

다양한 예제로 쉽게 배우는

# 오라클 SQL 과 PL/SQL

# 7장. DDL 과 Data Dictionary를 배웁니다

# 7장. DDL & Data Dictionary

## 1. CREATE

### 1)사용 예 1: 일반 테이블 생성하기

```
SCOTT>CREATE TABLE ddl_test
2 ( no NUMBER(3) ,
3   name VARCHAR2(10),
4   birth DATE DEFAULT SYSDATE )
5 TABLESPACE users ;
```



DDL_TEST		
NO	NAME	BIRTH

# 7장. DDL & Data Dictionary

## 2) 사용 예 2: 한글로 테이블 생성하기

```
SCOTT>CREATE TABLE 한글테이블  
2 ( 컬럼1 number ,  
3   컬럼2 varchar2(10) ,  
4   컬럼3 date ) ;
```

# 7장. DDL & Data Dictionary

## - 테이블 생성시 주의사항

1. 테이블 이름은 반드시 문자로 시작해야 합니다. 즉 숫자로 시작할 수는 없고 숫자가 포함되는 것은 가능합니다. 특수문자도 가능하지만 테이블 생성시 " (겹따옴표) 로 감싸야 하며 권장하지 않습니다.
2. 테이블 이름이나 컬럼 이름은 최대 30 bytes 까지 가능합니다. 즉 한글로 테이블 이름을 생성하실 경우 최대 15글자 까지만 가능하다는 뜻입니다.
3. 테이블 이름은 한 명의 사용자가 다른 오브젝트들의 이름과 중복으로 사용할 수 없습니다. 예를 들어 scott 사용자가 테이블명을 test 로 생성한 후 인덱스 이름을 test 로 동일하게 사용할 수 없다는 것입니다. 그러나 scott 사용자가 test 테이블 만들어도 다른 사용자인 hr 사용자는 test 라는 테이블 이름을 사용할 수 있습니다.
4. 테이블 이름이나 오브젝트 이름을 오라클이 사용하는 키워드를 사용하지 않기를 권장합니다. 오라클 키워드라 함은 오라클에서 사용하는 미리 정해진 SELECT , FROM 등과 같은 단어들을 말합니다. 생성이 안되는 것은 아니지만 사용시에 아주 불편하고 위험 할 수도 있기에 절대로 사용하지 말기를 권장합니다.

# 7장. DDL & Data Dictionary

## 3) 사용 예 3: Temporary Table (임시 테이블) 생성하기

```
CREATE GLOBAL TEMPORARY TABLE 테이블명
( 컬럼1 데이터 타입,
  컬럼2 데이터 타입, .....,
  ON COMMIT [ delete | preserve ] ROWS ;
```

-8.1 버전부터 등장함.

- 위 문법에서 마지막 행에 ON COMMIT delete ROWS 를 사용하면 COMMIT 시에 데이터를 삭제한다는 뜻이고 ON COMMIT preserve ROWS 를 사용하면 세션이 종료해야 데이터가 사라집니다.  
기본값은 **ON COMMIT delete ROWS** 입니다.

# 7장. DDL & Data Dictionary

- 실 습 1 : 터미널을 2 개 열어서 한쪽에서 생성 후 다른 쪽에서 조회 확인하기

- 터미널 1

```
SCOTT>CREATE GLOBAL TEMPORARY TABLE temp01
2 ( no number ,
3   name varchar2(10))
4 ON COMMIT DELETE ROWS ;
```

Table created.

```
SCOTT>INSERT INTO temp01 VALUES(1,'AAA');
```

1 row created.

```
SCOTT>SELECT * FROM temp01 ;
```

NO	NAME
1	AAA

# 7장. DDL & Data Dictionary

## - 터미널 2

```
SCOTT>SELECT * FROM temp01 ;
```

no rows selected    <- 다른 세션에서는 조회가 안됩니다



# 7장. DDL & Data Dictionary

## - 터미널 1

```
SCOTT>SELECT * FROM temp01 ;
```

```
NO NAME
```

```
-----
```

```
1 AAA
```

```
SCOTT>COMMIT ;
```

Commit complete.

```
SCOTT>SELECT * FROM temp01 ;
```

no rows selected <- 데이터가 삭제 되어서 조회가 안됩니다.

