/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DDL(Data Definition Language): 데이터 정의어

생성 : create 종류(table, user, view, sequence)

수정 : alter 종류

제거 : drop 종류

테이블 생성

- 구문

create table 테이블 이름(

컬럼 설정

)

컬럼설정

- 컬럼명 데이터타입 [default 값] [제약조건]

- 데이터타입

- 문자열: char/nchar - 고정길이, varchar2/nvarchar2/clob - 가변길이

- 숫자: number, number(전체자릿수, 소수부자릿수)

- 날짜: date, timestamp

- default: 기본값. 값을 입력하지 않을 때 넣어줄 기본값.

제약조건 설정

- Primary key (PK): 행식별 컬럼. 기본키. NOT NULL, 유일값(Unique)

- Unique key (UK) : 유일값을 가지는 컬럼. null을 가질 수 있다.

- not null (nn) : 값이 없어서는 안되는 컬럼.

- check key (ck) : 컬럼에 들어갈 수 있는 값의 조건을 직접 설정

- foreign key (FK): 다른 테이블의 primary key컬럼의 값만 가질 수 있는 컬럼. 다른테이블을 참조할 때 사용하는 컬럼

제약조건 설정

- 컬럼 레벨 설정

- 컬럼 설정에 같이 설정

- 테이블 레벨 설정

- 컬럼 설정뒤에 따로 설정

- 기본 문법 : constraint 제약조건이름 제약조건타입

- 테이블 제약 조건 조회

- USER\_CONSTRAINTS 딕셔너리 뷰에ㅐ서 조회

테이블 삭제

- 구분

DROP TABLE 테이블이름 [CASCADE CONSTRAINTS] : 부모테이블을 삭제할 시 FK로 가지고 있는 자식테이블들의 foreign key 제약조건을 삭제한다.

데이터가 실제로 날아가진 않는다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

drop table parent\_tb cascade constraints;

select \* from user\_constraints where table\_name = 'CHILD\_TB';

--컬럼레벨 제약조건 설정

create table parent\_tb(

no number constraint parent\_tb\_pk primary key,

name varchar2(50) not null, --not null은 컬럼 레벨로 설정.

bday date default sysdate, --기본값: sysdate, nullable 컬럼.

email varchar2(100) constraint parent\_email\_uk unique,

gender char(1) not null constraint parent\_gender\_ck check(gender in ('M','F'))

);

insert into parent\_tb (no, name, email, gender) values (100, '이름1', 'a@a.com', 'M');

insert into parent\_tb (no, name, bday, email, gender) values (101, '이름2', null, 'b@b.com', 'F');

insert into parent\_tb (no, name, bday, email, gender) values (102, '이름2', null, 'b@b.com', 'F'); --email은 uk

insert into parent\_tb (no, name, bday, email, gender) values (103, '이름2', null, 'c@b.com', 'f'); --gender: check제약 대문자 F나 M만 가능

select to\_char(bday, 'hh24:mi:ss') from parent\_tb;

select \* from user\_tables; -- 테이블 정보 조회

desc PARENT\_TB; --컬럼 정보

select \* from user\_constraints -- 모든 제약조건 정보 조회

where table\_name = 'PARENT\_TB';

--테이블 레벨의 제약조건 설정.

drop table child\_tb;

create table child\_tb(

no number, --pk

jumin\_num char(14), --uk

age number not null, --ck(10~90)

parent\_no number, --fk

constraint child\_pk primary key(no),

constraint child\_jumin\_uk unique(jumin\_num),

constraint child\_age\_ck check(age between 10 and 90),

--constraint child\_parent\_fk foreign key(parent\_no) references parent\_tb --이때 parent\_tb(컬럼명) 괄호를 생략시 PK를 참조함

--constraint child\_parent\_fk foreign key(parent\_no) references parent\_tb on delete set null

--on delete set null: 부모 테이블의 참조 행이 삭제되면 자식 테이블에서 해당 행의 컬럼값은 null로 바꾼다.

constraint child\_parent\_fk foreign key(parent\_no) references parent\_tb on delete cascade

--on delete cascade: 부모 테이블의 참조 행이 삭제 되면 자식의 행도 같이 삭제.

