

PLSQL单行函数和组函数详解 PDF转换可能丢失图片或格式  
，建议阅读原文

[https://www.100test.com/kao\\_ti2020/143/2021\\_2022\\_PLSQL\\_E5\\_8](https://www.100test.com/kao_ti2020/143/2021_2022_PLSQL_E5_8)

D\_95\_E8\_A1\_c102\_143071.htm 函数是一种有零个或多个参数并且有一个返回值的程序。在SQL中Oracle内建了一系列函数，这些函数都可被称为SQL或PL/SQL语句，函数主要分为两大类：单行函数、组函数。

1、SQL中的单行函数 SQL和PL/SQL中自带很多类型的函数，有字符、数字、日期、转换、和混合型等多种函数用于处理单行数据，因此这些都可被统称为单行函数。这些函数均可用于SELECT,WHERE、ORDER BY等子句中，例如下面的例子中就包含了

了TO\_CHAR,UPPER,SOUNDEX等单行函数。SELECT ename,TO\_CHAR(hiredate,day,DD-Mon-YYYY)FROM emp Where UPPER(ename) Like AL%ORDER BY SOUNDEX(ename)

单行函数也可以在其他语句中使用，如update的SET子句，INSERT的VALUES子句，DELETE的WHERE子句,认证考试特别注意在SELECT语句中使用这些函数，所以我们的注意力也集中在SELECT语句中。 NULL和单行函数 在如何理

解NULL上开始是很困难的，就算是一个很有经验的人依然对此感到困惑。NULL值表示一个未知数据或者一个空值，算术操作符的任何一个操作数为NULL值，结果均为提个NULL值，这个规则也适合很多函数，只

有CONCAT,DECODE,DUMP,NVL,REPLACE在调用了NULL参数时能够返回非NULL值。在这些中NVL函数时最重要的，因为他能直接处理NULL值，NVL有两个参数：NVL(x1,x2),x1和x2都是表达式，当x1为null时返回X2,否则返回x1。下面我

们看看emp数据表它包含了薪水、奖金两项，需要计算总的补偿

column name	emp_id	salary	bonus	key type	pk	nulls/unique
nn,u nnfk	table	datatype	number	number	numberlength	11.2 11.2

不是简单的将薪水和奖金加起来就可以了，如果某一行是null值那么结果就将是null，比如下面的例子：

```
0update emp set salary=(salary bonus)*1.1
```

这个语句中，雇员的工资和奖金都将更新为一个新的值，但是如果没有奖金，即 salary null,那么就会得出错误的结论，这个时候就要使用nvl函数来排除null值的影响。所以正确的语句是：

```
0update emp set salary=(salary nvl(bonus,0))*1.1
```

**单行字符串函数** 单行字符串函数用于操作字符串数据，他们大多数有一个或多个参数，其中绝大多数返回字符串

**ASCII()** c1是一字符串，返回c1第一个字母的ASCII码，他的逆函数是**CHR()**

```
SELECT ASCII(A) BIG_A,ASCII(z) BIG_z FROM emp
```

BIG\_A BIG\_z  
65 122  
CHR( < i > ) [NCHAR\_CS] i是一个数字，函数返回十进制表示的字符

```
0select CHR(65),CHR(122),CHR(223) FROM emp
```

CHR65  
CHR122  
CHR223

**A z B CONCAT(,)** c1,c2均为字符串，函数将c2连接到c1的后面，如果c1为null,将返回c2.如果c2为null,则返回c1，如果c1、c2都为null，则返回null。他和操作符||返回的结果相同

```
0select concat(slobo ,Svoboda) username from dual
```

username  
slobo Svoboda

**INITCAP()** c1为一字符串。函数将每个单词的第一个字母大写其它字母小写返回。单词由空格，控制字符，标点符号限制。

```
0select INITCAP(veni,vedi,vici) Ceasar from dual
```

