ORACLE基本数据类型（亦叫内置数据类型 built-in datatypes)可以按类型分为：字符串类型、数字类型、日期类型、LOB类型、LONG RAW& RAW类型、ROWID & UROWID类型。

在讲叙字符串类型前，先要讲一下编码。字符串类型的数据可依编码方式分成数据库字符集（CHAR/VARCHAR2/CLOB/LONG)和国际字符集(NCHAR/NVARCHAR2/NCLOB)两种。数据库中的字符串数据都通过字符集将字符转换为数字后(二进制)，才存储到数据块中。通过不同的编码集转换，即便是相同的字符，也可能会转换成不同的二进制编码。这也是产生乱码的原因。数据库的编码格式一般是在创建数据库时指定的。当然也可以修改数据库的编码。

查看数据库视图所包含的数据类型:SELECT \* FROM DBA\_TYPES WHERE OWNER IS NULL.具体细节情况参见**Oracle® Database**SQL Language Quick Reference 10/11*g 或官方文档*

**一 字符串类型**

    字符串数据类型还可以依据存储空间分为固定长度类型（CHAR/NCHAR) 和可变长度类型（VARCHAR2/NVARCHAR2)两种.

    所谓固定长度：是指虽然输入的字段值小于该字段的限制长度，但是实际存储数据时，会先自动向右补足空格后，才将字段值的内容存储到数据块中。这种方式虽然比较浪费空间，但是存储效率较可变长度类型要好。同时还能减少数据行迁移情况发生。

所谓可变长度：是指当输入的字段值小于该字段的限制长度时，直接将字段值的内容存储到数据块中，而不会补上空白，这样可以节省数据块空间。

**1.1：CHAR类型 CHAR(size [BYTE | CHAR])**

CHAR类型，定长字符串，会用空格填充来达到其最大长度。非NULL的CHAR（12）总是包*含12字节信息。CHAR字段最多可以存储2,000字节的信息。如果创建表时，不指定CHAR长度，则默认为1。另外你可以指定它存储字节或字符，例如 CHAR(12 BYTYE) CHAR(12 CHAR).一般*来说默认是存储字节，你可以查看数据库参数

NLS\_LENGTH\_SEMANTICS的值。

**SQL Code**

* SQL> show parameter nls\_length\_semantics;
* NAME                   TYPE           VALUE
* ------------------   ----------- -----------------
* nls\_length\_semantics   string          BYTE
* eg:
* CREATE TABLE TEST
* (
* NAME\_OLD CHAR(10),
* NAME\_NEW CHAR(10)
* )
* INSERT INTO TEST
* ( NAME\_OLD, NAME\_NEW)
* SELECT 'ABCDEFGHIJ' , '你清除字节与字符' FROM DUAL;
* COMMIT;
* INSERT INTO TEST
* ( NAME\_OLD, NAME\_NEW)
* SELECT '你清除字节与字符' , 'ABCDEFGHIJ' FROM DUAL;
* ORA-12899: 列 "SYS"."TEST"."NAME\_OLD" 的值太大 (实际值: 24, 最大值: 10)

注意：数据库的NLS\_CHARACTERSET 为AL32UTF8，即一个汉字占用三到四个字节。如果NLS\_CHARACTERSET为ZHS16GBK，则一个字符占用两个字节。

如果串的长度小于或等于250（0x01~0xFA）， Oracle 会使用1 个字节来表示长度。对于所有长度超过250 的串，都会在一个标志字节0xFE 后跟有两个字节来表示长度。因此，如果有一个包含“Hello World”的VARCHAR2(80)，则在块中可能如图12.-1 所示



**1.2： NCHAR类型**

这是一个包含UNICODE格式数据的定长字符串。NCHAR字段最多可以存储2,000字节的信息。它的最大长度取决于国家字符集。另外查询时，如果字段是NCHAR类型，则需要如下书写

SELECT translated\_description FROM product\_descriptions

WHERE translated\_name = N'LCD Monitor 11/PM';

**1.3 VARCHAR类型**

不要使用VARCHAR数据类型。使用VARCHAR2数据类型。虽然VARCHAR数据类型目前是VARCHAR2的同义词，VARCHAR数据类型将计划被重新定义为一个单独的数据类型用于可变长度的字符串相比，具有不同的比较语义。