);

insert into child\_tb values (100, '960428-2222222', 30, 100);

insert into child\_tb values (101, null, 30, 100);

insert into child\_tb values (102, null, 30, 100); --UK(unique)는 null은 몇개든 넣을 수 있다.(알 수 없는 값이기 때문)

insert into child\_tb values (103, '960428-2222222', 5, 100); --age: 10~90사이의 값만 가능

insert into child\_tb values (104, null, 30, 200); --FK: 200은 parent\_tb에 없는 값

select \* from parent\_tb;

select \* from child\_tb;

delete from parent\_tb where no = 100;

drop table publisher cascade constraints;

create table publisher(

publisher\_no number not null,

publisher\_name varchar2(50) not null,

publisher\_address varchar2(100),

publisher\_tel varchar2(20) not null ,

constraint publisher\_pk primary key(publisher\_no)

);

drop table publisher cascade constraints;

create table book(

isbn varchar2(13), --primary key는 not null을 줄 필요 없다.

title varchar2(50) not null,

author varchar2(50) not null,

page number(4) not null,

price number(8) not null,

publish\_date date default sysdate not null,

publisher\_no number not null,

constraint book\_pk primary key(isbn),

constraint book\_page\_ck check(page>=0),

constraint book\_price\_ck check(price>=0),

constraint book\_publisher\_fk foreign key(publisher\_no) references publisher

);

select \* from user\_constraints

where table\_name in ('BOOK','PUBLISHER');

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ALTER : 테이블 수정

컬럼 관련 수정

- 컬럼 추가

ALTER TABLE 테이블이름 ADD (추가할 컬럼설정 [, 추가할 컬럼설정])

- 하나의 컬럼만 추가할 경우 ( ) 는 생략가능

- 컬럼 수정

ALTER TABLE 테이블이름 MODIFY (수정할컬럼명 변경설정 [, 수정할컬럼명 변경설정])

- 하나의 컬럼만 수정할 경우 ( )는 생략 가능

- 숫자/문자열 컬럼은 크기를 늘릴 수 있다.

- 크기를 줄일 수 있는 경우 : 열에 값이 없거나 모든 값이 줄이려는 크기보다 작은 경우

- 데이터가 모두 NULL이면 데이터타입을 변경할 수 있다. (단 CHAR<->VARCHAR2 는 가능.)

- 컬럼 삭제

ALTER TABLE 테이블이름 DROP COLUMN 컬럼이름 [CASCADE CONSTRAINTS]

- CASCADE CONSTRAINTS : 삭제하는 컬럼이 Primary Key인 경우 그 컬럼을 참조하는 다른 테이블의 Foreign key 설정을 모두 삭제한다.

- 한번에 하나의 컬럼만 삭제 가능.

ALTER TABLE 테이블이름 SET UNUSED (컬럼명 [, ..])

ALTER TABLE 테이블이름 DROP UNUSED COLUMNS

- SET UNUSED 설정시 컬럼을 바로 삭제하지 않고 삭제 표시를 한다.

- 설정된 컬럼은 사용할 수 없으나 실제 디스크에는 저장되 있다. 그래서 속도가 빠르다.

- DROP UNUSED COLUMNS 로 SET UNUSED된 컬럼을 디스크에서 삭제한다.

- 컬럼 이름 바꾸기

ALTER TABLE 테이블이름 RENAME COLUMN 원래이름 TO 바꿀이름;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

제약 조건 관련 수정

-제약조건 추가

ALTER TABLE 테이블명 ADD CONSTRAINT 제약조건 설정

- 제약조건 삭제

ALTER TABLE 테이블명 DROP CONSTRAINT 제약조건이름

PRIMARY KEY 제거: ALTER TABLE 테이블명 DROP PRIMARY KEY [CASCADE]

- CASECADE : 제거하는 Primary Key를 Foreign key 가진 다른 테이블의 Foreign key 설정을 모두 삭제한다.