Ceasar  
Veni,Vedi,Vici

**INSTR(,[, < i > [,])** c1,c2均为字符串，i,j为整数。函数返回c2在c1中第j次出现的位置，搜索从c1的第i个字符开始。当没有发现需要的字符时返回0,

如果i为负数，那么搜索将从右到左进行，但是位置的计算还是从左到右，i和j的缺省值为1。0select INSTR(Mississippi,i,3,3) from dualINSTR(MISSISSIPPI,I,3,3)110select INSTR(Mississippi,i,-2,3) from dualINSTR(MISSISSIPPI,I,3,3)2

**INSTRB(,[i],[j])** 与INSTR( )函数一样，只是他返回的是字节，对于单字节INSTRB()等于INSTR() **LENGTH()** c1为字符串，返回c1的长度，如果c1为null，那么将返回null值。0select LENGTH(Ipso Facto) ergo from dualergo10 **LENGTHb()** 与LENGTH()一样，返回字节。 **lower()** 返回c的小写字符，经常出现在where子串中0select LOWER(colorname) from itemdetail WHERE LOWER(colorname) LIKE %white%**COLORNAME**Winterwhite **LPAD(, < i > [,])** c1,c2均为字符串，i为整数。在c1的左侧用c2字符串补足致长度i,可多次重复，如果i小于c1的长度，那么只返回i那么长的c1字符，其他的将被截去。c2的缺省值为单空格，参见RPAD。0select LPAD(answer,7,) padded,answer unpadded from question.**PADDED UNPADDED** Yes YesNO NOMaybe maybe **LTRIM(,)** 把c1中最左边的字符去掉，使其第一个字符不在c2中，如果没有c2，那么c1就不会改变。0select LTRIM(Mississippi,Mis) from dual**LTRppi** **RPAD(, < i > [,])** 在c1的右侧用c2字符串补足致长度i,可多次重复，如果i小于c1的长度，那么只返回i那么长的c1字符，其他的将被截去。c2的缺省值为单空格,其他与LPAD相似 **RTRIM(,)** 把c1中最右边的字符去掉，使其第后一个字符不在c2中，如果没有c2，那么c1就不会改变。 **REPLACE(,[,])** c1,c2,c3都是字符串，函数用c3代替出现在c1中的c2后返回。0select REPLACE(uptown,up,down)

from dual REPLACE downtown STBSTR(, < i > [,]) c1为一字符串  
 , i,j为整数 , 从c1的第i位开始返回长度为j的子字符串 , 如果j  
 为空 , 则直到串的尾部。 0select SUBSTR(Message,1,4) from  
 dual SUBSMess SUBSTRB(, < i > [,]) 与SUBSTR大致相同 , 只  
 是I,J是以字节计算。 SOUNDEX() 返回与c1发音相似的  
 词 0select SOUNDEX(dawes) Dawes SOUNDEX(daws) Daws,  
 SOUNDEX(dawson) from dual Dawes Daws Dawson D200 D200  
 D250 TRANSLATE(,,) 将c1中与c2相同的字符以c3代替 0select  
 TRANSLATE(fumble,uf,ar) test from dual TEXT ramble TRIM([[ ]]  
 from c3) 将c3串中的第一个 , 最后一个 , 或者都删除。 0select  
 TRIM( space padded ) trim from dual TRIMspace padded  
 UPPER() 返回c1的大写 , 常出现where子串中 0select name from  
 dual where UPPER(name) LIKE KI%NAMEKING 单行数字函数  
 单行数字函数操作数字数据 , 执行数学和算术运算。所有函  
 数都有数字参数并返回数字值。所有三角函数的操作数和值  
 都是弧度而不是角度 , oracle没有提供内建的弧度和角度的转  
 换函数。 ABS() 返回n的绝对值 ACOS() 反余玄函数 , 返回-1  
 到1之间的数。 n表示弧度 0select ACOS(-1) pi, ACOS(1) ZERO  
 FROM dual PI ZERO 3.14159265 0 ASIN() 反正玄函数 , 返回-1  
 到1 , n表示弧度 ATAN() 反正切函数 , 返回n的反正切值 , n  
 表示弧度。 CEIL() 返回大于或等于n的最小整数。 COS() 返  
 回n的余玄值 , n为弧度 COSH() 返回n的双曲余玄值 , n为数  
 字。 0select COSH() FROM dual COSH(1.4) 2.15089847 EXP() 返  
 回e的n次幂 , e=2.71828183. FLOOR() 返回小于等于N的最大  
 整数。 LN() 返回N的自然对数 , N必须大于0 LOG(,) 返回  
 以n1为底n2的对数 MOD() 返回n1除以n2的余数 , POWER(,)

