

ORACLE 常用函数功能演示

文档撰写：孟祥新

写在前面：

ORACLE 内置函数，相当于 C 语言中的库函数，提供了常规数据库编程中所需的绝大多数基本功能，并且通过函数的组合或嵌套使用，可以发挥其强大的功能。由于这些内置函数都是由 ORACLE 公司的编程大家所写，而且其执行代码是内嵌在 ORACLE 数据库系统中的，因此其执行效率极高。

在数据库编程中能广泛使用 ORACLE 所提供的内置函数，将大大提高程序代码的执行效率。“站在高人的肩膀上，将使你变得更高”。

演示之前的环境设置：

为了在 SQL*PLUS 中直观地显示函数返回的结果，我将“头”信息先屏蔽掉。操作如下：



一、单行函数

1. 数值函数

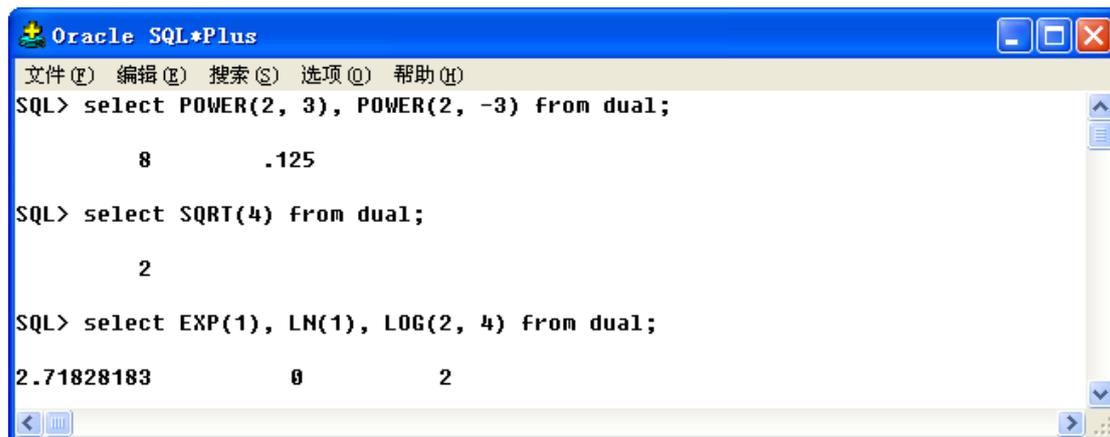
(1) 三角函数

- ① SIN、ASIN、SINH（正弦、反正弦、双曲正弦）
- ② COS、ACOS、COSH（余弦、反余弦、双曲余弦）
- ③ TAN、ATAN、TANH（正切、反正切，双曲正切）

(2) 幂、方根及对数运算函数

- ① POWER(m, n)（取 m 的 n 次幂）
- ② SQRT(n)（取 n 的平方根）
- ③ EXP(n)（取自然对数底 e 的 n 次方根）
- ④ LN(n)（取 n 的自然对数）
- ⑤ LOG(m, n)（取以 m 为底 n 的对数）

以上函数功能演示如下：



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select POWER(2, 3), POWER(2, -3) from dual;

           8           .125

SQL> select SQRT(4) from dual;