- NOT NULL <-> NULL 변환은 컬럼 수정을 통해 한다.

- ALTER TABLE 테이블명 MODIFY (컬럼명 NOT NULL), - ALTER TABLE 테이블명 MODIFY (컬럼명 NULL)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

--customers 카피 cust(not null을 제외한 제약 조건은 copy가 안됨)

create table cust

as

select \* from customers where 1=0; --데이터값은 가져오지 않고 컬럼만 복사해옴

--orders -> order

create table ord

as

select \* from orders where 1=0;

desc cust;

select \* from user\_constraints where table\_name = 'CUST';

--제약조건

--cust: pk

alter table cust add constraint cust\_pk primary key(cust\_id);

alter table ord add constraint ord\_cust\_fk foreign key(cust\_id) references cust;

--컬럼 수정

-- 컬럼 추가

alter table cust add (age number default 0 not null);

desc cust;

-- 수정

alter table cust modify (cust\_name null, address null, postal\_code null, phone\_number not null);

desc cust;

alter table cust modify (cust\_name varchar2(100));

-- 컬럼 이름 변경

alter table cust rename column age to cust\_age;

--컬럼 삭제

alter table cust drop column cust\_age, gender; --한번에 두개의 컬럼을 삭제 할 수 없다.

alter table cust drop column cust\_age;

--제약조건 삭제

alter table cust drop primary key cascade; --primary key는 제약조건 이름을 알 필요 없이 primary key라고 쓰면 된다. 참조하는 것도 모두 지운다.

--TODO: emp 테이블을 카피해서 emp2를 생성

drop table emp2;

create table emp2

as

select \* from emp where 1=0;

--TODO: gender 컬럼을 추가: type char(1)

desc emp2;

alter table emp2 add(gender char(1));

--TODO: email 컬럼 추가. type: varchar2(100), not null 컬럼

alter table emp2 add (email varchar2(100) not null);

--TODO: jumin\_num(주민번호) 컬럼을 추가. type: char2(14), null 허용. 유일한 값을 가지는 컬럼.

alter table emp2 add (jumin\_num varchar2(14) constraint emp2\_juminnum\_uk unique);

--TODO: emp\_id 를 primary key 로 변경

alter table emp2 add constraint emp2\_pk primary key(emp\_id);

select \* from user\_constraints where table\_name = 'EMP2';

--TODO: gender 컬럼의 M, F 저장하도록 제약조건 추가

alter table emp2 add constraint emp2\_gender\_ck check(gender in ('F','M'));

--TODO: salary 컬럼에 0이상의 값들만 들어가도록 제약조건 추가

alter table emp2 add constraint emp2\_salary\_ck check(salary>0);

--TODO: email 컬럼을 null을 가질 수 있되 다른 행과 같은 값을 가지지 못하도록 제약 조건 변경

alter table emp2 modify (email null);

alter table emp2 ADD (constraint emp2\_email\_uk unique(email));

--TODO: emp\_name 의 데이터 타입을 varchar2(100) 으로 변환

alter table emp2 modify (emp\_name varchar2(100));

desc emp2;

-- TODO: job\_id를 not null 컬럼으로 변경

alter table emp2 modify (job\_id not null);

--TODO: dept\_id를 not null 컬럼으로 변경

alter table emp2 modify(dept\_id not null);

--TODO: job\_id 를 null 허용 컬럼으로 변경

alter table emp2 modify (job\_id null);

--TODO: dept\_id 를 null 허용 컬럼으로 변경

alter table emp2 modify(dept\_id null);

--TODO: 위에서 지정한 emp\_email\_uk 제약 조건을 제거

alter table emp2 drop constraint emp2\_email\_uk;

--TODO: 위에서 지정한 emp\_salary\_ck 제약 조건을 제거

alter table emp2 drop constraint emp2\_salary\_ck;

--TODO: primary key 제약조건 제거

alter table emp2 drop primary key;

--TODO: gender 컬럼제거

alter table emp2 drop column gender;

--TODO: email 컬럼 제거

alter table emp2 drop column email;