

DDL과 제약조건

DDL

제약조건

목차

- DDL
 - ▣ CREATE 명령
 - ▣ ALTER 명령
 - ▣ TRUNCTATE 명령
 - ▣ DROP 명령
- 제약조건
 - ▣ 제약조건의 종류
 - ▣ 제약조건의 설정 방법
 - ▣ 제약조건의 관리

1. DDL

□ SOL 명령어들

- **DML** (Data Manipulation Language) : INSERT(입력) , UPDATE(변경) ,
DELETE(삭제) , MERGE(병합)
- **DDL** (Data Definition Language) : CREATE (생성) , ALTER (수정) ,
TRUNCATE (잘라내기) , DROP (삭제)
- * **DCL** (Data Control Language) : GRANT (권한 주기) , REVOKE (권한 뺏기)
- * **TCL** (Transaction Control Language): COMMIT (확정) , ROLLBACK (취소)
- * **SELECT** : 어떤 분류에서는 DQL (Data Query Language) 라고 하기도 함.

DDL-CREATE

□ 테이블 생성

```
CREATE TABLE 테이블명  
  ( 컬럼1 데이터타입,  
    컬럼2 데이터타입,  
    컬럼3 데이터 타입)
```

```
CREATE TABLE ddl_test  
  ( no  NUMBER(3) ,  
    name VARCHAR2(10),  
    birth DATE DEFAULT SYSDATE )
```

□ 한글로 테이블 생성하기

```
CREATE TABLE 한글테이블  
  ( 컬럼1 number ,  
    컬럼2 varchar2(10) ,  
    컬럼3 date );
```

1. DDL

□ 테이블 생성시 주의 사항

1. 테이블 이름은 반드시 문자로 시작. 숫자가 포함 가능. 특수문자도 가능하지만 테이블 생성시 " (겹따옴표) 로 감싸야 하며 권장하지 않음.
2. 테이블 이름이나 컬럼 이름은 최대 30 bytes 까지 가능, 즉 한글로 테이블 이름을 생성하실 경우 최대 10글자 까지만 가능.
3. 테이블 이름은 한 명의 사용자가 다른 오브젝트들의 이름과 중복으로 사용할 수 없음. 예를 들어 scott 사용자가 테이블명을 test 로 생성한 후 인덱스 이름을 test 로 동일하게 사용할 수 없음.
4. 테이블 이름이나 오브젝트 이름을 오라클이 사용하는 키워드를 사용하지 않기를 권장. 오라클 키워드란 오라클에서 사용하는 미리 정해진 SELECT , FROM 등과 같은 단어들을 말함. 생성은 가능하지만 사용시에 아주 불편하고 위험 할 수도 있기에 절대로 사용하지 말기를 권장

□ 테이블 복사하기

(1) 모든 칼럼 다 복사하기

```
CREATE TABLE dept3  
AS  
SELECT * FROM dept2;
```

(2) 특정 칼럼만 복사하기

```
CREATE TABLE dept4  
AS  
SELECT dcode , dname  
FROM dept2 ;
```

(3) 테이블의 구조(칼럼)만 가져오고 데이터 안 가져오기

```
CREATE TABLE dept5  
AS  
SELECT *  
FROM dept2  
WHERE 1=2 ;
```

(4) 가상 컬럼 테이블 생성하기 (11g , 12c 가능)

- Step 1. 가상 컬럼을 가지는 vt001 테이블을 생성합니다.

```
CREATE TABLE vtbl_1  
  ( no1  NUMBER ,  
    no2  NUMBER ,  
    no3  NUMBER GENERATED ALWAYS AS (no1 + no2 ) VIRTUAL ) ;
```

(4) 가상 컬럼 테이블 생성하기

-Step 2. vt001 테이블에 데이터를 입력합니다. (이 명령은 DML 에서 배웁니다)

```
INSERT INTO vtbl_1 VALUES (1,2,3);
```

명령의 7 행에서 시작하는 중 오류 발생 -

```
INSERT INTO vtbl_1 VALUES (1,2,3)
```

오류 발생 명령행: 7 열: 13

오류 보고 -

SQL 오류: ORA-54013: INSERT 작업은 가상 열에서 허용되지 않습니다.

54013. 0000 - "INSERT operation disallowed on virtual columns"

*Cause: Attempted to insert values into a virtual column

*Action: Re-issue the statment without providing values for a virtual column

(4) 가상 컬럼 테이블 생성하기

```
INSERT INTO vtbl_1(no1,no2)
VALUES(1,2);
```

1 row created.

```
SELECT * FROM vtbl_1 ;
```

NO1	NO2	NO3
-----	-----	-----

-------	--	--

1	2	3
---	---	---

가상 컬럼에는 사용자가 값을 입력할 수 없고 미리 정의된 공식으로 오라클이 입력.