**1.4： VARCHAR2类型**

变长字符串，与CHAR类型不同，它不会使用空格填充至最大长度。VARCHAR2最多可以存储4,000字节的信息。

**1.5： NVARCHAR2类型**

这是一个包含UNICODE格式数据的变长字符串。 NVARCHAR2最多可以存储4,000字节的信息。

**二. 数字类型**

**2.1 NUMBER类型**

NUMBER(P,S)是最常见的数字类型，可以存放数据范围为10^130~10^126（不包含此值)，需要1~22字节(BYTE)不等的存储空间。

P 是Precison的英文缩写，即精度缩写，表示有效数字的位数，最多不能超过38个有效数字

S是Scale的英文缩写，可以使用的范围为-84~127。Scale为正数时，表示从小数点到最低有效数字的位数，它为负数时，表示从最大有效数字到小数点的位数

下面是官方文档的示例

|  |  |  |
| --- | --- | --- |
| **Actual Data** | **Specified As** | **Stored As** |
| 123.89 | NUMBER | 123.89 |
| 123.89 | NUMBER(3) | 124 |
| 123.89 | NUMBER(6,2) | 123.89 |
| 123.89 | NUMBER(6,1) | 123.9 |
| 123.89 | NUMBER(3) | 124 |
| 123.89 | NUMBER(4,2) | exceeds precision |
| 123.89 | NUMBER(6,-2) | 100 |
| .01234 | NUMBER(4,5) | .01234 |
| .00012 | NUMBER(4,5) | .00012 |
| .000127 | NUMBER(4,5) | .00013 |
| .0000012 | NUMBER(2,7) | .0000012 |
| .00000123 | NUMBER(2,7) | .0000012 |
| 1.2e-4 | NUMBER(2,5) | 0.00012 |
| 1.2e-5 | NUMBER(2,5) | 0.00001 |

**2.2 INTEGER类型**

INTEGER是NUMBER的子类型，它等同于NUMBER（38,0），用来存储整数。若插入、更新的数值有小数，则会被四舍五入。

例如：

CREATE TABLE TEST

(

ID INTEGER

)

查看表TEST的DDL定义如下所示

CREATE TABLE "SYS"."TEST"

( "ID" NUMBER(\*,0)

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

INSERT INTO TEST

SELECT 12.34 FROM DUAL;

INSERT INTO TEST

SELECT 12.56 FROM DUAL;

SQL> SELECT \* FROM TEST;

ID

----------

12

13

**2.3 浮点数**

浮点数可以有一个十进制数点任何地方从第一个到最后一个数字，或者可以在所有有没有小数点。指数可能（可选） 用于以下数量增加的范围 （例如， 1.777e-20)。刻度值不适用于浮点数字，因为可以显示在小数点后的位数的数量不受限制。

二进制浮点数不同数量的值由 Oracle 数据库内部存储的方式。使用小数精度数存储值。完全相同号码存储范围和数量由支持的精度内的所有文本。正是因为使用小数精度（数字 0 到 9） 表示文本存储文本。使用二进制精度 （数字 0 和 1） 存储二进制浮点数。这种存储方案不能代表所有确切地使用小数精度的值。频繁地，将值从十进制转换为二进制的精度时出现的错误时撤消值回从二进制转换为十进制精度。在字面 0.1 是一个这样的例子。

Oracle 数据库提供了专为浮点数的两种数值数据类型：

**BINARY\_FLOAT**

BINARY\_FLOAT 是 32 位、 单精度浮点数字数据类型。可以支持至少6位精度,每个 BINARY\_FLOAT 的值需要 5 个字节，包括长度字节。

**BINARY\_DOUBLE**

BINARY\_DOUBLE 是为 64 位，双精度浮点数字数据类型。每个 BINARY\_DOUBLE 的值需要 9 个字节，包括长度字节。

在数字的列中，浮点数有小数精度。在 BINARY\_FLOAT 或 BINARY\_DOUBLE 的列中，浮点数有二进制的精度。二进制浮点数支持的特殊值无穷大和 NaN （不是数字）。

您可以指定列在表 2-4 范围内的浮点数。"数字文本"中定义了用于指定浮点数的格式。

Table 2-3 Floating Point Number Limits

|  |  |  |
| --- | --- | --- |
| **Value** | **Binary-Float** | **Binary-Double** |
| Maximum positive finite value | 3.40282E+38F | 1.79769313486231E+308 |
| Minimum positive finite value | 1.17549E-38F | 2.22507485850720E-308 |

**2.5 FLOAT类型**

FLOAT类型也是NUMBER的子类型。

Float(n),数 n 指示位的精度，可以存储的值的数目。N 值的范围可以从 1 到 126。若要从二进制转换为十进制的精度，请将 n 乘以 0.30103。要从十进制转换为二进制的精度，请用 3.32193 乘小数精度。126 位二进制精度的最大值是大约相当于 38 位小数精度。

**三.** **日期类型**

日期类型用于存储日期数据，但是并不是使用一般的格式（2012-08-08）直接存储到数据库的。

**3.1 DATE类型**

DATE是最常用的数据类型，日期数据类型存储日期和时间信息。虽然可以用字符或数字类型表示日期和时间信息，但是日期数据类型具有特殊关联的属性。为每个日期值，Oracle 存储以下信息： 世纪、 年、 月、 日期、 小时、 分钟和秒。一般占用7个字节的存储空间。

**3.2 TIMESTAMP类型**

这是一个7字节或12字节的定宽日期/时间数据类型。它与DATE数据类型不同，因为TIMESTAMP可以包含小数秒，带小数秒的TIMESTAMP在小数点右边最多可以保留9位