temporary table 생성 옵션이 ON COMMIT **DELETE** ROWS 이므로 커밋을 하면 모두 삭제됩니다

# 7장. DDL & Data Dictionary

## - 실 습 2 : 생성되어 있는 Temporary Table 조회하기

```
SCOTT>SELECT temporary, duration
```

```
2 FROM user_tables
```

```
3 WHERE table_name='TEMP01' ;
```

```
T DURATION
```

```
- - - - -
```

```
Y SYS$TRANSACTION
```

# 7장. DDL & Data Dictionary

## 4) 사용 예 4: 테이블 복사하기 (CTAS 라고도 합니다)

### (1) 모든 칼럼 다 복사하기

```
SCOTT>CREATE TABLE dept3  
2 AS  
3 SELECT * FROM dept2;
```

### (2) 특정 칼럼만 복사하기

```
SCOTT>CREATE TABLE dept4  
2 AS  
3 SELECT dcode , dname  
4 FROM dept2 ;
```

# 7장. DDL & Data Dictionary

## (3) 테이블의 구조(칼럼)만 가져오고 데이터 안 가져오기

```
SCOTT>CREATE TABLE dept5  
2 AS  
3 SELECT *  
4 FROM dept2  
5 WHERE 1=2 ;
```

# 7장. DDL & Data Dictionary

## (5) 사용 예 5: 가상 컬럼 테이블 생성하기 (11g 부터 추가된 기능)

- Step 1. 가상 컬럼을 가지는 vt001 테이블을 생성합니다.

```
SCOTT>CREATE TABLE vt001  
2 ( no1 number ,  
3   no2 number ,  
4   no3 number GENERATED ALWAYS AS (no1 + no2 ) VIRTUAL ) ;
```

위 명령어에서 4번 라인의 no3 컬럼은 no1+no2의 값을 가지는 가상 컬럼입니다.

# 7장. DDL & Data Dictionary

-Step 2. vt001 테이블에 데이터를 입력합니다.

```
SCOTT>INSERT INTO vt001 VALUES (1,2,3);  
INSERT INTO vt001 VALUES (1,2,3)
```

\*

```
ERROR at line 1:  
ORA-54013: INSERT operation disallowed on virtual columns
```

```
SCOTT>INSERT INTO vt001(no1,no2)  
2 VALUES(1,2);
```

```
1 row created.
```

위 step 2의 결과로 알 수 있듯이 가상 컬럼에는 사용자가 데이터를 입력할 수 없습니다.

# 7장. DDL & Data Dictionary

- Step 3. 입력된 데이터를 조회합니다.

```
SCOTT>SELECT * FROM vt001 ;
```

NO1	NO2	NO3
1	2	3

# 7장. DDL & Data Dictionary

- Step 4. 기존 값을 변경 한 후 가상 칼럼에 반영되는 지 확인합니다.

```
SCOTT>UPDATE vt001
2 SET no1=10 ;
```

1 row updated.

```
SCOTT>SELECT * FROM vt001 ;
```

NO1	NO2	<b>NO3</b>
10	2	<b>12</b>



# 7장. DDL & Data Dictionary

- Step 5. 인덱스와 제약조건이 생성 가능한 지 테스트 합니다.

```
SCOTT>INSERT INTO vt001 (no1, no2)
2 VALUES (3,4);
```

1 row created.

```
SCOTT>INSERT INTO vt001 (no1,no2)
2 VALUES(6,6) ;
```

```
INSERT INTO vt001 (no1,no2)
```

\*

ERROR at line 1:

ORA-00001: unique constraint (SCOTT.IDX\_VT001\_NO3) violated

# 7장. DDL & Data Dictionary

- Step 6. 새로운 가상 컬럼을 추가합니다.

```
SCOTT>ALTER TABLE vt001
2 ADD (no4 GENERATED ALWAYS AS ((no1*12)+no2)) ;
```

Table altered.

```
SCOTT>SELECT * FROM vt001 ;
```

NO1	NO2	NO3	NO4
10	2	12	122
3	4	7	40

위 Step 6 에서 보듯이 새로운 가상 컬럼이 추가 되면 즉시 값이 반영되어 생성됩니다.

