

## DATA TYPE의 종류

### ① 문자DATA TYPE

작은유형 : CHAR, VARCHAR2

긴유형 : LONG, CLOB

### ② 숫자 DATA TYPE

NUMBER

### ③ 날짜 DATA TYPE

DATE, TIMEZONE

### ④ BINARY

작은유형: RAW

긴유형 : LONG RAW, BLOB

### ⑤ 기타 DATA TYPE

ROWID

## CHAR

**1~2000 사이의 고정길이(fixed-length character strings) 저장 방식 ,**

정의된 길이만큼 저장 공간이 할당이 되며 입력 되고 난 나머지 공간은 공백문자(BLANK-PADDING)로 채워진다.

[용도] 데이터가 항상 지정된 길이를 전부 차지 하는 경우에 유리함.

EX) SEX CHAR(1) // 남녀성별

SOCIAL\_CODE CHAR(14) // 주민번호

EX) ENAME CHAR2(20) // 'SMITH'가 입력 된 경우 20 Bytes 공간할당

## VARCHAR

VARCHAR2와 동일한 DATA TYPE, 8i 이후버전에서는 없어진 데이터타입으로 VARCHAR2 사용 권고.

## VARCHAR2

**1~4000 ,가변길이(VARIABLE-length character strings)저장방식,**

입력된 길이 만큼 저장 공간이 할당된다. (ROW 내부에 길이표시 컬럼 1~3 Bytes)

[용도] 대부분의 문자 DATA에 사용

**[장점] 데이터 저장 공간을 유용하게 사용할수 있다**

EX) ENAME VARCHAR2(20) // 'SMITH'가 입력 된 경우 5 BYTES만 공간할당

[요구]

㉔ Oracle 12c에서 CHAR,VARCHAR2의 최대길이는 ?

NUMBER

NUMBER(P,S) : 실수(floating-point-number), 정수(fixed number), 가변길이(Packed Decimal) 저장.

P(Precision) : 1~ 38(전체 유효숫자) S(Scale): 소수점이하 자리수

[실수] WEIGHT NUMBER(4,2)

[정수] SALARY NUMBER(4)

[실수,정수] DCRATE NUMBER

[권고] PRECISION,SCALE을 표시하라 → 데이터 입력시 INTEGRITY CONSTRAINTS 역할을 한다.

EX) AGE NUMBER → AGE NUMBER(3)

[주의] P를 초과시 ERROR, S를 초과시 ROUND 처리

#### NUMBER TYPE의 데이터 저장과 포맷의 관계

입력값	지정형식	저장값
1234567.89	NUMBER	1234567.89
1234567	NUMBER	1234567
1234567.89	NUMBER(9)	1234568
1234567.49	NUMBER(9)	1234567
1234567.89	NUMBER(9,2)	1234567.89
1234567	NUMBER(9,1)	1234567
1234567.89	NUMBER(9,1)	1234567.9
1234567890.89	NUMBER(9,2)	ERROR
123456749.89	NUMBER(9,-2)	123456700
123456789.89	NUMBER(9,-2)	123456800

DATE

날짜와 시간 정보를 저장하고 연산이 가능하다. (+, - )

[저장 포맷] (YY)YYMMDDHHMISS (고정길이 7bytes , packed decimal)로 저장,

[Default Date 포맷] RR-MON-YY

EX) SELECT TO\_CHAR(SYSDATE,'YYYY-MM-DD-HH24:MI:SS') FROM DUAL;

[요구]

㉕ 설치된 Oracle DBMS의 Default Date 포맷은 어떻게 결정되는가? RR-MON-YY 인가 ?

LONG
------

가변길이 문자형, 1~ 2GIGA BYTES ,VARCHAR2의 특성을 가지고 있다.

