

데이터 무결성을 위한 제약 조건

CRUD

- -----만들어라 C reate
insert
into test(userid, userpw)
values('kim', '1111');
- -----읽어라 R ead
select * from test
where userid='lee';
- -----수정해라 U pdate
update test set userpw='3333'
where userid = 'lee';
- -----삭제해라 D elete
delete from test
where userid = 'lee';

데이터 무결성을 위한 제약 조건

테이블은 컬럼과 데이터로 이루어진다.

- 컬럼(column)
- 데이터(record)
- 줄(row)
- 기본키(primary key) = pk

pk가 없으면 다른 것들을 조합해서 pk로 만들 수도 있다.

또다른 테이블에서 pk를 가져다 쓰는 것을 '참조한다'라고 표현한다. ⇒ 외래키

- 외래키(foreign key) = fk

- RDB(Relation DB) - 참조형 DB(주로 pk와 fk를 묶는다.)
- pk가 없을 수도 있는데 이건 잘못 만든 것이다.
- 현업에서 주로 관계는 8~10 많게는 30~40개.

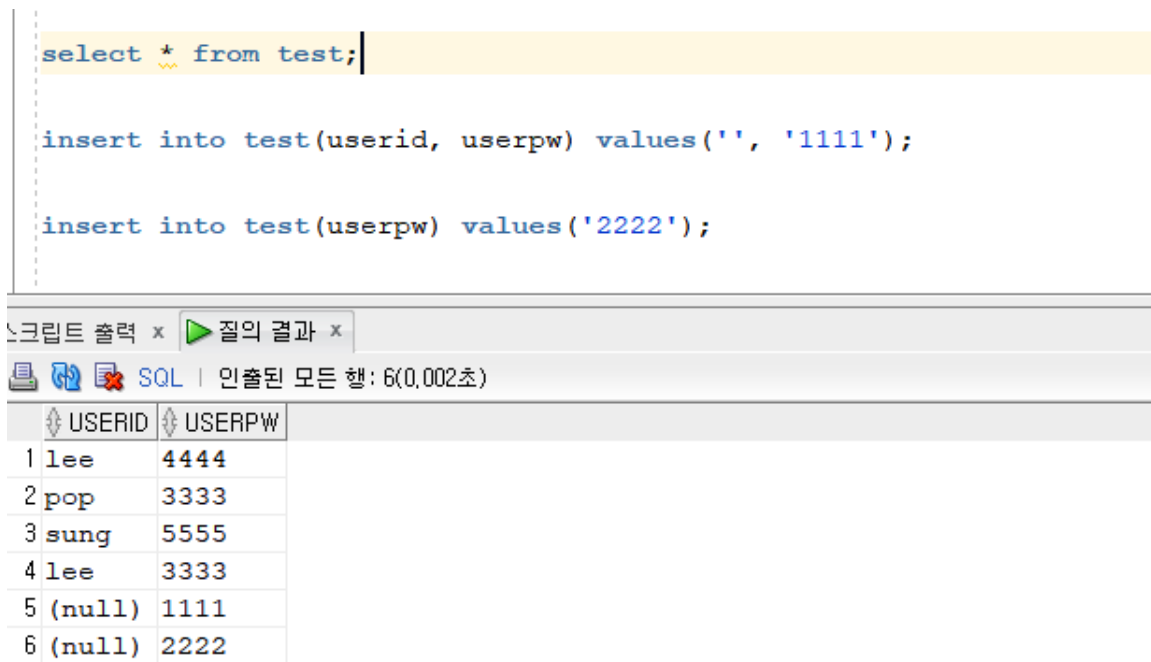
2.1 관계 데이터 모델의 개념

공식적인 용어	자주 사용되는 용어	파일 시스템 용어
릴레이션	테이블	파일
튜플	행/레코드	레코드
에트리뷰트	열	필드

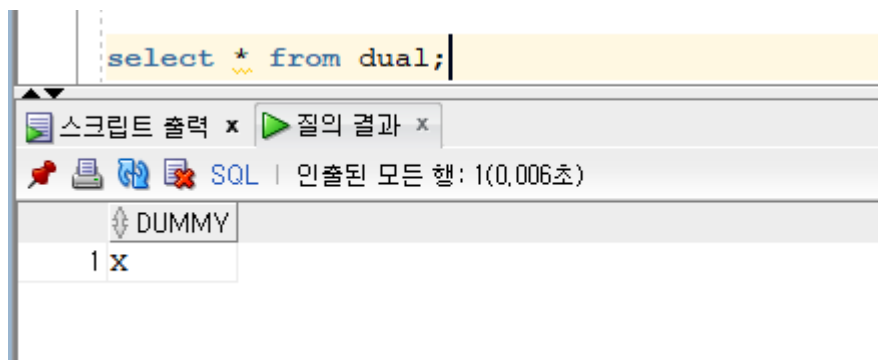
칸=컬럼=열=에트리뷰트=degree(열의 개수)

