database : table로 구성(table 집합)

table; record의 집합

record: field의 집합 table의 읽기/쓰기 단위

필드 단위로 읽기쓰기 안됨 레코드 단위로 가능

열을 field 행을 record 전체를 table

CRUD creat read update delete 삽입/읽기/갱신/삭제

CRUD 전용 언어를 SQL (structure query language)

Relationship DB

RDBMS->DBMS database management system db의집합+관리기능

sql을 쓰려면 dbms만 있으면 된다

RDBMS를 많이 쓰고 O(object)RDBMS, OO(object oriented??)DBMS도 있음

ORDBMS는 미디어 데이터 관리 할 때 많이 씀. 아직 완벽하지는 않음

DBMS – 오라클 MSL?, Mysql, progress등

이 많은 것들 다 클라이언트-서버 구조

그래서 우선 서버를 설치해야 함. 서버 설정을 하고 클라이언트 툴도 필요함

서버가 있어야 되고 그 서버에 접속하는 클라이언트 등 연결 방법이 필요

ODBC – open database connectivity 데이터 베이스 연결 방법???????

이건 메모리나 뭐가 용량이 큼??????뭐지??????

그래서 모바일은 **SQLite** 씀. file기반 RDBMS. 서버가 없음 database file 관리를 일반 파일처럼 관리함. 경량화된 데이터베이스. 즐겨찾기 관리, 주소록, 통화목록 관리, 미디어 목록

안드로이드는 데이터베이스 서버 직접 접속이 안됨. 속도랑 보안 때문에.

보안이 중요한거는 인터넷 연결 안함

파이썬은 SQLite 모듈이 기본적으로 설치되어있음.

트랜젝션 – 데이터 베이스 내에서 한꺼번에 수행되어야할 일련의 연산. 전부 되거나 전부 안되거나

우리는 DBMS에 SQL을 보내고 결과만 받아옴.

DBMS가 server, 우리가 client

DB엔진은 서버형태로 구성

DBMS는 데이터베이스는 모아서 관리하는거

interface 연결방법. message전달 통로

관계형 데이터베이스에 대한 표준 인터페이스를 제공한다

데이터베이스에 연결하는 방법은 달라질 수도 있음

클라이언트 – 서버 방법이기 때문에

ODBC는 연결방법이 동일함. 단 드라이버가 설치 됐을 때

파이썬은 데이터베이스 종류와는 관계없이 최소한 지원해야 하는 기능을 정의하고 있다

SQLite3 기본으로 자체 지원. 다른 데이터베이스에 연결해서 사용하려면 해당 데이터베이스의 파이썬 인터페이스 모듈을 설치 해야함

cursor는 2차원 뭐??????????????????

영구적으로 반영ㅎ 할 떄는 커밋

마지막에는 데이터베이스 연결 닫음

SQLite는 오픈소스

라이브러리가 많아서 서버는 설치하기가 어려움.

SQLite는 설치하거나 설정할게 따로 없다

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

여기에서 windows에서 tools찾아서 다운

DDL은 data definition language

**SQL**

**-DDL (data definition language) : create, drop**

**-DML (data manipulation language) : select insert, update, delete**

**-DCL(data control language) : grant**

**-TCL(transection control language): commit, rollback**

tools 찾아서 압축 풀고 C\아래로 폴더 옮김

그리고 cmd 에서 그 폴더 찾아가서 sqlite3치면 실행

>.open test.db 현재 디렉토리에 저 파일이 있으면 오픈 없으면 생성

>.database database가 생김 확인 가능

>.table table 없어서 아무것도 안나옴. 있으면 나옴

> CREATE TABLE person(

> ID INT PRIMARY KEY NOT NULL, //각각의 tebel에는ㄴ 프라이머리 키가 있어야함.

>NAME TEXT NOT NULL,

>AGE INT NOT NULL);

primary키는 중복안되는거 유일키

NOT NULL 무조건 있어야하는거 비어있으면 안됨

>.schema table명 하면 db의 스키마? 테이블 구조 나옴

>create table

>DROP TABLE person; 데이터 베이스 지우는거

다시 테이블 만들고

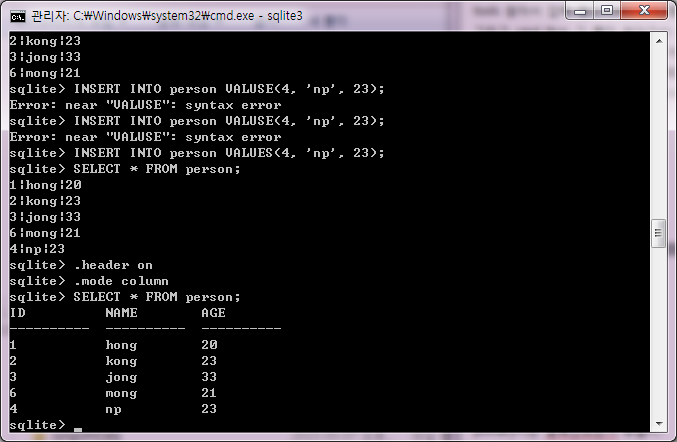
>INSERT INTO person VALUES(1,’hong’,23);