           2

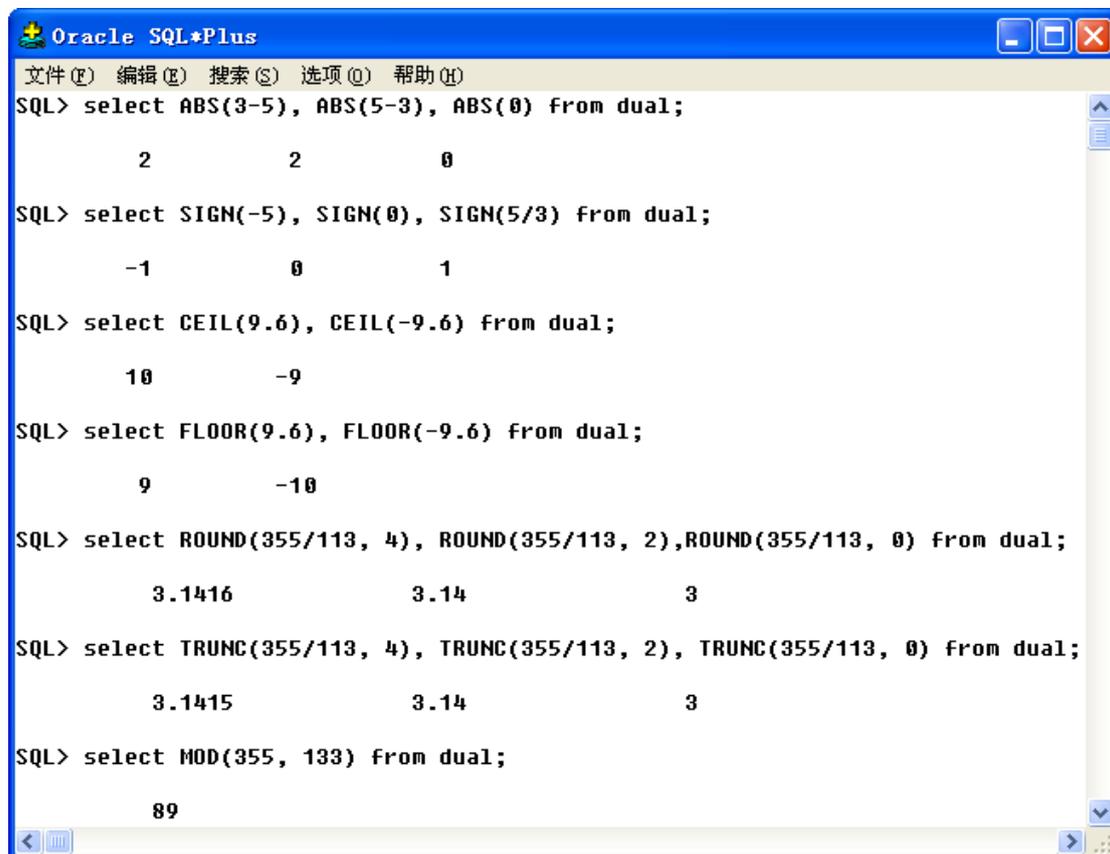
SQL> select EXP(1), LN(1), LOG(2, 4) from dual;

2.71828183           0           2
```

(3) 数值处理函数

- ① ABS(n) (取绝对值)
- ② SIGN(n) (符号函数)
- ③ CEIL(n) (取不小于 n 的最小整数)
- ④ FLOOR(n) (取不大于 n 的最大整数)
- ⑤ ROUND(n, [m]) (按 m 精度对 n 进行 4 舍 5 入)
- ⑥ TRUNC(n, [m]) (按 m 精度对 n 进行截取)
- ⑦ MOD(m, n) (取 m 除以 n 的余数)

以上函数功能演示如下:



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select ABS(3-5), ABS(5-3), ABS(0) from dual;

           2           2           0

SQL> select SIGN(-5), SIGN(0), SIGN(5/3) from dual;

          -1           0           1

SQL> select CEIL(9.6), CEIL(-9.6) from dual;

          10           -9

SQL> select FLOOR(9.6), FLOOR(-9.6) from dual;

           9           -10

SQL> select ROUND(355/113, 4), ROUND(355/113, 2), ROUND(355/113, 0) from dual;

           3.1416           3.14           3

SQL> select TRUNC(355/113, 4), TRUNC(355/113, 2), TRUNC(355/113, 0) from dual;

           3.1415           3.14           3

SQL> select MOD(355, 133) from dual;

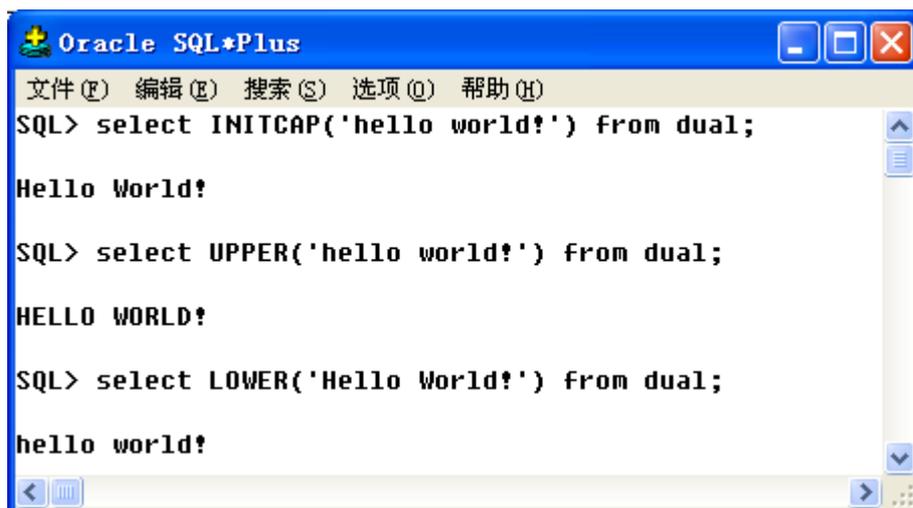
           89
```

