Oracle数据库字段类型说明

# 字符串类型

## Char

char数据类型存储固定长度的字符值。一个CHAR数据类型可以包括1到2000个字符。如果对CHAR没有明确地说明长度，它的默认长度则设置为1。 如果对某个CHAR类型变量赋值，其长度小于规定的长度，那么Oracle自动用空格填充。

注意：如果给了一个比固定长度更长的值，其尾部带有空格，则空格会被删除到固定的长度。如果值太大Oracle会返回一个错误，

## Varchar

用于保存变长的字符串数据。其中最大字节长度由(size)指定。每行长度可变，最大长度为每行4000字节。设置长度(size)前需考虑字符集为单字节或多字节。

注意：varchar是被废弃的类型，oracle目前都推荐使用varchar2。虽然varchar也可以用，但不确保未来的版本是否还支持

## Varchar2

用于保存变长的字符串数据。其中最大字节长度由(size)指定。每行长度可变，最大长度为每行4000字节。设置长度(size)前需考虑字符集为单字节或多字节。由于VARCHAR2数据类型只存储为该列所赋的字符（不加空格），所以VARCHAR2需要的存储空间比CHAR数据类型要小。

区别：varchar2把所有字符都占两字节处理(一般情况下)，varchar只对汉字和全角等字符占两字节，数字，英文字符等都是一个字节；VARCHAR2把空串等同于null处理，而varchar仍按照空串处理；大部分情况下建议使用varchar2类型，可以保证更好的兼容性。

## Nchar

Unicode数据类型，根据字符集而定的固定长度字符串，最大长度2000bytes。

## Nvarchar2

Unicode数据类型，根据字符集而定的可变长度字符串，最大长度4000bytes。

## Long

可变长字符列，最大长度限制为2GB，用于不需要作字符串搜索的长串数据。

说明：此类型是一个遗留下来的而且将来不会被支持的数据类型，逐渐被BLOB，CLOB，NCLOB等大的数据类型所取代。

# 数字类型

## Number

NUMBER数据类型精度可以高达38位，它有两个限定符，如：column NUMBER(precision,scale)。precision表示数字中的有效位。如果没有指定precision的话，Oracle将使用38 作为精度。scale表示小数点右边的位数，scale默认设置为0。如果把scale设成负数，Oracle将把该数字取舍到小数点左边的指定位数。

## Decimal

Oracle只是在语法上支持decimal类型，但是在底层实际上它就是number类型，支持decimal类型是为了能把数据从Oracle数据库移到其他数据库中。

## Float

FLOAT类型也是NUMBER的子类型。其格式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。

## BINARY\_FLOAT

BINARY\_FLOAT 是一种 32 位，单精度浮点数字数据类型。每个 BINARY\_FLOAT 值需要5 字节存储空间，其中 1 字节用于存储数据值的长度。支持 NUMBER 数据类型所提供的基本功能。但采用二进制精度，而NUMBER 采用十进制精度。因此能够提供更快的数学运算速度，且能减少占用的存储空间。

## BINARY\_DOUBLE

BINARY\_DOUBLE 是一种 64 位，双精度浮点数字数据类型。每个 BINARY\_DOUBLE 值需要9 字节存储空间，其中 1 字节用于存储数据值的长度。支持 NUMBER 数据类型所提供的基本功能。但采用二进制精度，而NUMBER 采用十进制精度。因此能够提供更快的数学运算速度，且能减少占用的存储空间。

# 日期数据类型

## Date

ORACLE最常用的日期类型，它可以保存日期和时间，常用日期处理都可以采用这种类型。DATE表示的日期范围可以是公元前4712年1月1日至公元9999年12月31日。

date类型在数据库中的存储固定为7个字节， 第1字节：世纪+100、 第2字节：年、第3字节：月、第4字节：天、第5字节：小时+1、第6字节：分+1、第7字节：秒+1。

## Timestamp

ORACLE常用的日期类型，它与date的区别是不仅可以保存日期和时间，还能保存小数秒，小数位数可以指定为0-9，默认为6位，所以最高精度 可以到ns(纳秒)，数据库内部用7或者11个字节存储，如果精度为0，则用7字节存储，与date类型功能相同，如果精度大于0则用11字节存储。第1字节：世纪+100、第2字节：年、第3字节：月、第4字节：天、 第5字节：小时+1、第6字节：分+1、第7字节：秒+1、 第8-11字节：纳秒，采用4个字节存储，内部运算类型为整形

注：TIMESTAMP日期类型如果与数值进行加减运算会自动转换为DATE型，也就是说小数秒会自动去除。

**TIMESTAMP WITH TIME ZONE**

对TIMESTAMP进行了扩展，用于存储时区。时间戳以及时区位移值，其中fractional\_seconds\_precision是数字在第二日期时间字段的小数部分数字的所有值。可接受的值是0到9。默认是6。默认格式是确定明确的NLS\_DATE\_FORMAT参数或隐式的NLS\_TERRITORY参数。大小固定为13字节。此数据类型包含日期时间字段YEAR，MONTH，日，小时，分钟，秒TIMEZONE\_HOUR和TIMEZONE\_MINUTE。它有一个明确的分数秒和时区。

