데이터베이스 구축

파일 탐색기에서 sqlite3.exe를 더블 클릭하면 명령 프롬프트가 실행되고, sqlite> 로 프롬프트가 표시됨

<https://www.sqlite.org/download.html>

데이터베이스 생성



자주 사용하는 SQLite 명령어

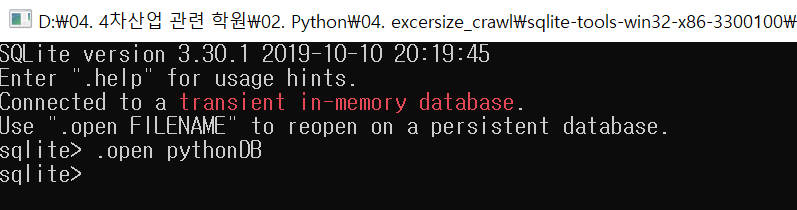
.open : 데이터베이스를 생성하거나 열어줌 데이터베이스 수정

.table : 현재 데이터베이스의 테이블 목록을 보여줌 데이터베이스 수정

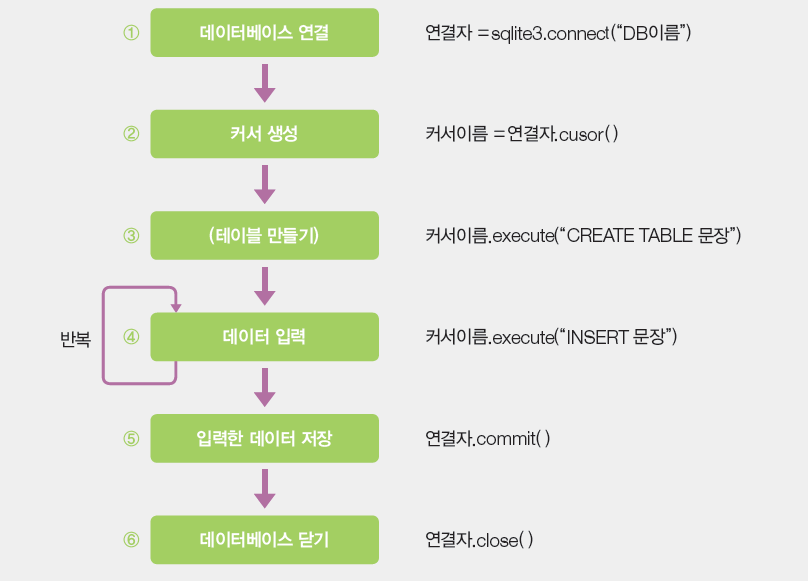
.schema 테이블이름 : 테이블의 열 및 데이터 형식 등 정보를 보여줌 schema 테이블 이름

.quit : SQLite 를 종료함

.open pythonDB



=> 파이썬에서 데이터 입력하는 코딩 순서



① 데이터베이스 연결

SQLite를 사용하기 위해 관련 모듈인 sqlite3를 임포트한 후, sqlite3.connect(“DB 이름”)으로 데이터베이스와 연결

import sqlite3

con=sqlite3.connect("C:/sqlite/pythonDB")

② 커서 생성

커서(Cursor)는 데이터베이스에 SQL문을 실행 또는 실행 결과를 돌려받는 통로

에서 연결한 연결자에 커서를 만들어야 함

cur = con.cursor( )

③ 테이블 만들기

테이블을 만드는 SQL문을 커서이름.execute() 함수의 매개변수로 넘겨주면, SQL문이 데이터베이스에 실행

cur.execute("CREATE TABLE userTable (id char(4), userName char(15), email char(15), birthYear int)")

④ 데이터 입력

데이터 입력도 SQL문을 사용해야 하므로 커서이름.execute() 함수를 사용

데이터 입력은 필요한 만큼 반복함

cur.execute("INSERT INTO userTable VALUES('john', 'John Bann', 'john@naver.com', 1990)")

cur.execute("INSERT INTO userTable VALUES('kim', 'Kim Chi', 'kim@daum.', 1992)")

cur.execute("INSERT INTO userTable VALUES('lee', 'Lee Pal', 'lee@paran.com', 1988)")

cur.execute("INSERT INTO userTable VALUES('park', 'Park Su', 'park@gmail.com', 1980)")

⑤ 입력한 데이터 저장

④에서 입력한 4건의 데이터는 임시로 저장된 상태임

이를 확실하게 저장하는 것을 커밋(Commit)이라고 함

con.commit( )

데이터베이스 닫기

con.close( )

=> 데이터 입력 프로그램 구현

# C:/doit/db\_insert.py

import sqlite3

## 변수 선언 부분 ##

con, cur = None, None

data1, data2, data3, data4 = "", "", "", ""

sql = ""

## 메인 코드 부분 ##

con = sqlite3.connect("C:/sqlite/naverDB")

cur = con.cursor()

while (True):

data1 = input("사용자ID ==> ")

if data1 == "":

break;

data2 = input("사용자이름 ==> ")

data3 = input("이메일 ==> ")

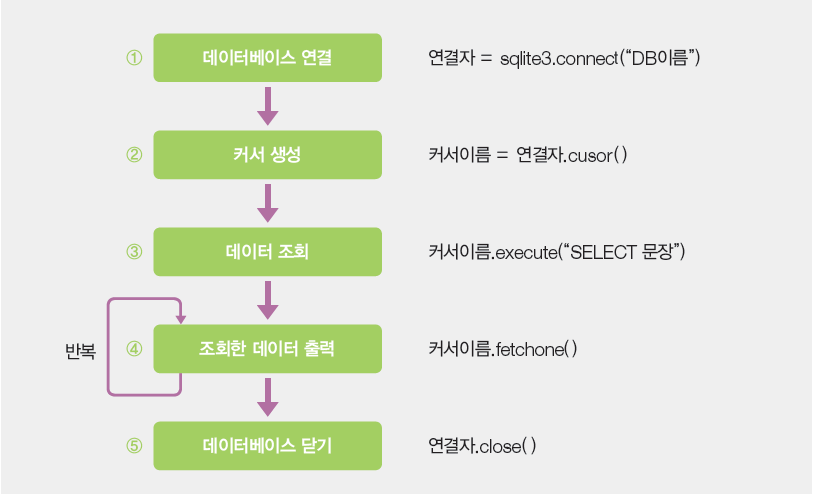
data4 = input("출생년도 ==> ")

sql = "INSERT INTO userTable VALUES('" + data1 + "', '" + data2 + "', '" + data3 + "', " + data4 + ")"

cur.execute(sql)

con.commit( )

con.close( )

파이썬에서 데이터 조회하는 코딩 순서

③에서는 커서에 SELECT로 조회한 결과를 한꺼번에 저장

④에서 조회한 데이터를 한 행씩 fetchone( ) 함수로 접근한 다음 출력

=>데이터 조회 프로그램 구현

# C:/doit/db\_select.py

import sqlite3

## 변수 선언 부분 ##

con, cur = None, None

data1, data2, data3, data4 = "", "", "", ""

row = None

## 메인 코드 부분 ##

con = sqlite3.connect("C:/sqlite/naverDB")

cur = con.cursor()

cur.execute("SELECT \* FROM userTable")

print("사용자ID 사용자이름 이메일 출생연도")

print("--------------------------------------------")

while (True):

row = cur.fetchone()

if row == None:

break;

data1 = row[0]

data2 = row[1]

data3 = row[2]

data4 = row[3]

print("%5s %15s %15s %d" % (data1, data2, data3, data4))

con.close( )