DDL

-테이블 정의 -테이블 수정 -테이블 삭제 -제약조건

not null 이 컬럼에는 공백이 있으면 안 된다.

unique 중복이 되면 안 된다.

primary key 의 제약조건 특징: unique, not null

foreign key 의 제약조건: 데이터를 아무거나 입력할 수 없다.

check: 컬럼에 입력되는 값에 대해서 제한을 둔다.

alter table member

drop column info;

info 열을 삭제하고 싶을 때

테이블에 컬럼을 추가하고 싶을 때

alter table member

add(point number,name varchar2(20),email varchar2(20));

alter table member

add(tel varchar(15), info varchar2(10));

컬럼의 데이터 타입을 변경하고 싶을 때

alter table member

modify(tel char(11));

컬럼을 삭제하고 싶을 때

alter table member

drop column info;

컬럼명을 변경하고 싶을 때

alter table member

rename column addr to address;

테이블 삭제하고 싶을 때

drop table member;

제약조건을 추가해 member 테이블 만들기

create table member(

id varchar2(10) primary key,

pass varchar2(10),

name varchar2(20) not null,

addr varchar2(20));

insert into member (id,name) values('jang',null);

ORA-01400: cannot insert NULL into ("SCOTT"."MEMBER"."NAME")

이름은 null이 있으면 안 된다고 명시해 놓았기에 name컬럼에 null을 집어 넣으면 에러가 뜬다.

insert into member(id,name,ssn) values('jang','장동건','12345');

insert into member(id,name,ssn) values('jang','장동하','13579');

ERROR at line 1:

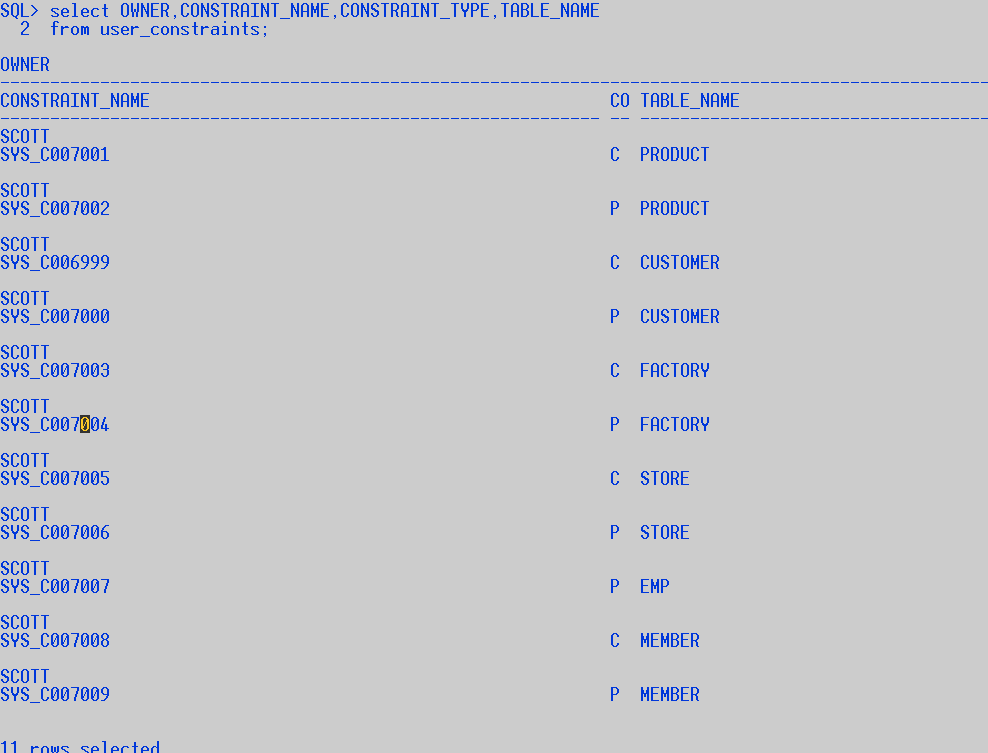
ORA-00001: unique constraint (SCOTT.SYS\_C007009) violated