返回n1的n2次方 ROUND(,) 返回舍入小数点右边n2位的n1的值，n2的缺省值为0，这回将小数点最接近的整数，如果n2为负数就舍入到小数点左边相应的位上，n2必须是整数。0select ROUND(12345,-2),ROUND(12345.54321,2) FROM dualROUND(12345,-2) ROUND(12345.54321,2)12300 12345.54

SIGN() 如果n为负数，返回-1,如果n为正数，返回1，如果n=0返回0. SIN ( ) 返回n的正玄值,n为弧度。 SINH() 返回n的双曲正玄值,n为弧度。 SQRT() 返回n的平方根,n为弧度 TAN ( ) 返回n的正切值,n为弧度 TANH() 返回n的双曲正切值,n为弧度 TRUNC(,) 返回截尾到n2位小数的n1的值，n2缺省设置为0，当n2为缺省设置时会将n1截尾为整数，如果n2为负值，就截尾在小数点左边相应的位上。

单行日期函数 单行日期函数操作DATA数据类型，绝大多数都有DATA数据类型的参数，绝大多数返回的也是DATA数据类型的值。 ADD\_MONTHS(, <i> ) 返回日期d加上i个月后的结果。i可以使任意整数。如果i是一个小数，那么数据库将隐式的他转换成整数，将会截去小数点后面的部分。 LAST\_DAY() 函数返回包含日期d的月份的最后一天 MONTHS\_BETWEEN(,) 返回d1和d2之间月的数目,如果d1和d2的日的日期都相同，或者都使该月的最后一天，那么将返回一个整数，否则会返回的结果将包含一个分数。 NEW\_TIME(,,) d1是一个日期数据类型，当时区tz1中的日期和时间是d时，返回时区tz2中的日期和时间。tz1和tz2时字符串。 NEXT\_DAY(,) 返回日期d后由dow给出的条件的第一天，dow使用当前会话中给出的语言指定了一周中的某一天，返回的时间分量与d的时间分量相同。 0select NEXT\_DAY(01-Jan-2000,Monday) "1st

Monday",NEXT\_DAY(01-Nov-2004,Tuesday) 7 "2nd Tuesday")  
from dual.1st Monday 2nd Tuesday03-Jan-2000 09-Nov-2004  
ROUND([,]) 将日期d按照fmt指定的格式舍入，fmt为字符串。  
SYADATE 函数没有参数，返回当前日期和时间。  
TRUNC([,]) 返回由fmt指定的单位的日期d.单行转换函数  
单行转换函数用于操作多数据类型，在数据类型之间进行转换。  
CHARTORWID() c 使一个字符串，函数将c转换为ROWID  
数据类型。SELECT test\_id from test\_case where  
rowid=CHARTORWID(AAAA0SAACAAAALiAAA)  
CONVERT([,]) c尾字符串，dset、sset是两个字符集，函数将  
字符串c由sset字符集转换为dset字符集，sset的缺省设置为数  
据库的字符集。HEXTORAW() x为16进制的字符串，函数  
将16进制的x转换为RAW数据类型。RAWTOHEX() x是RAW  
数据类型字符串，函数将RAW数据类转换为16进制的数据类  
型。ROWIDTOCHAR() 函数将ROWID数据类型转换  
为CHAR数据类型。TO\_CHAR([,]) x是一个data或number数  
据类型，函数将x转换成fmt指定格式的char数据类型，如果x  
为日期nlsparm=NLS\_DATE\_LANGUAGE 控制返回的月份和  
日份所使用的语言。如果x为数  
字nlsparm=NLS\_NUMERIC\_CHARACTERS 用来指定小数位  
和千分位的分隔符，以及货币符号  
。NLS\_NUMERIC\_CHARACTERS="dg",  
NLS\_CURRENCY="string" TO\_DATE([,]) c表示字符串，fmt  
表示一种特殊格式的字符串。返回按照fmt格式显示  
的c,nlsparm表示使用的语言。函数将字符串c转换成date数据  
类型。TO\_MULTI\_BYTE() c表示一个字符串，函数将c的担

