명령어

.exit

.help

.open (파일명: 제시안했을 때 메모리에 잡히는 데이터리스트를 쓰게 되는데 파일의 데이터를 쓰쓰 싶을 때 쓰는 명령어. 즉 파일불러오는 명령어)

.headers on/off

.mode (json,xml,column 등등)

DDL(data definition language)구조 처리 명령 CREATE / DROP / ALTER

데이터 처리 명령 DML(data management language)

C : INSERT

R : SELECT(제일 복잡하나 여러가지를 할 수 있다) / 목록(page) 단일 데이터 (key)

U : UPDate

D : DELETE

**SELECT 문의 ORDER BY 절**

SELECT column – list   
FROM table\_name  
[WHERE condition] // 대괄호는 옵션

[ORDER BY column1, column2, .. columnN] [ASC | DESC]; // default는 asc & 생략 column명 뒤에 붙는다.

Select = from contacts

Order by name

(WHERE 절이 없으면 모든 데이터처리 있으면 WHERE에 해당하는 것만 정리

Page를 위한 개념

**SELECT 문의 LIMITE 절 ( 추출)**

(Where 절에 선택된 데이터 중에서) count 개수만큼만 앞부분에서 뺴라!

SElECT column-list  
FROM table\_name  
LIMIT count [, offset] : offset +1 부터 count 개수만큼 추출  
order by가 있으면 정렬된 순서 앞에서부터 count만큼 없으면 insert 순!

TOP – N 으로 뽑아.

DELETE 문

테이블안의 데이터를 지우는 명령어 조건만 맞는다면 부분적으로 전체적으로 지울 수 있고  
DROP과는 달리 복원하는 방법도 있다.

지금까지는 성공과 실패의 기준이 단일 쿼리였다고 하면

TRANJECTION은 작업의 단위가 처음시작부터 마지막 쿼리까지이다. TRAN은 메모리상에서만 존재하면 전체가 실행되어 에러가 없을경우에만 실행된다.   
COMMIT : 전체적인 쿼리가 실행되어 성공했을 때 DB에 저장한다.

ROLLBACK : 가장 마지막에 COMMIT한 지점으로 되돌아간다. (DML만 해당한다.)

**UPDATE 문**

UPDATE table\_name

SET column1 = value1,

column2 = value2,

:

columnN=valueN

[ WHERE CONDITION ]; //조건에 만족하는 데이터만 대상으로한다.

WHERE 절이 없으면 전체 데이터가 수정됨

Id와 password

개발언어의 암호화함수를 써서 넣은것이다.

로그인시

SELECT \*

FROM CONTACTS   
WHERE EMAIL = ‘---‘ and CELL\_PHONE = ‘ ---‘

Row=cursor.fetchone() 을 통해

성공과 실패 두개로 로그인 성공과 실패를 알 수 있다.

다만 성공 실패 두개로 나뉘므로 실패 할 시 원인을 알 수 없다. Cell \_ phone 을 제거하고 이메일이 맞은 경우엔 if문을 돌려 찾는 경우가 있다. 그럴 경우엔 password는 암호화 되어있기 때문에 별도의 개발언어 암호화 함수를 사용한 경우와 데이터베이스에서 사용하는 암호화방식이 다를때 아이디또는 비밀번호가 잘못되었습니다와 비밀번호가 잘못되었습니다로 나뉘어진다.