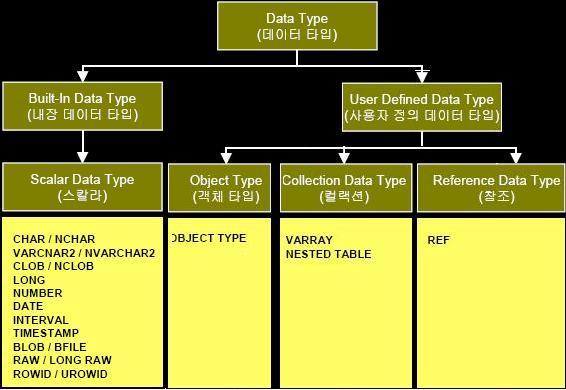
**SQL Data Type**

**I. Data & DataType**

**1. Data의 종류**

|  |  |
| --- | --- |
| Character |  |
| Numeric |  |
| Date |  |
| Binary | Multimedia   |  |  | | --- | --- | | Image | jpg, gif, bmp | | Audio | mp3, mid, wav, wma | | Moive | avi, mpg, mov, swf | | Text File | hwp, doc, txt | |

**2. Data Type의 종류**

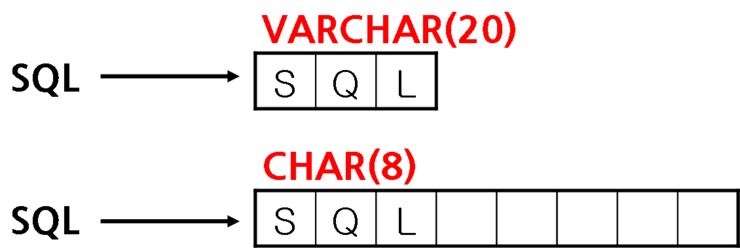


**II. Built-In Data Type (Scalar Data Type)**

**1. Character Data Type**

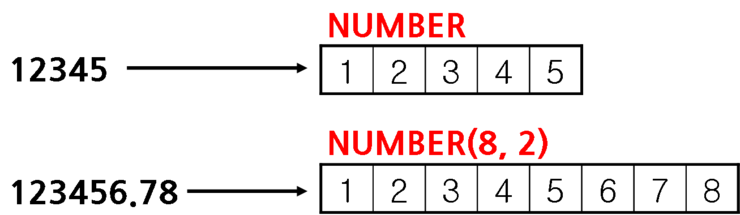
Unicode를 사용함으로써 Text 로 저장되는 모든 문자, 모든 언어를 사용 할 수 있는 장점이 있음.

|  |  |
| --- | --- |
| Char | 고정길이 문자형, 2000 byte 까지 저장 가능. |
| Varchar2 | Variable(가변)길이 문자형, 4000 byte |
| Nchar | 고정길이 문자형, 2000Byte, Unicode 문자(Transact-SQL)에 사용  예) 'column\_name' NCHAR(4)  --> 한글 4자리 문자형 Data columnn이 생성 됨 |
| Nvarchar | 가변길이 문자형, 4000Byte, Unicode에 사용 |



**2. Numeric Data Type**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number | 일반적인 고정길이 정수 , 최대 38자리 , NUMBER(precision, scale);  - precision은 소수점을 포함한 전체 자리수를 의미하며 scale은 소수점 이하 자리수  - scale생략 precision만 지정하면 소수점 이하는 반올림 되어 정수 값만 저장  -precision과 scale을 모두 생략하면 입력한 데이터 값만큼 공간이 할당   |  |  | | --- | --- | | Numeric(p, s), Decimal(p, s) | NUMBER(p, s)로 해석 | | Ingeger, Int, Smallint | NUMBER(38)로 해석 | | Float(p), Double precision | FLOAT(126)로 해석 | | Ral | FLOAT(63)로 해석 | |
| Floating-Point | 실수,NUMBER(전체자리수, 소수점이하 자리수) |