[제약]        TABLE당 1개의 LONG type정의 가능  
SELECT list절에, UPDATE의 SET 절에, INSERT의 VALUE 절에만 사용  
WHERE,GROUP BY,ORDER BY,CONNECT BY,DISTINCT에 사용(X)  
NULL,NOT NULL 제약사항(INTEGRITY CONSTRAINTS)만이 가능하다  
CANNOT BE INDEXED  
FUNCTION의 RETURN TYPE 이 될수 없다.  
SUBSTR,INSTR 사용(X)  
CREATE TABLE AS ~ , INSERT INTO SELECT ~

[요구]

- ㉠ 사용상 제약이 많은 LONG type의 대신 CLOB의 정의 및 특징에 대해 조사

RAW
-----

가변길이,RAW BINARY(GRAPHIC,SOUND,DOCUMENTS) DATA TYPE 1~2000  
CAN BE INDEXED

LONG RAW
----------

가변길이,RAW BINARY(GRAPHIC,SOUND,DOCUMENTS) DATA TYPE 1~2GIGA  
CANNOT BE INDEXED.

- \* 멀티미디어DATA를 TABLE의 한 컬럼으로 사용할수 있다. SELECT,INSERT만 가능
- \* 작은양의 멀티미디어에 사용 할 것을 권고.

[요구]

- ㉢ 사용상 제약이 많은 LONG RAW 대신 BLOB의 정의 및 특징에 대해 조사

ROWID
-------

테이블내에서도 ROWID를 저장할수있도록하는 데이터타입

[요구]

- ㉣ ROWID의 정의 및 용도 조사

DECIMAL, INTEGER, SMALL INTEGER, FLOAT
--

ANSI 호환 및 DB2등과의 TABLE 생성 SQL문장 레벨 호환성을 위해 존재 하며 실제 생성시에는 내부적으로는 ORACLE DATA TYPE으로 바뀐다.

테이블 생성(문자 DATA TYPE)
----------------------

```

① DROP TABLE CUSTOMER;           -- 기존 테이블 삭제
② CREATE TABLE CUSTOMER(          -- 고객정보 관리 테이블
    ID      VARCHAR2(8),           -- 가변길이 데이터 타입
    PWD     CHAR(8),               -- 고정길이 데이터 타입
    SEX     CHAR(1)                -- 성별 [M|F] 성별은 항상 고정된 크기의 데이터가 저장
);

③ SELECT * FROM TAB;              -- 내소유의 모든 테이블 목록 보기
④ DESC CUSTOMER                   -- 생성된 테이블의 구조(컬럼명, 데이터 타입, 컬럼길이) 확인

⑤ INSERT INTO CUSTOMER VALUES('XMAN','XMAN', 'M');           -- 컬럼명 생략
⑥ INSERT INTO CUSTOMER(ID,PWD,SEX) VALUES('ORACLE','OCM', 'F'); -- 컬럼명 지정
⑦ COMMIT;                      -- 변경사항을 DB에 반영

// INSERT시에 특정 COLUMN에 NULL값 삽입방법
// 명시적(Explicit) 방법 2가지
⑧ INSERT INTO CUSTOMER(ID,PWD,SEX) VALUES('NULL1','ZMAN',NULL); -- 'NULL' 과 차이
⑨ INSERT INTO CUSTOMER(ID,PWD,SEX) VALUES('NULL2','YMAN','');

// 암시적(Implicit) 1가지
⑩ INSERT INTO CUSTOMER(ID,PWD) VALUES('NULL3','LMAN');       -- 대상 컬럼 생략

```

고정길이와 가변길이의 차이점 비교
--------------------

```

// 데이터 저장시 차이점
⑪ SELECT ID,LENGTH(ID),VSIZE(ID),PWD,LENGTH(PWD), VSIZE(PWD) FROM CUSTOMER;
⑫ SELECT ID,REPLACE(ID, ' ', '?'),PWD,REPLACE(PWD, ' ', '?') FROM CUSTOMER;

// 비교시 차이점(BLANK-PADDING, NON-BLANK PADDING)
// VARCHAR2 TYPE 비교시에는 NON-BLANK PADDING 방식
⑬ SELECT * FROM CUSTOMER WHERE ID = 'XMAN ' ;           -- 'XMAN'
⑭ SELECT * FROM CUSTOMER WHERE ID = PWD;                -- 가변길이와 고정길이 비교
⑮ SELECT * FROM CUSTOMER WHERE ID = SUBSTR('XMAN',1,2)||SUBSTR('XMAN',3,2);

