



# SQL,PL/SQL(ORACLE)

## 4-1. DDL DML TCL

## 1. DDL (Data Definition Language)

- 1) **CREATE** : 테이블, 시퀀스 등과 같은 객체를 생성할 때  
테이블명 최대 30byte(한글 15자), 유니크(중복허용안함)  
오라클 키워드(예약어)를 사용하지 말 것(충돌날 수 있음)

### 일반테이블

```
CREATE TABLE 테이블명(  
    컬럼명1 컬럼1타입,  
    컬럼명2 컬럼2타입,  
    ...  
);
```

### 임시테이블

```
CREATE GLOBAL TEMPORARY  
TABLE 테이블명(  
    컬럼명1 컬럼1타입,  
    컬럼명2 컬럼2타입,  
    ) ON COMMIT [DELETE | PRESERVE] ROWS;
```

```
CREATE table sample1(  
    num      number primary key,  
    fname    varchar2(20),  
    lname    varchar2(20) not null,  
    tel      varchar2(20) not null,  
    addr     varchar2(100),  
    regdate  date default sysdate  
);
```

**CREATE** sequence sample1\_seq; //자동번호부여용 시퀀스 생성

**\*\* CREATE TABLE** 테이블명 **AS SELECT \* FROM** 대상테이블; //데이터포함.

**\*\* CREATE TABLE** 테이블명 **AS SELECT \* FROM** 대상테이블 **WHERE 1=2;**  
//데이터없이 구조만.

2-1) **ALTER** : 객체를 변경할 때  
 ( ADD, MODIFY, RENAME, DROP COLUMN, SET UNUSED COLUMN)  
 CREATE table sample2 (num number);

ALTER table sample2 **ADD** (fname varchar2(20)); // 추가  
 ALTER table sample2 **ADD** (lname varchar2(20)); // 추가  
 ALTER table sample2 **ADD** (tel varchar2(20)); // 추가  
 ALTER table sample2 **ADD** (addr varchar2(100)); // 추가  
 ALTER table sample2 **ADD** (memo varchar2(100)); // 추가

ALTER table sample2 **MODIFY** (memo date); //타입,길이,not null,primary key값 수정  
 ALTER table sample2 **MODIFY** (num number primary key); //default 설정 수정  
 ALTER table sample2 **MODIFY** (memo date default sysdate); //default 설정 수정

ALTER table sample2 **RENAME COLUMN** memo **TO** regdate; //컬럼명 수정

ALTER table sample2 **DROP COLUMN** addr; // 삭제  
 ALTER table sample2 **ADD** (addr varchar2(100)); // 추가

ALTER table sample2 **ADD** (addr2 varchar2(100)); // 추가  
 ALTER table sample2 **ADD** (addr3 varchar2(100)); // 추가  
 ALTER table sample2 **ADD** (addr4 varchar2(100)); // 추가  
 ALTER table sample2 **SET UNUSED COLUMN** addr2; // 사용하지않는 컬럼 한개 설정  
 ALTER table sample2 **SET UNUSED** (addr3,addr4); // 사용하지않는 컬럼 여러개 설정  
 ALTER table sample2 **DROP UNUSED COLUMNS**; // 사용하지않는 설정된 컬럼들 지우기

## 2-2) ALTER : 객체를 변경할 때

(ADD CONSTRAINT,DROP CONSTRAINT,  
DISABLE CONSTRAINT, ENABLE CONSTRAINT,READ ONLY,READ WRITE)

CREATE table sample3 ( num number, join\_id varchar2(20) );

CREATE table sample4 ( join\_id varchar2(20) primary key );

ALTER table sample3 ADD CONSTRAINT SAMPLE\_PK PRIMARY KEY(num);

ALTER table sample3 DROP CONSTRAINT SAMPLE\_PK;

ALTER table sample3 ADD CONSTRAINT SAMPLE\_FK\_JOIN\_ID

FOREIGN KEY(join\_id) REFERENCES sample4(join\_id);

ALTER table sample3 DROP CONSTRAINT SAMPLE\_FK\_JOIN\_ID;

ALTER table sample3 DISABLE CONSTRAINT SAMPLE\_FK\_JOIN\_ID;//제약조건 비활성화

ALTER table sample3 ENABLE CONSTRAINT SAMPLE\_FK\_JOIN\_ID;//제약조건 활성화

ALTER table sample3 DISABLE PRIMARY KEY CASCADE;// 제약조건 무시하고 기본키 비활성화

ALTER table sample3 ENABLE PRIMARY KEY;// 제약조건 무시하고 기본키 활성화

ALTER table sample3 READ ONLY; ALTER table sample3 READ WRITE;//11g 추가기능.