이렇게 되면 기본키의 unique 제약조건을 위배하기에 에러가 뜬다.

alter table member

add constraint mem\_ssn\_uni unique(ssn);

사용자가 추가한 제약조건을 검색하고 싶을 때 사용한다



alter table member

add(ssn varchar2(15));

마지막 문제 10년으로 계산할 것

mydept 테이블 생성

create table mydept(

code varchar2(20) primary key,

name varchar2(20));

데이터 입력

insert into mydept values('001','전산실');

insert into mydept values('002','인사팀');

저장하기

commit;

member테이블에 deptcode란 컬럼 추가하기

alter table member

add(deptcode varchar2(20));

member테이블에 deptcode 컬럼에 외래키 제약조건 추가하기

alter table member

add constraint mem\_dcode\_fk foreign key(deptcode)

references mydept(code);

제약조건 잘 들어갔는지 아래의 명령어로 확인하기

select OWNER,CONSTRAINT\_NAME,CONSTRAINT\_TYPE,TABLE\_NAME

from user\_constraints;

멤버 테이블에 값 입력하기

insert into member(id,name,ssn,deptcode) values('lee','이민호','54321','001');

insert into member(id,name,ssn,deptcode) values('lee2','이민호','543210','003');

ERROR at line 1:

ORA-02291: integrity constraint (SCOTT.MEM\_DCODE\_FK) violated - parent key not found

자주 보게 될 에러이다. 부모 키에 없는 값을 입력했을 때 나오는 문제이다.

부모 키에 없는 값을 입력했구나 인식하고 부모키에 있는 값을 보고 수정할 것

1.

drop table mydept;

ERROR at line 1:

ORA-02449: unique/primary keys in table referenced by foreign keys

기본키가 설정된 테이블을 부모 테이블을

외래키가 설정된 테이블을 자식 테이블이라고 한다.

외래키를 먼저 삭제하고 작업

제약조건을 먼저 삭제하고 작업

제약조건을 먼저 제거하고 작업하는 것이 가장 좋은 방법이다.

rollback은 DML에 처리한다.

값에 대한 제약조건 설정하기

alter table member

add constraint member\_ck check(addr in('인천','서울','경기'));

값 입력하기

insert into member(id,name,addr) values('kang','강감찬','인천');

값을 입력할 때 내가 지정한 이외의 값이 왔기에 에러가 난다.

insert into member(id,name,addr) values('kang2','강감찬2','부산');

ERROR at line 1:

ORA-02290: check constraint (SCOTT.MEMBER\_CK) violated

제약조건 제거하기

alter table member

drop constraint member\_ck;

not null: 공백 있으면 안 된다.

unique: 중복 안 된다.

check: 내가 지정한 값만 와야 한다.

primary key: 중복 공백 x

foreign key: 값을 참조해서 쓰기에 그(기본 키) 테이블에 없는 값은 사용할 수 없다.

시퀀스: 하나하나 순차적으로 증가해야 하는 컬럼의 값

기본키로 만들어 주고 싶을 때

20191223||100000

주문 목록 테이블 만들기

create table myorder(

ord\_num varchar2(10) primary key,

id varchar2(20));

주문 목록 상세 테이블 만들기

create table order\_detail(

ord\_num varchar2(10),

prd\_num varchar2(20));

create sequence myorder\_seq;

주문 목록에서 주문 번호는 주문이 들어올 때마다 새롭게 증가하지만, 주문 상세 목록의 주문 번호는 현재 발생된 주문 번호가 들어가야 한다. 그러므로 주문 목록에는 myorder\_seq.nextval이지만 주문 상세 목록에는

myorder\_seq.currval이 들어가야 한다.