# 7장. DDL & Data Dictionary

- Step 7. 테이블에서 가상 칼럼 내역을 조회합니다.

```
SCOTT>set line 200
SCOTT>col column_name for a10
SCOTT>col data_type for a10
SCOTT>col data_default for a25
SCOTT>
SCOTT>SELECT column_name
       2      , data_type
       3      , data_default
       4 FROM user_tab_columns
       5 WHERE table_name = 'VT001'
       6 ORDER BY column_id ;
```

COLUMN_NAME	DATA_TYPE	DATA_DEFAULT
-----	-----	-----
NO1	NUMBER	
NO2	NUMBER	
<b>NO3</b>	<b>NUMBER</b>	<b>"NO1"+"NO2"</b>
<b>NO4</b>	<b>NUMBER</b>	<b>"NO1"*12+"NO2"</b>

# 7장. DDL & Data Dictionary

## Step 8. 조건절을 활용한 가상컬럼 생성하기

```
SCOTT>CREATE TABLE panmae10
2 (no      NUMBER,
3  pcode  CHAR(4),
4  pdate  CHAR(8),
5  pqty   NUMBER,
6  pbungi NUMBER(1)
7  GENERATED ALWAYS AS
8  (
9  CASE
10   WHEN SUBSTR(pdate,5,2) IN ('01','02','03') THEN 1
11   WHEN SUBSTR(pdate,5,2) IN ('04','05','06') THEN 2
12   WHEN SUBSTR(pdate,5,2) IN ('07','08','09') THEN 3
13   ELSE 4
14  END ) virtual );
```

Table created.

# 7장. DDL & Data Dictionary

```
SCOTT>INSERT INTO panmae10 (no,pcode,pdate,pqty)
2  VALUES(1,'100','20110112',10) ;
```

1 row created.

```
SCOTT>INSERT INTO panmae10 (no,pcode,pdate,pqty)
2  VALUES(2,'200','20110505',20);
```

1 row created.

```
SCOTT>INSERT INTO panmae10 (no,pcode,pdate,pqty)
2  VALUES(3,'300','20110812',30);
```

1 row created.

# 7장. DDL & Data Dictionary

```
SCOTT>INSERT INTO panmae10 (no,pcode,pdate,pqty)
2  VALUES(4,'400','20111024',40);
```

1 row created.

```
SCOTT>COMMIT ;
```

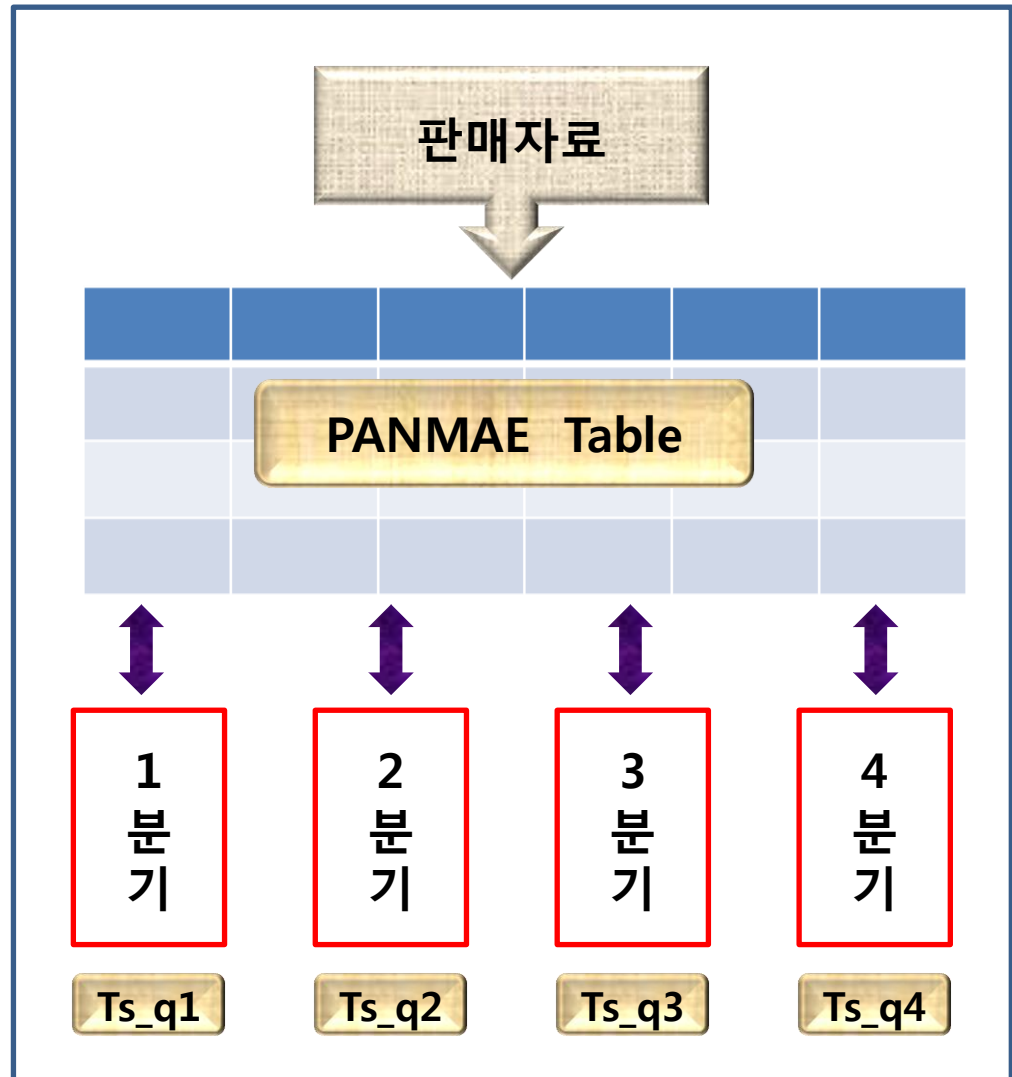
Commit complete.

```
SCOTT>SELECT * FROM panmae10 ;
```