## 3) DROP : 객체를 삭제할 때

DROP table sample3;

DROP table sample4 CASCADE CONSTRAINT;//제약조건 무시 삭제

DROP table sample4 CASCADE CONSTRAINT PURGE;//recyclebin저장무시

## 4) TRUNCATE : DELETE 처럼 데이터를 삭제할때 사용.

차이점-DELETE는 롤백가능, DDL 은 롤백이 안된다.

TRUNCATE TABLE SAMPLE1;

## 2. DML (Data Manipulation Language)

테이블내의 데이터를 입력,수정,삭제,병합할 수 있는 SQL언어

### 1-1) INSERT 문 : 테이블의 데이터를 저장(입력)할 때 사용

[ 형식 ] INSERT INTO 테이블명(컬럼명1,컬럼명2,...) VALUES (값1,값2,...)

[입력시 제약사항]

- 한번에 하나의 행만 입력할 수 있다.
- INSERT 절에 명시되는 컬럼갯수와 VALUES 절의 개수는 같아야 한다.  
( INSERT 절에 명시가 되지 않은 경우 테이블의 모든 컬럼갯수가 적용 )
- 입력될 값의 데이터타입과 컬럼타입은 같아야 한다.
- 자동 converting
  - a. NUMBER 타입에 '숫자형태의 문자열' 입력시  
자동으로 컨버팅(문자열 --> NUMBER)기능이 있음
  - b. 문자열타입(VARCHAR2)에 '숫자'를 입력시  
자동으로 컨버팅(NUMBER -> 문자열) 기능이 있음



## 1-2) INSERT 문 : 다음을 작성하자.

```
create table sample1(  
  num      number primary key,  
  fname    varchar2(20),  
  lname    varchar2(20) not null,  
  tel      varchar2(20) not null,  
  addr     varchar2(100),  
  regdate  date default sysdate  
);  
create sequence sample1_seq;
```

```
insert into  
sample1( num,  
         fname,  
         lname,  
         tel,  
         addr )  
values( sample1_seq.nextval,  
        dbms_random.string('A',19),  
        dbms_random.string('Q',19),  
        '010-0000-0000',  
        'seoul' );
```

```
PL/SQL  
begin  
  for i in 1..50 loop  
    insert into  
      sample1(num,fname,lname,tel,addr)  
    values(sample1_seq.nextval,  
           dbms_random.string('A',19),  
           dbms_random.string('Q',19),  
           '010-0000-0000',  
           'seoul');  
  end loop;  
  commit;  
end;  
/  
Select count(*) from sample1;
```

## 2) UPDATE 문 : 테이블의 데이터를 변경(수정)할 때 사용

[ 형식 ] UPDATE 테이블명 SET 컬럼명1=수정값1, 컬럼명2=수정값2,...  
WHERE 조건

```
UPDATE sample1 SET fname ='yangssem' WHERE num=1;  
select * from sample1;
```

```
UPDATE sample1 SET fname ='yangssem2' WHERE fname = 'yangssem';
```

```
UPDATE sample1  
SET fname ='yangssem3',  
    lname = 'oracle',  
    tel='010-2755-9494',  
    addr='pusan',  
    regdate='13/04/20'  
WHERE fname = 'yangssem2';
```

```
UPDATE sample1  
SET num=500001  
WHERE fname = 'yangssem3';
```



### 3) DELETE 문 : 테이블의 데이터를 삭제할 때 사용

[ 형식 ] DELETE FROM 테이블명 WHERE 조건절

```
DELETE FROM sample1 WHERE num=3;
```

```
DELETE FROM sample1 WHERE fname like '%yangssem%';
```

### 4-1) MERGE 문 : 테이블의 데이터를 병합 할 때 사용

[ 형식 ] **MERGE INTO** 토탈테이블 t **USING** 작은테이블 s **ON** (병합조건 t.id=s.id)  
**WHEN MATCHED THEN**  
UPDATE SET 업데이트내용 t.col1=s.col1  
[DELETE WHERE 조건절]  
**WHEN NOT MATCHED THEN**  
INSERT VALUES(컬럼명 s1.id, s1.col1, s1.col2...);

강사 양영환

```
create table TEST1 (pnum number,p2num number,pcount number,price number);  
insert into test1 values(1001,2001,100,5000);  
insert into test1 values(1002,2002,100,3000);  
insert into test1 values(1003,2003,100,2000);
```

```
create table TEST2 (pnum number,p2num number,pcount number,price number);  
insert into test2 values(6001,7001,300,7000);  
insert into test2 values(6002,7002,300,8000);  
insert into test2 values(6003,7003,300,9000);
```

```
create table TEST_merge (pnum number,p2num number,pcount number,price  
number);
```