(4) 가상 컬럼 테이블 생성하기

- Step 4. 기존 값을 변경 한 후 가상 컬럼에 반영되는 지 확인.

```
UPDATE vtbl_1
```

```
2 SET no1=10 ;
```

1 row updated.

```
SELECT * FROM vtbl_1 ;
```

NO1	NO2	NO3
-----	-----	------------

-------	--	--

10		
----	--	--

	2	
--	---	--

		12
--	--	-----------

Stage 3. 데이터를 변경하고 관리하기

Step 6. 새로운 가상 컬럼을 추가합니다.

```
ALTER TABLE vtbl_1
```

```
ADD (no4 GENERATED ALWAYS AS ((no1*12)+no2)) ;
```

Table altered.

```
SELECT * FROM vtbl_1 ;
```

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

Stage 3. 데이터를 변경하고 관리하기

Step 7. 테이블에서 가상 컬럼 내역을 조회합니다.

```
HR>SELECT column_name
      2      , data_type
      3      , data_default
      4 FROM user_tab_columns
      5 WHERE table_name = 'VTBL_1'
      6 ORDER BY column_id ;
```

COLUMN_NAME	DATA_TYPE	DATA_DEFAULT
NO1	NUMBER	
NO2	NUMBER	
NO3	NUMBER	"NO1"+"NO2"
NO4	NUMBER	"NO1"*12+"NO2"

DDL-ALTER

□ ALTER 명령

- ▣ 만들어져 있는 오브젝트를 변경하는 명령어
- ▣ 즉 테이블 같은 경우에는 컬럼을 추가하거나 컬럼을 삭제하거나 컬럼 이름이나 테이블 이름을 바꾸는 등의 작업을 할 수 있음.
- ▣ 부하가 많이 걸리는 명령어이므로 사용량이 많은 시간에 수행하는 것은 아주 위험하니 특히 조심.

DDL-ALTER

□ 새로운 칼럼을 추가하기

```
CREATE TABLE dept6  
AS  
SELECT dcode , dname  
FROM dept2  
WHERE dcode IN(1000,1001,1002) ;
```

```
ALTER TABLE dept6  
ADD ( LOC VARCHAR2(10) );
```

□ 기본값 지정하여 칼럼 추가

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

DDL-ALTER

□ 테이블 칼럼 이름 변경

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

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

□ 칼럼의 데이터 크기 변경

```
DESC dept7;
```

Name	Null?	Type
-----	-----	-----
DCODE		VARCHAR2(6)
.....		

```
ALTER TABLE dept7 MODIFY(dcode VARCHAR2(10)) ;
```

```
DESC dept7;
```

□ 칼럼 삭제하기

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

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


5) 읽기 전용 테이블로 변경하기 (11g , 12c 가능)

```
CREATE TABLE t_read  
( no NUMBER ,  
  name VARCHAR2(10) );
```

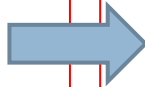
Table created.

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

1 row created.

```
COMMIT ;
```

Commit complete.



```
SELECT * FROM t_read ;
```

NO	NAME
1	AAA

```
ALTER TABLE t_read read only ;
```

Table altered.

Stage 3. 데이터를 변경하고 관리하기

- 읽기 전용으로 변경된 테이블에 데이터 입력 시도함.

```
INSERT INTO t_read  
VALUES (2,'BBB') ;
```

```
INSERT INTO t_read  
*
```

ERROR at line 1:

ORA-12081: update operation not allowed on table "HR"."T_READ"

Stage 3. 데이터를 변경하고 관리하기

- 읽기전용으로 변경된 테이블에 컬럼 추가 시도함

```
ALTER TABLE t_read  
  ADD (tel number default 111) ;
```

```
ALTER TABLE t_read
```

*

ERROR at line 1:

ORA-12081: update operation not allowed on table "HR"."T_READ"

ALTER TABLE t_read read write ; <- 읽기 쓰기 가능하도록 변경.

- 읽기 전용인 테이블이라도 삭제는 가능합니다.

```
DROP TABLE t_read ;
```

Table dropped.