NO	PCOD	PDATE	PQTY	PBUNGI
1	100	20110112	10	1
2	200	20110505	20	2
3	300	20110812	30	3
4	400	20111024	40	4

# 7장. DDL & Data Dictionary

## 5) 파티션 테이블 생성하기



# 7장. DDL & Data Dictionary

## - 오라클 버전별로 제공하는 파티션 종류

Oracle Version	지원 또는 추가된 PARTITION 종류
Oracle 8	Range PARTITION Fundmental maintenance operations Static pruning
Oracle 8i	Hash PARTITION , Range-Hash PARTITION Merge PARTITIONs Dynamic pruning PARTITION-wise joins
Oracle 9i R1	List PARTITION Global index maintenance
Oracle 9i R2	Range-List PARTITION Fast SPLIT Default PARTITION for List



# 7장. DDL & Data Dictionary

## - 오라클 버전별로 제공하는 파티션 종류

Oracle 10g R1	Composite PARTITION Range(-list, -hash) Internal re-architecture Global hash-partitioned indexes Local index maintenance
Oracle 10g R2	One million partitions Multi-dimensional pruning Resource optimized drop table
Oracle 11g	Range, list hash partitioning Extended composite partitioning <ul style="list-style-type: none"> <li>- range (-list , -hash , range)</li> <li>- list (-range, -list – hash)</li> <li>- interval (-range , -list, -hash)</li> </ul> Interval partitioning REF partitioning Virtual column based partitioning Enable infinite partitioning Design possibilities and boost manageability

# 7장. DDL & Data Dictionary

## (1) Range PARTITION (범위 파티셔닝)

이 파티셔닝은 주로 특정 기준에 의해서 범위를 나눌 때 사용하는 방법입니다. 예를 들어 판매 테이블을 파티셔닝 하는데 판매 날짜를 기준으로 범위를 나누는 경우나 포털 사이트의 카페 테이블을 파티셔닝 하는데 카페 번호로 범위를 나눌 경우 등에 사용됩니다.

단점은 각 파티션 별로 데이터가 균등하게 분포되지 않을 수도 있어서 성능이 보장이 안된다는 점입니다.

실습은 교재 229 – 243 페이지를 참고하세요

# 7장. DDL & Data Dictionary

## (2) HASH PARTITION (8i 버전부터 지원)

이 기법은 Hash 함수가 데이터를 각 테이블스페이스 별로 균등하게 분포시키기 때문에 데이터가 균등하게 분포되어 성능이 향상된다는 장점이 있습니다.