## 4-2) MERGE 문 : 데이터가 쌓일수록 느려짐,복잡한 문법,다양한 변형, 자주쓰임.

test1

	PNUM	P2NUM	PCOUNT	PRICE
1	1001	2001	100	5000
2	1002	2002	100	3000
3	1003	2003	100	2000

test2

	PNUM	P2NUM	PCOUNT	PRICE
1	6001	7001	300	7000
2	6002	7002	300	8000
3	6003	7003	300	9000

test\_merge

	PNUM	P2NUM	PCOUNT	PRICE
1	1003	2003	100	2000
2	1002	2002	100	3000
3	1001	2001	100	5000
4	6001	7001	300	7000
5	6003	7003	300	9000
6	6002	7002	300	8000

### \*\*\*1. TEST\_MARGE 테이블 + TEST1테이블

```
MERGE INTO test_merge tm USING test1 t1
ON (tm.pnum=t1.pnum)
WHEN MATCHED THEN
UPDATE SET tm.p2num=t1.p2num
WHEN NOT MATCHED THEN
INSERT VALUES(t1.pnum, t1.p2num, t1.pcount, t1.price);
```

### \*\*\*2. TEST\_MARGE 테이블 + TEST2테이블

```
MERGE INTO test_merge tm USING test2 t2
ON (tm.pnum=t2.pnum)
WHEN MATCHED THEN
UPDATE SET tm.p2num=t2.p2num
WHEN NOT MATCHED THEN
INSERT VALUES(t2.pnum, t2.p2num, t2.pcount, t2.price);
```

select \* from test\_merge;

문제] 임의의 값으로 수정한 내용을 병합해서 확인하라

## 2. TCL (Transection Control Language)

DML작업을 한 세트로 관리하는 논리적 작업.

COMMIT, SAVEPOINT, ROLLBACK

- 1) COMMIT : 트랜잭션의 끝이자 새로운 시작을 나타냄.
- 2) SAVEPOINT : 현재까지 트랜잭션을 특정이름으로 저장함  
하나의 트랜잭션 안에 SAVEPOINT를 줘서 구역을 나눈다.
- 3) ROLLBACK : 트랜잭션 실행 이후로 되돌리거나 SAVEPOINT 지정된 상태로 되돌린다.

예) 부서테이블을 복사해서 TEST\_TCL 테이블을 만들자.

```
CREATE TABLE TEST_TCL AS SELECT * FROM DEPARTMENTS;
```

```
SELECT * FROM TEST_TCL;
```

```
COMMIT; //트랜잭션 시작
```

```
UPDATE TEST_TCL SET DEPARTMENT_ID=50;
```

```
UPDATE TEST_TCL SET LOCATION_ID=1004 WHERE DEPARTMENT_NAME='IT';
```

```
DELETE FROM TEST_TCL WHERE DEPARTMENT_NAME='Marketing';
```

```
SELECT * FROM TEST_TCL;
```

```
ROLLBACK;
```

```
SELECT * FROM TEST_TCL;
```

CREATE, ALTER, DROP, RENAME, TRUNCATE등의 DDL문을 사용할때에는 자동으로 COMMIT이 적용 된다.

[실습] 게시판(test\_board) 테이블을 만들자.

( CREATE, INSERT, UPDATE, DELETE, DROP ,ALTER,TRUNCATE)

[ 컬럼 ]

글번호(정수) WNUM 기본키

작성자(문자) WRITER NULL허용안함

글제목(문자) TITLE NULL허용안함

글내용(문자) CON NULL허용안함

작성날짜(날자) WDATE NULL허용안함 기본값 시스템현재날자

조회수(정수) VCOUNT 기본값 0

- 글번호는 자동번호 부여해서 증가시킬 것.
- 임의의 데이터를 10개 추가.
- 글번호가 4번인 데이터를 삭제.
- 작성자로도 데이터를 삭제.
- 작성자 YANGSSEM 추가
- 작성자가 'YANGSSEM' 사람의 글내용을 '즐거운 ORACLE SQL' 로 수정.
- 글제목이나 내용중에 ORACLE 라는 단어가 있는 모든 글 검색.
- 댓글허용(COMM\_CH,문자) 컬럼 추가
- 모든데이터를 잘라내시오.
- 테이블을 삭제 해보세요.