이런식으로 내용 추가

>SELECT \* FROM person; 하면 입력한 내용 볼 수 있음

>.header on

>.mode column 하고 출력하면 보기 좋게 나옴



>SELECT NAME from person;

>SELECT \* FROM person WHERE age>20;

>SELECT \* FROM person WHERE age>20 ORDER BY name; 이름 순으로

>UPDATE person SET age=33 WHERE name=’hong’ hong의 age를 33으로 바꿔라

>DELETE FROM person WHERE name=’gong

<http://sqlitebrowser.org/>

여기서

winows.exe 깔아서 실행하고 아까 만든 db 파일 열면’테이터 베이스생김 데이터 보기에서 데이터들 볼 수도 있음

sql실행에서 select id, name, age from person where age>32 이런 명령어 치고 화살표 모양으로 실행 시키면 실행 결과 볼 수도 있음

>>> import sys

>>> sys.path

['C:\\maroroWorkspace', 'C:\\Python\\Python36\\Lib\\idlelib', 'C:\\Python\\Python36\\python36.zip', 'C:\\Python\\Python36\\DLLs', 'C:\\Python\\Python36\\lib', 'C:\\Python\\Python36', 'C:\\Python\\Python36\\lib\\site-packages']

>>> import sqlite3

>>> conn = sqlite3.connect('company.db') database객체에 연결 하는게 connect

>>> conn

<sqlite3.Connection object at 0x00000000023E03B0>

이렇게하면 작업디렉토리에 company.db생성됨

>>> curs=conn.cursor() cursor객체 생성

>>> curs

<sqlite3.Cursor object at 0x0000000002F9EB90>

>>> sqlite3.sqlite\_version\_info

(3, 14, 2)

>>> curs.execute('drop table if exists employee')

<sqlite3.Cursor object at 0x0000000002F9EB90>

>>> curs.execute('create table employee(name,age)')

<sqlite3.Cursor object at 0x0000000002F9EB90>

>>> conn.commit()

이렇게하고 db browser에서 다시 열어서 보면 테이블 생성한게 보임

primary key 없어도 되긴 하는데 나중에 중복 데이터 때문에 쿼리 할 때 문제 생길 ㅅ n있고 다른 테이블이랑 join도 불가함

>>> curs.execute("insert into employee values('hong',28)") # 바깥쪽 “ ”랑 안쪽 ’ ’랑 구별해줘야함. 바껴도 상관없음

<sqlite3.Cursor object at 0x0000000002F9EB90>

>>> conn.commit()

>>> values={('kim',54),('park',34),('yun',21)}

>>> curs.executemany('insert into employee values(?,?)',values)

<sqlite3.Cursor object at 0x0000000002F9EB90>

>>> conn.commit()

튜플 하나가 레코드 하나

레코드 3개 추가한거. insert에 보면 (?,?)가 있는데 ?에 튜플에서 하나씩 들어감

>>> sql='insert into employee values(?,?)'

>>> curs.execute(sql,('honggildon',39))

<sqlite3.Cursor object at 0x0000000002F9EB90>

>>> conn.commit()

“? placeholder” 변화가 생기는데에 ?를 써서 데이터는 항상 튜플형식

>>> sql='select name, age from employee'

>>> rows=curs.execute(sql)

>>> for row in rows:

print(row)

('hong', 28)

('kim', 54)

('yun', 21)

('park', 34)

('honggildon', 39)

>>> curs.execute('select \* from employee')

<sqlite3.Cursor object at 0x0000000002F9EB90>

>>> rows=curs.fetchall()

>>> for row in rows:

print(row)

('hong', 28)

('kim', 54)

('yun', 21)

('park', 34)

('honggildon', 39)

fetchall은 execute한 결과를 다 가져오겠다는거

select는 commin을 안해도도 됨 보여주는거니깐

맨 마지막에는 close 해줘야함

>>>conn.close()

테이블 만드는 구문을 파이썬 스크립트에서 구현해줄 필요 없이 db브라우저에서 만들어서 같이 배포해도 됨. 데이터 집어 넣는 구문만 넣어줘도 되니깐

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

with conn:

cursor=conn.cursor()

cursor.execute('select \* from pets')

rows=cursor.fetchall()

for row in rows:

print(row)

with문쓰면 에러 나도 알아서 close 해줌

fetchall은 다 가져오는거

print('{:=^50}'.format(''))

cursor.execute('select \* from pets')

print(cursor.fetchone())# list형식

print('{:=^50}'.format(''))

print(cursor.fetchmany(3)) #튜플을 리스트로 묶어서

print('{:=^50}'.format(''))

print(cursor.fetchall())

cursor 객체 이용해서 DML이용

**DML - select insert, update, delete**