insert into myorder values(myorder\_seq.nextval,'jang');

insert into order\_detail values(myorder\_seq.currval,'prd001');

insert into order\_detail values(myorder\_seq.currval,'prd002');

insert into order\_detail values(myorder\_seq.currval,'prd003');

주문 목록과 주문 상세 목록 테이블 데이터 확인하기

select \* from myorder;

select \* from order\_detail;

insert into myorder values(myorder\_seq.nextval,'jang');

insert into myorder values(myorder\_seq.nextval,'jang');

insert into myorder values(myorder\_seq.nextval,'jang');

위와 같이 3개의 주문 목륵을 만들고 테이블 데이터를 확인하면

select \* from myorder;

ORD\_NUM ID

-------------------- ----------------------------------------

1 jang

2 jang

3 jang

4 jang

아래와 같이 나온다.

select myorder\_seq.currval from dual;

currval

4

sequence는 중간에 데이터를 삭제했을 때 그 삭제된 번호부터 오는게 아니라 다음 번호부터 온다.

sequence삭제하기

drop sequence myorder\_seq;

Sequence dropped.

열쇠 아이콘은 기본 키이다.

not null

unique

check

2)not null 제약조건

not null은 수정하면서 할 수 없다.

테이블을 생성하면서 해 주어야 한다.

판매 현황을 조회하기 위해서 임의로 5건 정도 입력한다.

또 고객별 판매현황을 집계하기 위해서

2개 정도의 상품을 중복해서 구매하도록 작성해 주어야 한다.

DDL은 commit안 해도 자동으로 저장이 된다.

create table TB\_MEM(

MEM\_ID varchar2(10) primary key,

MEM\_NM varchar2(20) not null,

MEM\_CD varchar2(10),

PWD varchar2(20) not null,

PWD\_HINT varchar2(90) not null,

PWD\_HINT\_ANS varchar2(90) not null,

SSN varchar2(30) not null,

BIRTH\_DT varchar2(20),

MALE\_FLG varchar2(5) not null,

TEL\_NO varchar2(20),

CELL\_NO varchar2(20),

ADDR varchar2(120) not null,

EMAIL varchar2(30),

EMAIL\_RCV\_FLG varchar2(10),

MEM\_REG\_DTM date);

alter table TB\_MEM

add constraint mem\_ssn\_uni unique(ssn);

alter table TB\_MEM

add constraint mem\_cd\_ck check(MEM\_CD in('Silver','Gold','VIP','VVIP','Guest'));

alter table TB\_MEM

modify(MEM\_REG\_DTM date);

alter table TB\_MEM

modify(MEM\_REG\_DTM varchar2(30));

제약조건 제거하기

alter table TB\_MEM

drop constraint mem\_cd\_ck;

alter table TB\_MEM

add constraint mem\_cd\_ck check(MEM\_CD in('Silver','Gold','VIP','VVIP','Guest'));

create table TB\_PRODUCT(

PRD\_NO varchar2(20) primary key,

PRD\_NM varchar2(40),

CATEGORY\_NO varchar2(20),

PRD\_DES varchar2(110),

SELL\_PRC\_UNIT number,

SELL\_START\_DT varchar2(10),

SELL\_END\_DT varchar2(10),

DLV\_CONSTRAINT varchar2(50),

AS\_FLG varchar2(5),

RETURN\_FLG varchar2(5),

IMG\_ORG\_FILE\_NM varchar2(20),

IMG\_GEN\_FILE\_NM varchar2(20),

REG\_ID varchar2(10),

REG\_DTM date);

alter table TB\_PRODUCT

modify(REG\_DTM varchar2(30));

create table TB\_ORDER(

ORD\_NO varchar2(20) primary key,

MEM\_ID varchar2(10) not null,

ORD\_AMT number not null,

RCVR\_NM varchar2(20) not null,

RCVR\_TEL\_NO varchar2(20) not null,

RCVR\_ADDR varchar2(120) not null,

DLV\_CORP\_NO varchar(40) not null,

DLV\_REQ varchar(90) not null,

ORD\_STAT\_CD varchar(20) not null,

ORD\_DTM date not null);