子截字符转换成多字节字符。 TO\_NUMBER([,[,] c表示字符串，fmt表示一个特殊格式的字符串，函数返回值按照fmt指定的格式显示。nlsparm表示语言，函数将返回c代表的数字。 TO\_SINGLE\_BYTE() 将字符串c中得多字节字符转化成等价的单字节字符。该函数仅当数据库字符集同时包含单字节和多字节字符时才使用 其它单行函数 BFILENAME(,) dir是一个directory类型的对象，file为一文件名。函数返回一个空的BFILE位置值指示符，函数用于初始化BFILE变量或者是BFILE列。 DECODE(,,[,,,[]) x是一个表达式，m1是一个匹配表达式，x与m1比较，如果m1等于x，那么返回r1,否则,x与m2比较，依次类推m3,m4,m5....直到有返回结果。 DUMP(,[,[,[,[]]) x是一个表达式或字符，fmt表示8进制、10进制、16进制、或则单字符。函数返回包含了有关x的内部表示信息的VARCHAR2类型的值。如果指定了n1,n2那么从n1开始的长度为n2的字节将被返回。 EMPTY\_BLOB() 该函数没有参数，函数返回一个空的BLOB位置指示符。函数用于初始化一个BLOB变量或BLOB列。 EMPTY\_CLOB() 该函数没有参数，函数返回一个空的CLOB位置指示符。函数用于初始化一个CLOB变量或CLOB列。 GREATEST() exp\_list是一列表表达式，返回其中最大的表达式，每个表达式都被隐含的转换第一个表达式的数据类型，如果第一个表达式是字符串数据类型中的任何一个，那么返回的结果是varchar2数据类型，同时使用的比较是非填充空格类型的比较。 LEAST() exp\_list是一列表表达式，返回其中最小的表达式，每个表达式都被隐含的转换第一个表达式的数据类型，如果第一个表达式是字符串数据类型中的任何一个，将返回的结果是varchar2数据类型，同

时使用的比较是非填充空格类型的比较。UID 该函数没有参数，返回唯一标示当前数据库用户的整数。USER 返回当前用户的用户名 USERENV() 基于opt返回包含当前会话信息。opt的可选值为：ISDBA 会话中SYSDBA脚色响应，返回TRUE SESSIONID 返回审计会话标示符 ENTRYID 返回可用的审计项标示符 INSTANCE 在会话连接后，返回实例标示符。该值只用于运行Parallel 服务器并且有多个实例的情况下使用。LANGUAGE 返回语言、地域、数据库设置的字符集。LANG 返回语言名称的ISO缩写。TERMINAL 为当前会话使用的终端或计算机返回操作系统的标示符。VSIZE() x是一个表达式。返回x内部表示的字节数。SQL中的组函数 组函数也叫集合函数，返回基于多个行的单一结果，行的准确数量无法确定，除非查询被执行并且所有的结果都被包含在内。与单行函数不同的是，在解析时所有的行都是已知的。由于这种差别使组函数与单行函数有在要求和行为上有微小的差异。