그러나 이 기법은 데이터를 사용자가 분산하지 않고 Hash 함수가 분산시키기 때문에 데이터의 관리 등이 매우 어렵다는 단점 또한 존재합니다.

실습은 교재 243 – 244 페이지를 참고하세요

# 7장. DDL & Data Dictionary

## (3) LIST PARTITION (목록 분할 파티션)

이 기법은 파티셔닝 할 항목을 관리자가 직접 지정하는 방식입니다. 예를 들어 회사 제품 중에 A 제품용 파티션, B 제품용 파티션, C 제품용 파티션을 각각 생성하는 방식입니다. 이 방식은 잘 설정 할 경우 빠른 성능을 보장 할 수 있지만 잘못 설정 될 경우 오히려 성능이 저하 될 수 있기에 아주 주의해서 생성해야 합니다.

실습은 교재 245 – 250 페이지를 참고하세요

# 7장. DDL & Data Dictionary

## (4) COMPOSITE PARTITION (복합 파티션)

Composite Partition 은 위에서 언급한 여러 가지 파티션을 복합적으로 사용하는 것입니다.

예를 들어 학생이 많을 경우 학년별로 먼저 나누고 각 학년별로 다시 반으로 나누는 것처럼 파티셔닝을 먼저 한 후 다시 세부적으로 파티셔닝을 다시 하는 방법입니다.

오라클 버전별로 지원하는 종류가 아래와 같습니다.

8i : range – hash

9i : range –list 추가 지원

11g : range –range , list – range , list – list , list – hash 추가 지원

실습은 교재 251 – 256 페이지를 참고하세요

# 7장. DDL & Data Dictionary

## (5) Interval PARTITION (11g New Feature)

11g부터 추가된 기능으로 Range PARTITION 의 확장 형입니다.

Range PARTITION에서 만약 파티션의 범위(한계)를 벗어난 데이터가 입력이 될 경우 에러가 발생하면서 입력이 되지 않는데 이런 문제는 interval PARTITION 을 활용하면 **오라클이 필요한 파티션을 스스로 생성한 후 데이터를 입력하게 됩니다.**

이 때 자동으로 생성되는 파티션들은 모두 동일한 범위의 크기를 가지게 되며 파티션 이름은 오라클이 자동으로 지정하게 됩니다.