주문 테이블 회원아이디 외래 키 설정하기

alter table TB\_ORDER

add constraint ord\_mem\_id\_fk foreign key(mem\_id)

references TB\_MEM(mem\_id);

제약조건 잘 들어갔는지 아래의 명령어로 확인하기

select OWNER,CONSTRAINT\_NAME,CONSTRAINT\_TYPE,TABLE\_NAME

from user\_constraints;

create table TB\_ORDER\_PRODUCT(

ORD\_NO varchar2(20),

PRD\_NO varchar2(20),

QTY integer,

constraint op\_ord\_no foreign key(ORD\_NO)

references TB\_ORDER(ORD\_NO),

constraint op\_prd\_no foreign key(PRD\_NO)

references TB\_PRODUCT(PRD\_NO)

);

주문금액 = 단가 \* 수량; 으로 계산할 것

JDBC

자바 데이터베이스 커넥티비티

자바와 데이터베이스 커넥트 하는 방법

오라클은 독보적으로 관공서, 대기업에서 다 쓴다.

오라클을 하나만 쓰는 곳은 없다.

자바로 DB를 연동시키는 방법은 어려운 것처럼 느껴질 수 있다.

자바랑 DB 연동하기 위해서

프린터기에서 보면 프린터기를 연동하기 위해 드라이버를 제조사 홈페이지에서 다운 받는데 그 작업을 해준다.

C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib\ ojdbc6.jar

여기의 이 파일을 다운 받아야 한다.

jre system library

referenced libraries

ojdbc6.jar → oracle.jdbc.driver → OracleDriver.class

이 파일들은 재산이기에 decompiler로 통해 접근할 수 없다.

그래서

다음과 같은 순서로 JDBC를 적용합니다.

1. 드라이버로딩

2. 커넥션설정하기

3. Statement객체 생성하기

4. SQL문실행하기

5. 결과처리

6. 자원반납

드라이버를 로딩한다는 것은 JVM이 어 떤 DBMS를 사용하는지 인식시키는 작업으로 java에서 오라클을 연동할 수 있도록 오라클에서 제공해준 드라이버 파일을 JVM의 메모리에 로딩시키는 작업을 하는 단계

자바에서 DB로 접근할 때는 이러한 메소드를 쓰라고 자바 API안에 명시되어 있다.

핵심클래스를 오라클에 올리는 작업을 해 줘야 한다.

자바에서 쓰는 클래스를 뽑아놓은 것

class class

클래스의 정보가 이미 static 영역에 올라가 있다는 의미이다.

클래스의 정보가 올라가는 곳에 그 정보를 올린다.

static Connection getConnection(String url, String user, String password)

오라클이 인식하는 IP정보, Port정보를 뽑아 주어야 한다.

접속하면 접속할 수 있는 정보만 리턴해준다.

내부에 접속해서 작업을 수행할 수 없다.

mysql에 접속하면 mysql에 접속할 수 있는 정보가 리턴된다.

위 메소드는 리턴 타입에 부모를 설정해 놓은 메소드이다.

[oracle.jdbc.driver.T4CConnection@4d76f3f8](mailto:oracle.jdbc.driver.T4CConnection@4d76f3f8)

T4CConnection: 접속에 관한 정보를 가지고 있는 클래스 객체이다.

자바는 표준 API를 활용하여 사용한다.

statement의 주된 역할은 oracle, mysql을 실행하기 위해 존재하는 것들이다.

String sql = "insert into tb\_board values(1,'jang','연습','연습',sysdate,0)";

SQL을 실행하기 위한 객체정보:oracle.jdbc.driver.OracleStatementWrapper@34c4973

연결 실패:ORA-00001: unique constraint (SCOTT.SYS\_C007041) violated



boardnum가 unique 해야 하는데 1로 겹치기 때문에 unique 제약조건을 위배 하였다는 에러가 발생한다.