**3.3 TIMESTAMP WITH TIME ZONE类型**

这是TIMESTAMP类型的变种，它包含了时区偏移量的值

**3.4 TIMESTAMP WITH LOCAL TIME ZONE类型**

**3.5 INTERVAL YEAR TO MOTH**

**3.6 INTERVAL DAY TO SECOND**

**四.** **LOB类型**

内置的LOB数据类型包括BLOB、CLOB、NCLOB、BFILE（外部存储）的大型化和非结构化数据，如文本、图像、视屏、空间数据存储。BLOB、CLOB、NCLOB类型

**4.1 CLOB 数据类型**

   它存储单字节和多字节字符数据。支持固定宽度和可变宽度的字符集。CLOB对象可以存储最多 (4 gigabytes-1) \* (database block size) 大小的字符

**4.2 NCLOB 数据类型**

   它存储UNICODE类型的数据，支持固定宽度和可变宽度的字符集，NCLOB对象可以存储最多(4 gigabytes-1) \* (database block size)大小的文本数据。

**4.3 BLOB 数据类型**

   它存储非结构化的二进制数据大对象，它可以被认为是没有字符集语义的比特流，一般是图像、声音、视频等文件。BLOB对象最多存储(4 gigabytes-1) \* (database block size)的二进制数据。

**4.4 BFILE 数据类型**

二进制文件，存储在数据库外的系统文件，只读的，数据库会将该文件当二进制文件处理

**五.** **RAW & LONG RAW类型**

**5.1 LONG类型**

它存储变长字符串，最多达2G的字符数据（2GB是指2千兆字节， 而不是2千兆字符），与VARCHAR2 或CHAR 类型一样，存储在LONG 类型中的文本要进行字符集转换。ORACLE建议开发中使用CLOB替代LONG类型。支持LONG 列只是为了保证向后兼容性。CLOB类型比LONG类型的限制要少得多。 LONG类型的限制如下：

1.一个表中只有一列可以为LONG型。(Why?有些不明白)

2.LONG列不能定义为主键或唯一约束，

3.不能建立索引

4.LONG数据不能指定正则表达式。

5.函数或存储过程不能接受LONG数据类型的参数。

6.LONG列不能出现在WHERE子句或完整性约束（除了可能会出现NULL和NOT NULL约束）

官方文档描叙如下：

The use of LONG values is subject to these restrictions:

A table can contain only one LONG column.

You cannot create an object type with a LONG attribute.

LONG columns cannot appear in WHERE clauses or in integrity constraints (except that they can appear in NULL and NOT NULL constraints).

LONG columns cannot be indexed.

LONG data cannot be specified in regular expressions.

A stored function cannot return a LONG value.

You can declare a variable or argument of a PL/SQL program unit using the LONG datatype. However, you cannot then call the program unit from SQL.

Within a single SQL statement, all LONG columns, updated tables, and locked tables must be located on the same database.

LONG and LONG RAW columns cannot be used in distributed SQL statements and cannot be replicated.

If a table has both LONG and LOB columns, then you cannot bind more than 4000 bytes of data to both the LONG and LOB columns in the same SQL statement. However, you can bind more than 4000 bytes of data to either the LONG or the LOB column.

In addition, LONG columns cannot appear in these parts of SQL statements:

GROUP BY clauses, ORDER BY clauses, or CONNECT BY clauses or with the DISTINCT operator in SELECT statements

The UNIQUE operator of a SELECT statement

The column list of a CREATE CLUSTER statement

The CLUSTER clause of a CREATE MATERIALIZED VIEW statement

SQL built-in functions, expressions, or conditions

SELECT lists of queries containing GROUP BY clauses

SELECT lists of subqueries or queries combined by the UNION, INTERSECT, or MINUS set operators

SELECT lists of CREATE TABLE ... AS SELECT statements

ALTER TABLE ... MOVE statements

SELECT lists in subqueries in INSERT statements

**5.2 LONG RAW 类型**，能存储2GB 的原始二进制数据（不用进行字符集转换的数据）

**5.3 RAW类型**

用于存储二进制或字符类型数据，变长二进制数据类型，这说明采用这种数据类型存储的数据不会发生字符集转换。这种类型最多可以存储2,000字节的信息

**六.** **ROWID & UROWID类型**

在数据库中的每一行都有一个地址。然而，一些表行的地址不是物理或永久的，或者不是ORACLE数据库生成的。

例如，索引组织表行地址存储在索引的叶子，可以移动。

例如,外部表的ROWID（如通过网关访问DB2表）不是​​标准的ORACLE的rowid。

ORACLE使用通用的ROWID（UROWIDs）的存储地址的索引组织表和外表。索引组织表有逻辑urowids的，和国外表的外urowids，。UROWID这两种类型的存储在ROWID伪（堆组织的表的物理行id）。

创建基于逻辑的rowid在表中的主键。逻辑的rowid不会改变，只要主键不改变。索引组织表的ROWID伪UROWID数据类型。你可以访问这个伪列，你会堆组织表的ROWID伪（即使用一个SELECT ...ROWID语句）。如果你想存储的rowid索引组织表，那么你就可以定义一列的表型UROWID到列检索值的ROWID伪。