□ TRUNCATE 명령어

- ▣ 테이블 데이터를 전부 삭제하고 사용하던 빈 공간을 반납
- ▣ 테이블은 삭제 되지 않음

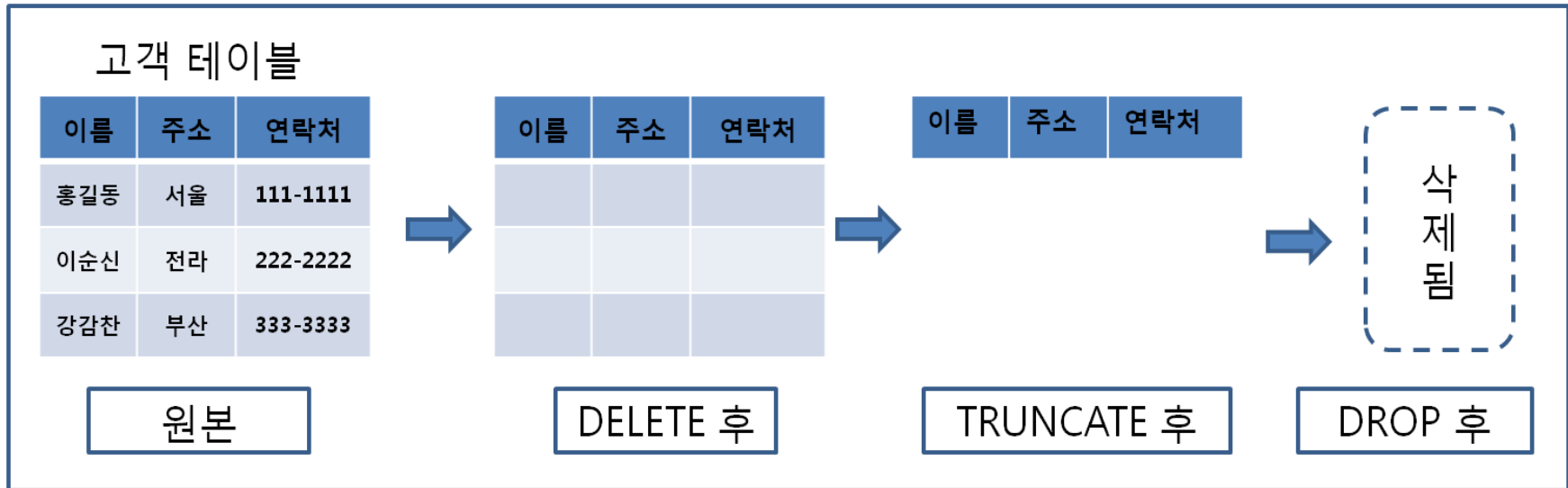
```
TRUNCATE TABLE dept7 ;
```

□ DROP 명령어

- ▣ 테이블 자체를 삭제하는 명령어

```
DROP TABLE dept7 ;
```

□ DELETE , TRUNCATE , DROP 명령어의 차이점



Stage 3. 데이터를 변경하고 관리하기

6. 데이터 디렉터리 (Dictionary)

- * 오라클 데이터베이스의 메모리 구조와 파일에 대한 구조 정보들
- * 각 오브젝트들이 사용하고 있는 공간들의 정보들
- * 제약 조건 정보들
- * 사용자에 대한 정보들
- * 권한이나 프로파일 , 룰에 대한 정보들
- * 감사(Audit) 에 대한 정보들

□ 디렉터리

- ▣ Base Table 과 Data Dictionary View 로 구성됨
- ▣ Base Table은 DBA 이라 할 지라도 접근을 못함.
- ▣ 사용자(DBA 포함)들은 Data Dictionary View 를 통해서만 접근 가능
- ▣ Base Table 은 Database 를 생성하는 시점에 자동으로 생성됨
- ▣ Data Dictionary View 는 Catalog.sql 이란 파일이 수행되어야만 만들어 짐.
- ▣ Data Dictionary View 는 DBCA로 Database 를 생성 할 때는 자동으로 수행되지만 CREATE DATABASE 라는 명령어로 수동으로 Database 를 생성할 때는 수행되지 않으므로 DBA 가 수동으로 생성해야만 함

□ Data Dictionary View

▣ Static Data Dictionary View / Dynamic Performance View

- ▣ Static Data Dictionary View : 그 속에 담고 있는 내용들이 자동으로 변경되지 않고 수동으로 변경작업을 해 주어야 한다는 뜻
- ▣ Dynamic Performance View 는 변경 내용이 바로 반영
- ▣ Static Data Dictionary view는 다시 접두어가 **DBA_** , **ALL_** , **USER_** 로 시작되는 3가지로 나뉘게 되고 Dynamic Performance View는 접두어가 **V\$**로 시작하는 것들이 대표적.