**3. Datetime Data Type**

|  |  |
| --- | --- |
| Date | 고정길이 7Byte, 세기/년/월/일/시/분/초 등을 표기하는데 사용.  표현범위 : BC 4712.01.01 ~ AD 9999.12.31  표현형식: 기본형식(YY/MM/DD) : 예) 2014년 08월 09일은 "14/08/09”  영문판 (DD-MON-YY (일-월-년도)) : 예) 2014년 12월 10일은 "10/DEC/14" |
| Timestamp | 고정길이 9Byte, 세기/년/월/일/시/분/초/도 데이터 저장, 도 단위를 6~9자리까지 저장 ①TIMESTAMP : TIMEZONE과 관련된 정보를 저장하지 않음  ②TIMESTAMP WITH TIMEZONE : TIMSTAMP Datatype에 지역시간대 추가  ③TIMESTAMP WITH LOCAL TIMEZONE : SESSION의 기본 TIMEZONE 값을 사용 |
| Interval | 년과 월에 대한 날짜의 기간 저장  ①INTERVAL YEAR(y) TO MONTH(m)  : '년/개월' <5바이트 고정 길이 사용>  ②INTERVAL DAY(d) TO SECOND(s) : '일/시/분/초' <11바이트 고정 길이 사용> |

예)

|  |  |
| --- | --- |
| DATE | 14/01/25 |
| TIMESTAMP | 14/01/25 16:20:12. 000000000 |
| TIMESTAMP WITH TIME ZONE | 14/01/25 16:20:12. 000000000 ASIA/SEOUL |
| TIMESTAMP WHIT LOCAL TIME ZONE | 14/01/25 16:20:12. 000000000 |

**4. BLOB (Binary Large OBject)**

**Binary file( graphic,audio, video)은 멀티미디어 타입**

|  |  |
| --- | --- |
| Raw | Binary Data, 2000Byte |
| Long Raw | 가변길이, **Binary** Data, 2GB |
| Long | **Character**, 가변길이, 2GB,  LOB 전에 사용되던 문자 타입, 2GB저장, LONG 대신 LOB 타입을 사용하는 것이 좋음. |

※ ① 하나의 테이블에 하나의 컬럼만 사용가능 (한 ROW에는 최대 하나의 LONG 데이터 타입만 사용 가능)

② Index Key로 사용할수 없다.

③ SQL함수나 , select문의 where, order by, group by절에 사용불가

③ Not Null이외의 제약조건을 사용할수 없다

**5. LOB (Large OBject)**

2GB를 초과하는 큰 용량의 데이터 저장시 사용 (8i부터 사용)

①SELECT, UPDATE등을 수행하기 위해서는 별도의 PL/SQL 패키지, API, DBMS\_LOB 등 사용여야 함

②한 ROW에 여러 LOB 타입 데이터 사용 가능

|  |  |
| --- | --- |
| BLOB | 가변길이 Binary file, 4GB |
| BFILE | 가변길이 **B**inary file을 외부에 저장, 4GB, BFILE에 저장된 데이터는 READ-ONLY |
| CLOB | 가변길이 **C**haracter Data, 4GB |
| NCLOB | 가변길이 u**N**icode **C**harater(다양한 국가 언어 지원), 4GB |

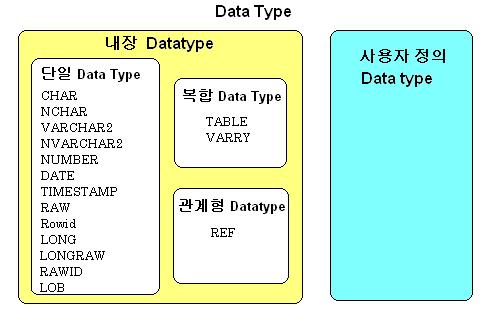
**※ BLOB & LOB 비교**

|  |  |
| --- | --- |
| **BLOB** | **LOB** |
| **RAW, LONG, LONG** | **BLOB, CLOB, NCLOB, BFILE** |
| **테이블에 하나의 컬럼만 사용가능** | **테이블에 여러 개의 컬럼 사용 가능** |
| **2GB** | **4GB** |
| **select문을 실행하면 실제 값 출력** | **select문을 실행하면 실제값이 저장된 위치 출력** |
| **DB 내부에 저장** | **DB 내부 또는 외부에 저장** |
| **분할기능 사용 불가** | **분할기능 사용 가능** |
| **데이터를 순서대로 읽는다** | **데이터를 임의대로 읽는다** |