**TIMESTAMP WITH LOCAL TIME ZONE**

所有的TIMESTAMP WITH TIME ZONE值，但下列情况除外：数据标准化数据库的时区时，存储在数据库中。当数据被检索，用户可以看到在会话时区中的数据。

默认格式是确定明确的NLS\_DATE\_FORMAT参数或隐式的NLS\_TERRITORY参数。的大小不同的7至11个字节，取决于精度。

**INTERVAL YEAR TO MONTH**

存储期间年数和月的时间，其中year\_precision是数字的年份日期时间字段的数量。可接受的值是0到9。默认是2。大小固定为5个字节。

**INTERVAL DAY TO SECOND**

其格式为：INTERVAL DAY [(day\_precision)] TO SECOND [(fractional\_seconds)]，是存储一段时间以天，小时，分钟和秒，其中day\_precision是数字在DAY日期时间字段的最大数量。可接受的值是0到9。默认是2。fractional\_seconds\_precision是数字中的第二个字段的小数部分的数量。可接受的值是0到9。默认是6。大小固定为11个字节。

**INTERVAL YEAR TO MONTH**

Oracle语法：INTERVAL YEAR [（year\_precision ）] TO MONTH，用来表示一段时间差, 只精确到年和月. year\_precision是数字年的时间段，接受的值为0到9。默认值是2。大小固定为5个字节。

# 大型对象（LOB）数据类型

**Clob**

最大4G，存储单字节字符型数据。适用于存储超长文本。

**Nclob**

最大4G，存储多字节国家字符型数据。适用于存储超长文本。

**Blob**

最大4G，存储二进制数据。适用于存储图像、视频、音频等。

**BFile**

最大长度是4GB，在数据库外部保存的大型二进制对象文件，最大长度是4GB。这种外部的LOB类型，通过数据库记录变化情况，但是数据的具体保存是在数据库外部进行的。 Oracle 可以读取、查询BFILE，但是不能写入，不参与事务。

**RAW和LONG RAW数据类型**

## RAW

raw(n)格式，其中n=1 to 2000，可变长二进制数据，在具体定义字段的时候必须指明最大长度n，Oracle 用这种格式来保存较小的图形文件或带格式的文本文件，如Miceosoft Word文档。raw是一种较老的数据类型，将来会逐渐被BLOB、CLOB、NCLOB等大的对象数据类型所取代。

## LONG RAW

最大长度是2GB。可变长二进制数据，最大长度是2GB。Oracle 用这种格式来保存较大的图形文件或带格式的文本文件，如Miceosoft Word文档，以及音频、视频等非文本文件。在同一张表中不能同时有long类型和long raw类型，long raw也是一种较老的数据类型，将来会逐渐被BLOB、CLOB、NCLOB等大的对象数据类型所取代。

# ROWID和UROWID数据类型

## Rowid

ROWID为该表行的唯一标识，是一个伪列，可以用在SELECT中，但不可以用INSERT, UPDATE来修改该值。ROWID列，Oracle使用ROWID列来建立内部索引。你可以引用ROWID的值，但ROWID并不存放在数据库中，你可以创建一个表包含ROWID数据类型。但Oracle不保证该值是合法的rowids。用户必须确保该rowid值是真实合法的。

## Urowid

UROWID支持逻辑和物理的rowids，列UROWID类型可以存储各种rowids, 从8.1以后的Oracle才有UROWID类型，它也可以用来保存以前的ROWID类型数据信息。

# 特殊数据类型

## Xmltype

XMLType是Oracle从9i开始特有的数据类型，是一个继承了Blob的强大存在，可以用来存储xml并提供了相当多的操作函数，用来直接读取xml文档和管理节点。理论上可以保存2G大小的数据。作为XML数据的特殊存储类型。Xmltype提供了适合的保存、检索和操作的支持。

## Xmlindex

XMLIndex 索引可用于 SQL/XML 函数 XMLExists()、XMLTable() 和 XMLQuery()，而且它在查询的任何部分都有效；它并不限于用在 WHERE 子句中。而您可能用于 XML 数据的其他任何索引都无法做到这一点。

XMLIndex 因此可提高对 SELECT 列表数据和 FROM 列表数据的访问速度，对 XML 片段提取尤为有用。基于函数的索引和 CTXXPath 索引。您无需预先了解将在查询中使用的XPath 表达式。XMLIndex 完全是通用的，而基于函数的索引则不同。可将 XMLIndex 索引用于基于 XML 模式或非基于模式的数据。它可用于二进制 XML 和非结构化存储模型。B 树索引仅适用于以对象关系方式存储（结构化存储）的基于模式的数据；它对于存储在二进制XML 或CLOB 实例中的基于 XML 模式的数据效率低下。可使用 XMLIndex 索引通过 XPath 表达式进行搜索，这些表达式的目标是集合（即文档中出现多次的节点）。而函数索引则不然。