실습은 교재 256 – 263 페이지를 참고하세요

# 7장. DDL & Data Dictionary

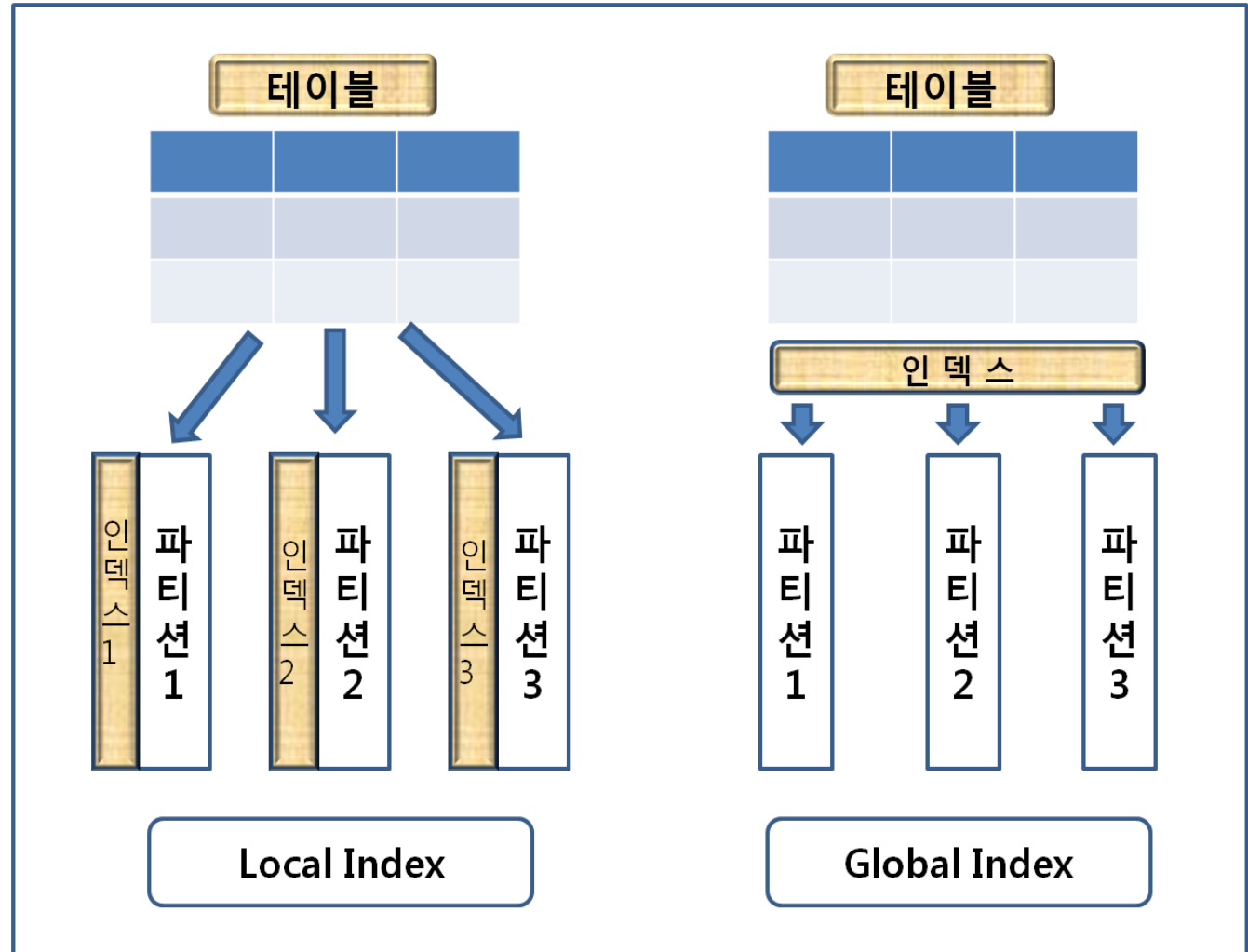
## (6) SYSTEM PARTITION (11g New Feature)

System PARTITION 은 PARTITION key 를 파티션 생성시에 지정하지 않고 데이터를 삽입할 때 직접 지정하는 방식입니다. 또한 검색할 때도 파티션 명을 명시해야만 해당 파티션에서 데이터를 찾게 되며 만약 명시하지 않으면 전체 파티션에서 데이터를 읽게 되어 속도가 급격히 저하 될 수 있습니다. 앞서 살펴 보았던 어떤 파티션도 적용할 수 없을 때 사용하기 위해 제공되는 기능입니다.

실습은 교재 264 페이지를 참고하세요

# 7장. DDL & Data Dictionary

## 6) 파티션의 인덱스





# 7장. DDL & Data Dictionary

## - Local Index 와 Global Index

Local Index 는 다시 Local Prefixed Index 와 Local Non-Prefixed Index 로 나누어 집니다.

Local Prefixed Index 는 파티션을 나눌 때 기준이 되는 컬럼으로 인덱스를 생성한 것이고 Local Non-Prefixed Index 는 파티션 생성 기준 컬럼 이외의 컬럼으로 인덱스를 생성한 것입니다.

Local Prefixed Index 는 Unique / Non Unique index 모두 생성할 수 있습니다.

반면 Global Index 는 인덱스와 파티션의 컬럼이나 범위가 다르게 생성됩니다

실습은 교재 265 – 267 페이지를 참고하세요

# 7장. DDL & Data Dictionary

## 2. ALTER 명령

Alter 명령어는 만들어져 있는 오브젝트를 변경하는 명령어입니다.

즉 테이블 같은 경우에는 컬럼을 추가하거나 컬럼을 삭제하거나 컬럼 이름이나 테이블 이름을 바꾸는 등의 작업을 할 수 있습니다. 이 명령어는 부하가 많이 걸리는 명령어이므로 사용량이 많은 시간에 수행하는 것은 아주 위험하니 특히 조심하셔야 합니다.