**6. ROW**

   데이터베이스 내부 처리용 데이터 타입,  TABLE의 ROW에 대한 고유 식별자

|  |  |
| --- | --- |
| SHORT ROWID | Table row 주소, 6바이트, 16진수 문자열로 나타냄 |
| LONG ROWID | **Table row 주소,** 10바이트, 16진수 문자열로 나타냄 |
| UNIVERSAL ROWID(UROWID) | row에 대한 논리적인 주소, 3850Byte 3가지 Type로 나눠진다  ① PHYSICAL : 일반 테이블 및 클러스터, 파티션, 인덱스, 인덱스 파티션 및                                           서브파티션에 대한 주소(위치) 저장  ② LOGICAL  : Index 객체테이블의 row 주소를 기록  ③ REMOTE  : 오라클 이외 데이터베이스에 대한 ROWID |



**III. User Defined Data Type**

**1. OBJECT TYPE**

   - 사용자가 정의한 복합 데이터 타입(composite data type)으로, 데이터 타입과  이 데이터들을 조작하기 위한

함수 혹은 프로시져를 묶어서 사용자가 정의한 새로운 데이터 타입

   - OBJECT TYPE 자체를 테이블이나 컬럼으로 사용할 수 있다.

|  |  |  |
| --- | --- | --- |
| 구성요소 | ① NAME | OBJECT TYPE에 대한 고유 식별자 |
| ② ATTRIBUTE | BUILT-IN TYPE 또는 다른 OBJECT TYPE |
| ③ METHOD | : PLSQL, C, JAVA와 같은 언어로 작성된 모듈 |

**2. VARRAY**

  - 순서를 갖는 엘리먼트들의 배열을 저장

  - Scalar Type 처럼 단일 값처럼 간주 되며, DML에 의해 조작 가능

**3. NESTED TABLE**

   Table을 하나의 컬럼에 저장할 때 사용

**4. REF(Reference)**

**IV. Data Type 살펴보기**

■ 데이터베이스에 저장되는 자료 형태  
   -. 기본 데이터를 저장하기 위한 구조로 테이블을 사용한다.  
   -. 테이블은 표처럼 볼 수 있도록 로우(ROW)와 컬럼(COLUMN)으로 구성한다.



① 접속한 데이터베이스 계정에 생성된 모든 테이블 이름 조회

|  |
| --- |
| **SQL> SELECT \* FROM tab;** |
| http://cfile1.uf.tistory.com/image/13133D465097C002125140 |

**※ Data Dictionary Table**

 -. tab을 Data Dictionary Table 라 함.

 -. sys, system 사용자들이 소유.

 -. DB가 자기를 관리하는데 사용하는 시스템 카탈로그 제공.

 -. 오라클에서 제공되는 카탈로그는 스크립트에 의해 생성.

 -. DB 업그레이드, 설치, 기능 추가 시 카탈로그 스크립트를 실행하여 딕셔너리 테이블을 업데이트 해야 한다.

■ 데이터 구조를 살펴보기 위한 DESC

|  |
| --- |
| **DESC[RIBE] 테이블 명** |

  DESC[RIBE] 명령어는 컬럼 이름, 데이터 형, 길이와  NULL 허용 유무 등과 같은 특정 테이블의 정보알려준다.

**※ emp, deptno 테이블은 오라클을 설치하게 되면 제공되는 실습용 테이블 이다.**

① emp 테이블의 구조

|  |
| --- |
| **SQL> DESC emp;** |
| http://cfile1.uf.tistory.com/image/150959365097C6622FFF87 |

위 결과를 정리하면 아래와 같다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 사원번호 | 사원명 | 직급 | 해당관리자(상관) | 입사일자 | 급여 | 커미션 | 부서번호 |

② dept 테이블의 구조

|  |
| --- |
| **SQL> DESC dept;** |
| http://cfile10.uf.tistory.com/image/121B56355097C8E50E8F5B |

위 결과를 정리하면 아래와 같다.

|  |  |  |
| --- | --- | --- |
| **DEPTNO** | **DNAME** | **LOC** |
| 부서번호 | 부서명 | 지역 |

**※ NULL의 의미**

 -. 한 로우의 어떤 컬럼의 값이 정해지지 않았을 때, 이를 NULL값 이라고 한다.

 -. NULL은 할당 받지 않은 값, 모르는 값, 정해지지 않은 값을 의미한다.

 -. 0(숫자의 한자리)이나, 스페이스(문자의 한자리)와는 다른 값이다.

**[출처]** [Oracle Sql Data Type](http://blog.naver.com/artmagic/30082800872)|**작성자** [오락클](http://blog.naver.com/artmagic)