줄=레코드=행=튜플=cardinal

- 도메인: 한 에트리뷰트에 나타날 수 있는 것들(ex.스마트폰, tv)
- degree(차수): 한 테이블에 들어 있는 에트리뷰트 수
- 널값(null value): 알려지지 않음./적용할수없음을 나타내기 위해 널값을 사용.
ex) insert into test(userid, userpw) values(' ', '1111');



- 릴레이션 스키마: 릴레이션의 이름과 릴레이션의 애트리뷰트들(열)의 집합
: 릴레이션을 위한 틀(framework)



: record 딱 한 개만 갖고 있는 테이블(oracle이 자체적으로 제공해주는 내부 테이블)

```
insert into test(userid, userpw) values('lee', '1111');
select userid || userpw from test;
```

크립트 출력 x | 실행 결과 x

SQL | 인출된 모든 행: 2(0,002초)

	USERID USERPW
1	1111
2	lee1111

: 글자와 글자를 합칠 때 || 를 사용한다.

- number(10, 4) : 숫자 전체 10자리(소수점4자리 포함해서)를 받는데 소수점 4자리

- desc test;

이름	널? 유형
USERID	VARCHAR2 (20)
USERPW	VARCHAR2 (20)

- 릴레이션(테이블): 튜플(행)들의 집합

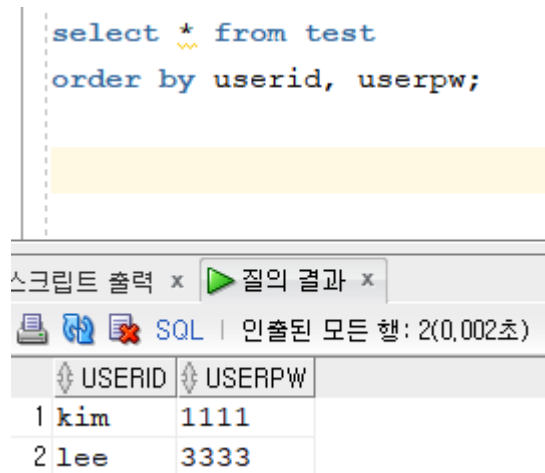
특성:

- 각 릴레이션은 오직 하나의 레코드 타입만 포함
- 한 애트리뷰트 내의 값들은 모두 같은 유형
- 애트리뷰트들의 순서는 중요하지 않음.
- 동일한 튜플이 두 개 이상 존재하지 않음.(pk가 있다는 전제하에)

정렬하는 법 - order by

- default - asc

```
select * from test
order by userid, userpw;
```



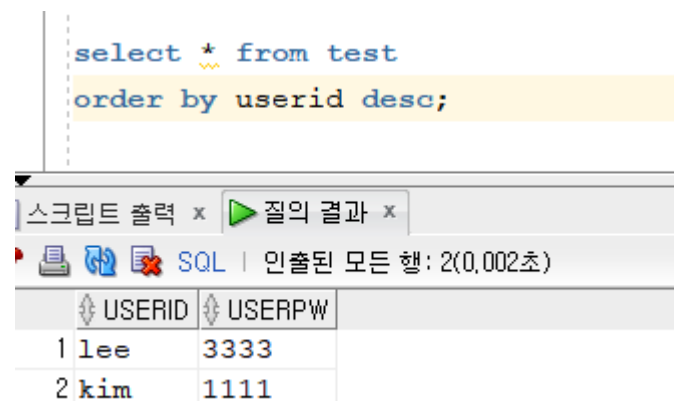
스크립트 출력 x 실행 결과 x

SQL | 인출된 모든 행: 2(0,002초)

	USERID	USERPW
1	kim	1111
2	lee	3333

- desc

```
select * from test
order by userid desc;
```



스크립트 출력 x 실행 결과 x

SQL | 인출된 모든 행: 2(0,002초)