# 7장. DDL & Data Dictionary

## 1) 사용 예 1 : 새로운 컬럼을 추가하기

```
SCOTT>CREATE TABLE dept6
2 AS
3 SELECT dcode , dname
4 FROM dept2
5 WHERE dcode IN(1000,1001,1002) ;
Table created.
```

```
SCOTT>SELECT * FROM dept6;
```

DCODE	DNAME
1000	경영지원부
1001	재무관리팀
1002	총무팀

```
SCOTT>ALTER TABLE dept6
2 ADD ( LOC VARCHAR2(10) );
```

Table altered.

```
SCOTT>SELECT * FROM dept6;
```

DCODE	DNAME	LOC
1000	경영지원부	
1001	재무관리팀	
1002	총무팀	

# 7장. DDL & Data Dictionary

## - 기본값 지정하여 추가하기

```
SCOTT>ALTER TABLE dept6
2 ADD ( LOC2 varchar2(10) DEFAULT '서울' );
```

```
SCOTT>SELECT * FROM dept6;
```

DCODE	DNAME	LOC	LOC2
1000	경영지원부		서울
1001	재무관리팀		서울
1002	총무팀		서울

# 7장. DDL & Data Dictionary

## 2) 사용 예 2: 테이블의 칼럼 이름 변경하기

```
SCOTT> ALTER TABLE dept6 RENAME COLUMN LOC2 TO AREA ;
```

```
SCOTT> RENAME dept6 TO dept7 ; -- 테이블 이름 변경하기
```

# 7장. DDL & Data Dictionary

## 3) 사용 예 3: 칼럼의 데이터 크기를 변경하기

SCOTT>DESC dept7;

Name	Null?	Type
<b>DCODE</b>		<b>VARCHAR2(6)</b>
DNAME	NOT NULL	VARCHAR2(20)
LOC		VARCHAR2(10)
AREA		VARCHAR2(10)

SCOTT>ALTER TABLE dept7

2 MODIFY(dcode **VARCHAR2(10)**);

Table altered.

SCOTT>DESC dept7;

Name	Null?	Type
<b>DCODE</b>		<b>VARCHAR2(10)</b>
DNAME	NOT NULL	VARCHAR2(20)
LOC		VARCHAR2(10)
AREA		VARCHAR2(10)

# 7장. DDL & Data Dictionary

## 4) 사용 예 4 : 칼럼 삭제하기

```
SCOTT>ALTER TABLE dept7 DROP COLUMN loc ;
```

```
SCOTT>ALTER TABLE dept7 DROP COLUMN loc CASCADE CONSTRAINTS ;
```

# 7장. DDL & Data Dictionary

## 5) 읽기 전용 테이블로 변경하기 - 11g New Feature

```
SCOTT>CREATE TABLE t_read  
2 ( no NUMBER ,  
3   name VARCHAR2(10) );
```

Table created.

```
SCOTT>INSERT INTO t_read  
2 VALUES (1,'AAA');
```

1 row created.

```
SCOTT>COMMIT ;
```

Commit complete.



# 7장. DDL & Data Dictionary

```
SCOTT>SELECT * FROM t_read ;
```

```
NO NAME
```

```
-----  
1 AAA
```

```
SCOTT>ALTER TABLE t_read read only ; <- 읽기전용으로 변경합니다.
```

Table altered.

- 읽기 전용으로 변경된 테이블에 데이터 입력 시도함.

```
SCOTT>INSERT INTO t_read
```

```
2 VALUES (2,'BBB') ;
```

```
INSERT INTO t_read
```

```
*
```

ERROR at line 1:

ORA-12081: update operation not allowed on table "SCOTT"."T\_READ"

# 7장. DDL & Data Dictionary

- 읽기전용으로 변경된 테이블에 컬럼 추가 시도함

```
SCOTT>ALTER TABLE t_read  
2 ADD (tel number default 111) ;  
ALTER TABLE t_read  
*
```

ERROR at line 1:

ORA-12081: update operation not allowed on table "SCOTT"."T\_READ"