```

// CHAR TYPE 비교시에는 BLANK PADDING 방식

- ① SELECT \* FROM CUSTOMER WHERE PWD = 'XMAN ';
- ② SELECT \* FROM CUSTOMER WHERE PWD = SUBSTR('XMAN',1,2)||SUBSTR('XMAN',3,2);
- ③ SELECT \* FROM CUSTOMER  
WHERE PWD LIKE SUBSTR('XMAN',1,2)||SUBSTR('XMAN',3,2)||'%';

#### 테이블 생성(숫자 DATA TYPE)

- ④ DROP TABLE TST\_NUMBER;
- ⑤ CREATE TABLE TST\_NUMBER(  
NUM NUMBER, -- LENGTH가 없는 경우?  
AGE NUMBER(3) DEFAULT 18, -- DEFAULT  
TAX NUMBER(7,2)  
);
- ⑥ INSERT INTO TST\_NUMBER VALUES(123.5,123.5,123.5);
- ⑦ INSERT INTO TST\_NUMBER VALUES(123, 123,123);
- ⑧ INSERT INTO TST\_NUMBER VALUES(123, 12345,123); -- 정수형자리 초과시
- ⑨ INSERT INTO TST\_NUMBER VALUES(123, 123,123.56789); -- 소수점자리 초과시
- ⑩ INSERT INTO TST\_NUMBER VALUES(123, '123',123.56789);
- ⑪ INSERT INTO TST\_NUMBER(ALL\_NUM,F\_TAX) VALUES(456,456);
- ⑫ SELECT \* FROM TST\_NUMBER;

#### 테이블 생성(날짜 DATA TYPE)

- ⑬ DROP TABLE TST\_DATE;
- ⑭ CREATE TABLE TST\_DATE(  
CHAR\_HIREDATE VARCHAR2(8),  
DATE\_HIREDATE DATE,  
LOG\_DATE DATE DEFAULT SYSDATE );

[요구]

- ㉓ DEFAULT 에 TO\_CHAR(SYSDATE,'YY/MM/DD')를 사용할수 있는지 확인 하는 SQL을 작성하십시오

- ⑮ INSERT INTO TST\_DATE(CHAR\_HIREDATE,DATE\_HIREDATE)  
VALUES(TO\_CHAR(SYSDATE,'YYYYMMDD'),SYSDATE);

- ① INSERT INTO TST\_DATE(CHAR\_HIREDATE,DATE\_HIREDATE)  
VALUES('19990921',TO\_DATE('990921','YYMMDD'));
- ② SELECT \* FROM TST\_DATE;

ALTER

- ③ ALTER TABLE TST\_DATE ADD(NAME VARCHAR2(20), AGE NUMBER(3));  
DESC TST\_DATE
- ④ ALTER TABLE TST\_DATE DROP COLUMN AGE;  
DESC TST\_DATE

DROP

- SELECT \* FROM TAB;
- ⑤ DROP TABLE TST\_DATE;  
DESC TST\_DATE  
SELECT \* FROM TAB;

SUBQUERY에 의한 TABLE 생성

[용도]기존 테이블로부터 ① 테이블 구조 ② 데이터 ③ NOT NULL 제약사항을 COPY하여 새로운 테이블을 생성

- ⑥ DROP TABLE EMP\_CTAS;
- ⑦ CREATE TABLE EMP\_CTAS AS SELECT \* FROM EMP;
- ⑧ DESC EMP\_CTAS -- 테이블 구조 확인 , NOT NULL 확인(?)
- ⑨ SELECT \* FROM EMP\_CTAS; -- 데이터 확인
- ⑩ DROP TABLE EMP\_CTAS\_WHERE;
- ⑪ CREATE TABLE EMP\_CTAS\_J  
AS SELECT EMPNO,ENAME,DEPT.DNAME,SAL \* 12 AS ANNUAL\_SAL  
FROM EMP , DEPT  
WHERE EMP.DEPTNO = DEPT.DEPTNO AND EMP.DEPTNO IN (10,20);
- ⑫ DESC EMP\_CTAS\_J
- ⑬ SELECT \* FROM EMP\_CTAS\_J;