	USERID	USERPW
1	lee	3333
2	kim	1111

데이터 무결성(data integrity)

: 데이터의 정확성 또는 유효성을 의미

: 일관된 데이터베이스 상태를 정의하는 규칙들

무결성 제약조건

: 데이터베이스 상태가 만족 시켜야 하는 조건

: 사용자에게 의한 데이터베이스 변경이 일관성을 깨지 않도록 보장하는 수단

: 데이터베이스 변경 시 DBMS가 자동으로 일관성 조건을 검사

- 기본 키와 엔티티 무결성 제약조건

1) Not null

2) Unique

- 참조 무결성 제약조건

1) 참조 무결성 제약조건은 두 릴레이션의 연관된 튜플들 사이의 일관성을 유지하는데 사용

2) 외래키(FK)는 어떤 릴레이션의 기본키(PK)이다.

create table

varchar2는 4000byte까지 허용.

long은 8000byte까지 허용.

clob은 이미지, 음향 등을 포함해 4gb까지 허용.

```
create table member(  
  mseq number primary key,  
  mno varchar2(4000),  
  yrate number(5, 2) default 10 not null  
);
```

SQL 문법

create → 테이블을 포함한 객체 등을 만들 수 있음.

insert → 데이터를 만드는 것.

drop table dept; → 객체를 없애기

delete → 데이터 자체를 없애기

update → 데이터를 수정

alter → 객체 수정

select → 객체, 데이터 뽑아오기

commit → 데이터베이스에 반영해라

rollback → 커밋 이전으로 돌아가라.

- commit, rollback은 객체 명령어에는 해당되지 않고 데이터 명령어에만 적용 가능 (insert, select, update, delete)
- DDL(create, select, alter, drop) - 데이터 정의어
- DML(insert, select, update, delete) - 데이터 조작어
- DCL(commit, rollback) - 데이터 컨트롤어

references → 그냥 쓰면 pk를 가져오게함.

references dept(dname) → 이렇게 조건을 달면 이것을 가져오게 한다.

to_date('17-12-1980', 'dd-mm-yyyy') → 문자를 날짜로 바꿔줘라.

to_date('17/12/1980', 'dd/mm/yyyy') → 문자를 날짜로 바꿔줘라.

```
select to_char(sysdate) from dual; --> 문자형으로 변경
select to_date('2020-06-15') from dual; --> 날짜형으로 변경
```

- 날짜 대표적 포맷

: YYYY-MM-DD HH24:MI:SS

- alias - column이름 바꿔서 select하기

```
select empno em from emp;
```

질의 결과 x

SQL | 인출된 모든 행: 14(0.003초)

	EM
1	7369
2	7499
3	7521
4	7566
5	7654
6	7698
7	7782
8	7788
9	7839
10	7844
11	7876
12	7900
13	7902
14	7934

- distinct - 중복 제거


```
select distinct deptno from emp;
```

의 결과 x

SQL | 인출된 모든 행: 3(0,002초)

	DEPTNO
1	30
2	20
3	10

```
select distinct job from emp;
```

의 결과 x

SQL | 인출된 모든 행: 5(0,002초)

	JOB
1	CLERK
2	SALESMAN
3	PRESIDENT
4	MANAGER
5	ANALYST

- 커미션이 null이 아닌 사원의 사번과 사원명 출력

```
--커미션이 null이 아닌 사원의 사번과 사원명 출력
select empno, ename, comm from emp
where comm is not null;
```

질의 결과 x			
SQL 인출된 모든 행: 4(0,002초)			
	EMPNO	ENAME	COMM
1	7499	ALLEN	300
2	7521	WARD	500
3	7654	MARTIN	1400
4	7844	TURNER	0

- 그룹함수

```
--그룹함수 집계 함수
select sum(comm), avg(comm) from emp;
```

질의 결과 x		
SQL 인출된 모든 행: 1(0,002초)		
	SUM(COMM)	AVG(COMM)
1	2200	550

- 컬럼이름 aliases → as

: 컬럼에 웬만하면 공백 넣으면 안된다. 절대!

```
--rename
select sum(comm) as sumcom, avg(comm) as avgsum
from emp;
```

질의 결과 x		
SQL 인출된 모든 행: 1(0,002초)		
	SUMCOM	AVGSUM
1	2200	550

desc tablename

: 테이블의 스키마 정보를 보여준다.