2、组（多行）函数与单行函数相比，oracle提供了丰富的基于组的，多行的函数。这些函数可以在0select或0select的having子句中使用，当用于0select子串时常常都和GROUP BY一起使用。AVG([{DISYINCT|ALL}]) 返回数值的平均值。缺省设置为ALL.SELECT AVG(sal),AVG(ALL sal),AVG(DISTINCT sal) FROM scott.empAVG(SAL) AVG(ALL SAL) AVG(DISTINCT SAL)1877.94118 1877.94118 1916.071413 COUNT({\*|DISTINCT|ALL}) 返回查询中行的数目，缺省设置是ALL,\*表示返回所有的行。MAX([{DISTINCT|ALL}]) 返回选择列表项目的最大值，如果x是字符串数据类型，他返回一个VARCHAR2数据类型，如果X是一个DATA数据类型，



返回一个日期，如果X是numeric数据类型，返回一个数字。注意distinct和all不起作用，应为最大值与这两种设置是相同的。MIN([{DISTINCT|ALL}]) 返回选择列表项目的最小值。STDDEV([{DISTINCT|ALL}]) 返回选者的列表项目的标准差，所谓标准差是方差的平方根。SUM([{DISTINCT|ALL}]) 返回选择列表项目的数值的总和。

VARIANCE([{DISTINCT|ALL}]) 返回选择列表项目的统计方差。用GROUP BY给数据分组 正如题目暗示的那样组函数就是操作那些已经分好组的数据，我们告诉数据库用GROUP BY怎样给数据分组或者分类，当我们在SELECT语句的SELECT子句中使用组函数时，我们必须把为分组或非常数列放置在GROUP BY子句中，如果没有用group by进行专门处理，那么缺省的分类是将整个结果设为一类。

```
0select stat,counter(*) zip_count from zip_codes GROUP BY state.ST  
ZIP_COUNT-- -----AK 360AL 1212AR 1309AZ 768CA 3982
```

在这个例子中，我们用state字段分类；如果我们要将结果按照zip\_codes排序,可以用ORDER BY语句，ORDER BY子句可以使用列或组函数。

```
0select stat,counter(*) zip_count from  
zip_codes GROUP BY state ORDER BY COUNT(*) DESC.ST  
COUNT(*)-- -----NY 4312PA 4297TX 4123CA 3982
```

用HAVING子句限制分组数据 现在你已经知道了在查询的SELECT语句和ORDER BY子句中使用主函数，组函数只能用于两个子串中，组函数不能用于WHERE子串中，例如下面的查询是错误的：错误SELECT sales\_clerk,SUN(sale\_amount) FROM gross\_sales WHERE sales\_dept=OUTSIDE AND SUM(sale\_amount)>10000 GROUP BY sales\_clerk 这个语句中数

数据库不知道SUM()是什么，当我们需要指示数据库对行分组，然后限制分组后的行的输出时，正确的方法是使

用HAVING语句：`SELECT sales_clerk,SUN(sale_amount)`

`FROM gross_sales WHERE sales_dept=OUTSIDE GROUP BY`

`sales_clerkHAVING SUM(sale_amount)>10000.` 3、嵌套函数

函数可以嵌套。一个函数的输出可以是另一个函数的输入。操

作数有一个可继承的执行过程。但函数的优先权只是基于位

置，函数遵循由内到外，由左到右的原则。嵌套技术一般用

于象DECODE这样的能被用于逻辑判断语句IF...THEN...ELSE

的函数。嵌套函数可以包括在组函数中嵌套单行函数，或者

组函数嵌套入单行函数或组函数中。比如下面的例子

：`SELECT deptno, GREATEST(COUNT(DISTINCT`

`job),COUNT(DISTINCT mgr) cnt,COUNT(DISTINCT job)`

`jobs,COUNT(DISTINCT mgr) mgrsFROM empGROUP BY`

`deptno.` 100Test 下载频道开通，各类考试题目直接下载。详细

请访问 [www.100test.com](http://www.100test.com)