DELETE, TRUNCATE, DROP 의 차이점
------------------------------

	분류	삭제대상	저장공간
DROP	DDL	TABLE정의, DATA	공간반납
TRUNCATE	DDL	DATA(전체)	공간반납
DELETE	DML(ROLLBACK가능)	DATA (WHERE)	공간반납(X)

① CONNECT SYSTEM/MANAGER

SHOW USER

② CREATE TABLE SCOTT.LARGE\_TBL

AS

SELECT O.OWNER, O.OBJECT\_NAME, E.ENAME FROM DBA\_OBJECTS O, SCOTT.EMP E; -- 카디션 곱

③ SELECT COUNT(\*) FROM SCOTT.LARGE\_TBL; -- 대량의 SAMPLE DATA 생성

④ INSERT INTO SCOTT.LARGE\_TBL SELECT \* FROM SCOTT.LARGE\_TABLE;

-- 5~10번 실행

SELECT COUNT(\*) FROM SCOTT.LARGE\_TBL;

⑤ COMMIT;

⑥ CONN SCOTT/TIGER

SHOW USER

⑦ CREATE TABLE DROP\_TBL AS SELECT \* FROM LARGE\_TBL;

CREATE TABLE TRUNCATE\_TBL AS SELECT \* FROM LARGE\_TBL;

CREATE TABLE DELETE\_TBL AS SELECT \* FROM LARGE\_TBL;

⑧ SET TIMING ON

// DROP : TABLE & DATA 삭제, 시간CHECK

⑨ DROP TABLE DROP\_TBL;

ROLLBACK;

DESC DROP\_TBL

SELECT \* FROM DROP\_TBL;

// TRUNCATE TEST: DATA만 삭제, 시간CHECK

⑩ TRUNCATE TABLE TRUNCATE\_TBL;

ROLLBACK;

DESC TRUNCATE\_TBL

SELECT \* FROM TRUNCATE\_TBL;

// DELETE TEST: DATA만 삭제, 시간 CHECK

⑪ DELETE FROM DELETE\_TBL;

-- 시간이 오래걸리는 이유는?

ROLLBACK;

-- ROLLBACK 이 가능한 이유는?

[요구 ~CAN]

㉔ DELETE 연산이 ROLLBACK이 가능한 이유를 찾아 설명 하십시오

// ALTER

① ALTER TABLE DELETE\_TBL ADD(ADDR VARCHAR2(60));

② ALTER TABLE DELETE\_TBL ADD(AGE NUMBER(3) DEFAULT 18);

--왜 오랜 시간이?

③ SELECT \* FROM DELETE\_TBL WHERE ROWNUM < 10;

④ DROP TABLE DELETE\_TBL;

⑤ DROP TABLE TRUNCATE\_TBL;

테이블명
------

단수명, 고유성,

A~Z, a~z, 0~9, #,\_,\$, 만 허용. → T-TL

문자명으로 시작해야 한다.

대소문자 구분하지 않는다-> 대문자로 만들어진다.

EX) CREATE TABLE SCOTT.TBL\_NM(X DATE);

CREATE TABLE SCOTT.TBL\_NM(X DATE);

CREATE TABLE SCOTT.tbl\_nm(X DATE);

CREATE TABLE SCOTT.10\_TBL\_NM(X DATE);

[요구]

㉔ 테이블명과 컬럼명을 한글로 생성할 수 있는지를 확인할 수 있는 SQL예제를 작성하십시오

㉔ ㉔예제에서 작성한 테이블명 과 컬럼명을 변경하는 SQL을 예제로 작성 하십시오