□ Static Dictionary View 종류

- ▣ **USER_** 로 시작하는 디렉터리리는 해당 사용자가 생성한 오브젝트들만 조회 할 수 있다는 뜻
- ▣ **ALL_**로 시작하는 디렉터리리는 해당 사용자가 생성한 오브젝트를 포함하여 해당 사용자가 접근 가능한 모든 오브젝트를 조회 가능
- ▣ **DBA_** 로 시작하는 디렉터리리는 데이터베이스 내의 거의 모든 오브젝트들을 다 볼수 있지만 DBA 권한을 가진 사람만이 이 디렉터리를 조회 가능

제약 조건

□ 제약조건의 종류

조건 이름	의 미
NOT NULL	이 조건이 설정 된 컬럼에는 NULL 값이 입력되지 못합니다.
UNIQUE	이 조건이 설정 된 컬럼에는 중복된 값이 입력되지 못합니다.
PRIMARY KEY	이 조건은 NOT NULL + UNIQUE 의 의미를 가지며 테이블 내에서 데이터들끼리의 유일성을 보장하는 컬럼에 설정 할 수 있으며 테이블 당 1개만 설정할 수 있습니다.
FOREIGN KEY	이 조건은 다른 테이블의 컬럼을 참조해서 무결성 검사를 하게 됩니다.
CHECK	이 조건으로 설정된 값만 입력을 허용하고 나머지는 거부됩니다.

제약조건

- 각 제약 조건의 설정 방법
 - ▣ 테이블 생성시 동시에 설정하기
 - ▣ 테이블 생성 후 추가하기

제약 조건

□ 테이블 생성시에 설정하기

```
CREATE TABLE emp3
( no NUMBER(4)
  CONSTRAINT emp3_no_pk PRIMARY KEY ,
  name VARCHAR2(10)
  CONSTRAINT emp3_name_nn NOT NULL,
  jumin VARCHAR2(13)
  CONSTRAINT emp3_jumin_nn NOT NULL
  CONSTRAINT emp3_jumin_uk UNIQUE ,
  area NUMBER(1)
  CONSTRAINT emp3_area_ck CHECK ( area < 5 ),
  deptno VARCHAR2(6)
  CONSTRAINT emp3_deptno_fk REFERENCES dept2(dcode)
);
```

□ 약식으로 생성하기

```
CREATE TABLE emp4
( no NUMBER(4) PRIMARY KEY ,
  name VARCHAR2(10) NOT NULL,
  jumin VARCHAR2(13) NOT NULL UNIQUE ,
  area NUMBER(1) CHECK ( area < 5 ),
  deptno VARCHAR2(6) REFERENCES dept2(dcode)
);
```

□ 테이블 생성 후 추가하기

```
ALTER TABLE emp4  
  ADD CONSTRAINT emp4_name_uk UNIQUE(name) ;
```

```
ALTER TABLE emp4  
  ADD CONSTRAINT emp4_area_nn NOT NULL(area) ;
```

```
ADD CONSTRAINT emp4_area_nn NOT NULL(area)  
          *
```

```
ERROR at line 2:  
ORA-00904: : invalid identifier
```

```
ALTER TABLE emp4  
  MODIFY (area constraint emp4_area_nn NOT NULL) ;
```

Table altered.