- 읽기 전용인 테이블 삭제 시도함

```
SCOTT>DROP TABLE t_read ;
```

Table dropped.

```
SQL> ALTER TABLE t_read read write ; -- 읽기 쓰기 모드로 변경함
```

# 7장. DDL & Data Dictionary

## 3. TRUNCATE 명령

```
SCOTT>TRUNCATE TABLE dept7 ;
```

## 4. DROP 명령

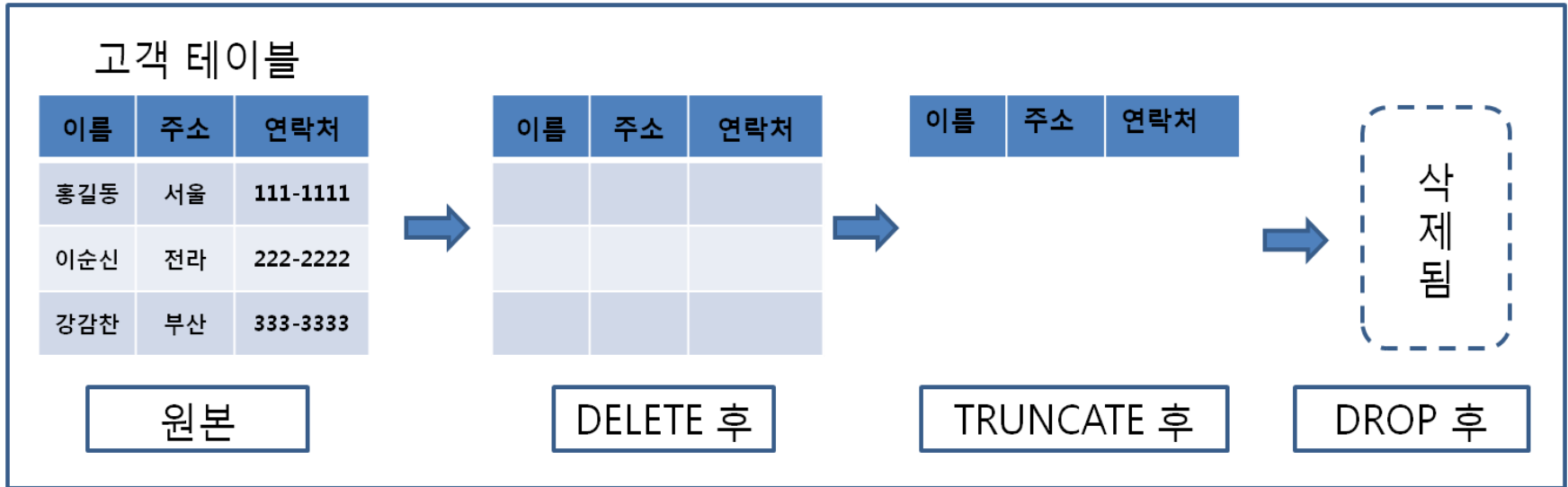
```
SCOTT>DROP TABLE dept7 ;
```

오라클 10g 부터는 위 명령어로 테이블을 삭제할 경우 테이블이 삭제 되는 것이 아니라 마치 윈도우에서 파일 삭제 시 휴지통으로 가는 것처럼 휴지통으로 보내지게 됩니다.

그리고 삭제된 테이블의 이름은 BIN\$....로 변경됩니다. 이 기능은 10g 부터 등장한 FLASHBACK 의 휴지통이라는 기능 때문입니다

# 7장. DDL & Data Dictionary

## 5. DELETE , TRUNCATE , DROP 명령어의 차이점



# 7장. DDL & Data Dictionary

## 6. 데이터 디렉터리 ( Dictionary )

- 데이터 디렉터리에 저장되어 있는 주요 내용들
  - \* 오라클 데이터베이스의 메모리 구조와 파일에 대한 구조 정보들
  - \* 각 오브젝트들이 사용하고 있는 공간들의 정보들
  - \* 제약 조건 정보들
  - \* 사용자에 대한 정보들
  - \* 권한이나 프로파일 , 롤에 대한 정보들
  - \* 감사(Audit) 에 대한 정보들
- Base Table 과 Data Dictionary View 로 이원화 되어 있음.
- DBA\_ , ALL\_ USER\_ 로 구분되어 있음.
- Static Dictionary 와 Dynamic Performance View 로 나눌 수 있음.

실습은 교재 277 – 278 페이지 참고하세요