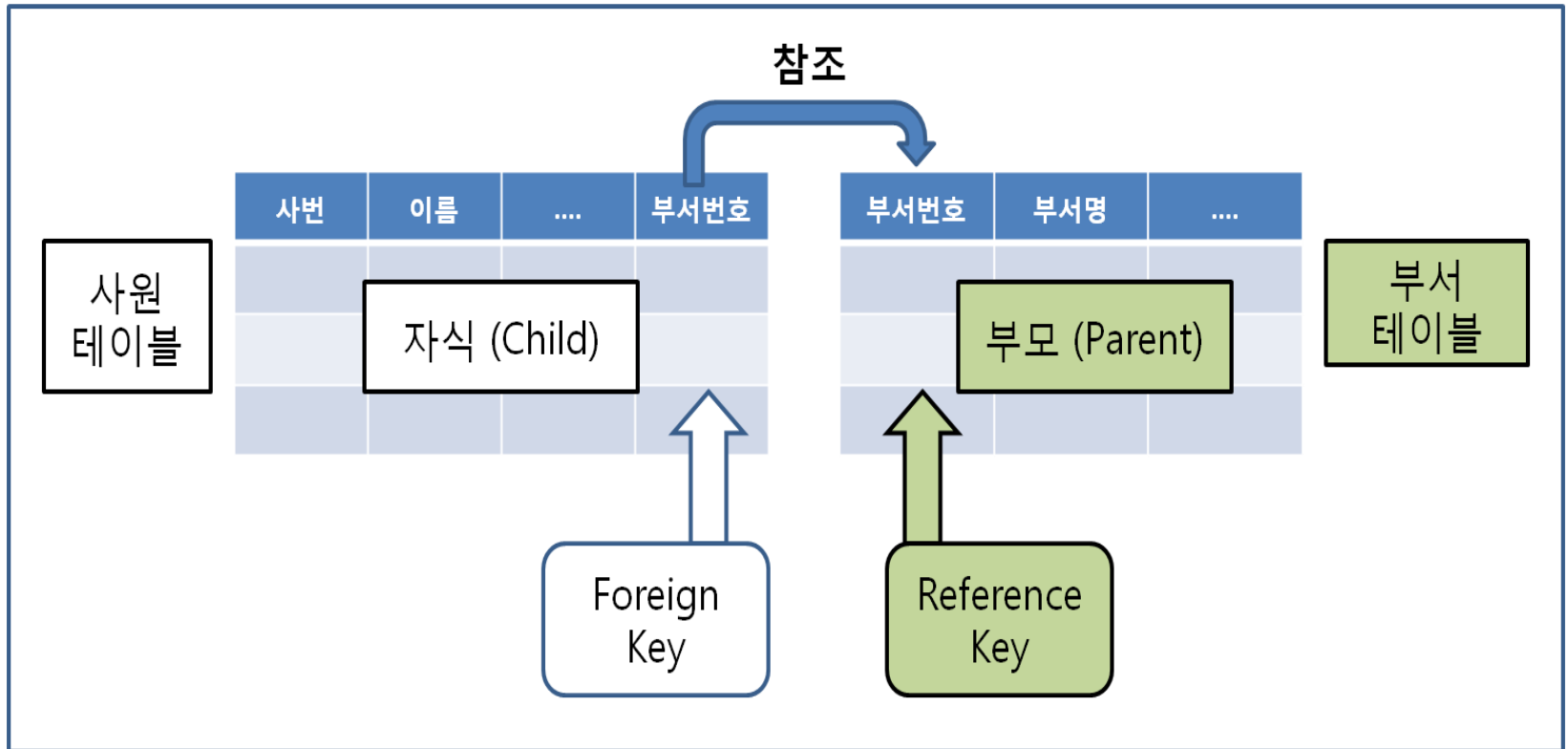
NULL <-> NOT NULL 변경방법

□ FOREIGN KEY 추가하기

- emp4 테이블의 no 컬럼이 emp2 테이블의 empno 컬럼의 값을 참조하도록 참조키 제약조건을 설정하세요.(emp4 테이블이 자식테이블입니다)

```
ALTER TABLE emp4  
  ADD CONSTRAINT emp4_no_fk FOREIGN KEY(no)  
  REFERENCES emp2(empno) ;
```

FOREIGN KEY 제약조건



□ - FOREIGN KEY 추가 시 주의사항

```
ALTER TABLE emp4
  ADD CONSTRAINT emp4_name_fk FOREIGN KEY(name)
    REFERENCES emp2(name) ;
REFERENCES emp2(name)
      *
```

ERROR at line 3:

ORA-02270: no matching unique or primary key for this column-list

```
ALTER TABLE emp2 ADD CONSTRAINT emp2_name_uk UNIQUE(name);
```

Table altered.

```
ALTER TABLE emp4
  ADD CONSTRAINT emp4_name_fk FOREIGN KEY(name)
    REFERENCES emp2(name) ;
```

Table altered.

□ ON DELETE CASCADE

- ▣ FOREIGN KEY 를 설정 후 부모 테이블 의 데이터를 지우고 싶은 데 만약 자식테이블에서 부모테이블의 해당 데이터를 참조하고 있을 경우 지울 수가 없습니다. 이럴 경우를 대비해서 FOREIGN KEY를 생성할 때 설정함.
- ▣ 이 옵션을 주면 부모테이블의 데이터가 지워지면 자식 테이블의 데이터도 함께 지우라는 의미임.

□ ON DELETE SET NULL

- ▣ 이 옵션은 부모테이블의 데이터가 지워질 경우 자식테이블의 값을 NULL 로 설정하라는 의미임.