2. 字符处理函数

(1) 对字符串的大小写处理的函数

- ① INITCAP (功能: 将字符串中每个单词的首字母, 变换为大写。)
- ② UPPER (功能: 将字符串中的所有字母, 转换为大写。)
- ③ LOWER (功能: 将字符串中的所有字母, 转换为小写。)

以上函数功能演示如下:

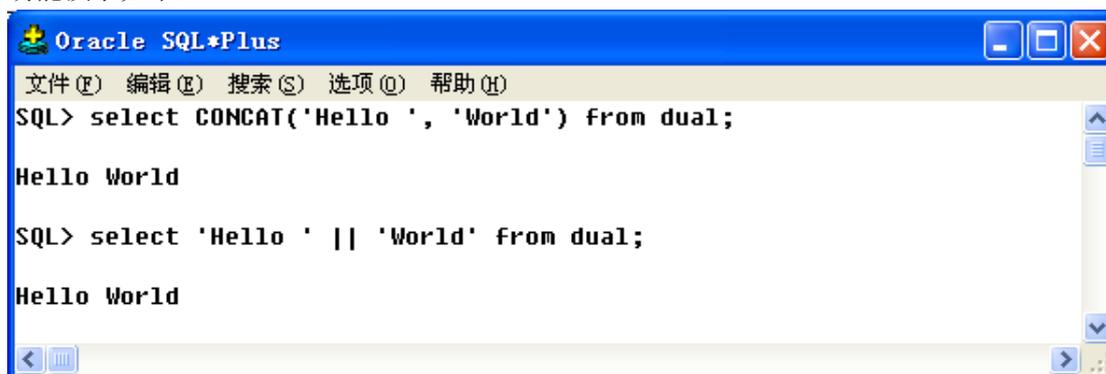


```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select INITCAP('hello world!') from dual;
Hello World!
SQL> select UPPER('hello world!') from dual;
HELLO WORLD!
SQL> select LOWER('Hello World!') from dual;
hello world!
```

(2) 对字符串进行处理的函数

- ① CONCAT (字符串连接函数, 也可用 || 替代)

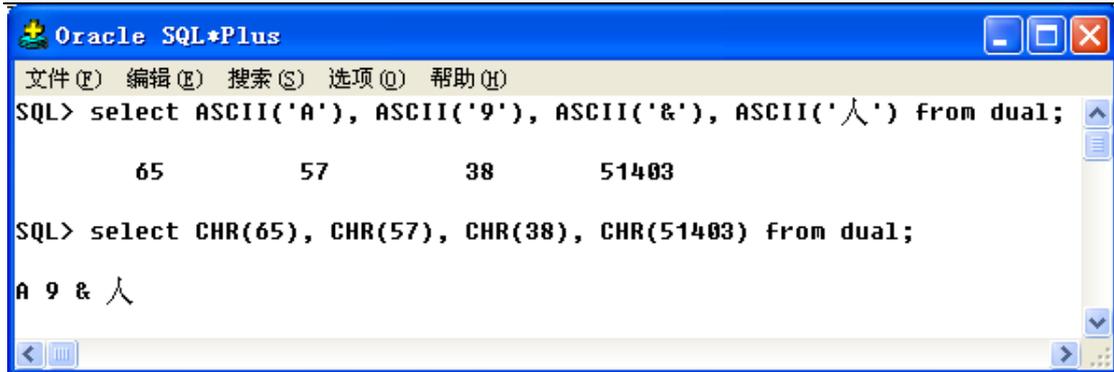
功能演示如下:



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select CONCAT('Hello ', 'World') from dual;
Hello World
SQL> select 'Hello ' || 'World' from dual;
Hello World
```

- ② ASCII (取字符的编码)
- ③ CHR (将编码转换为对应的字符)

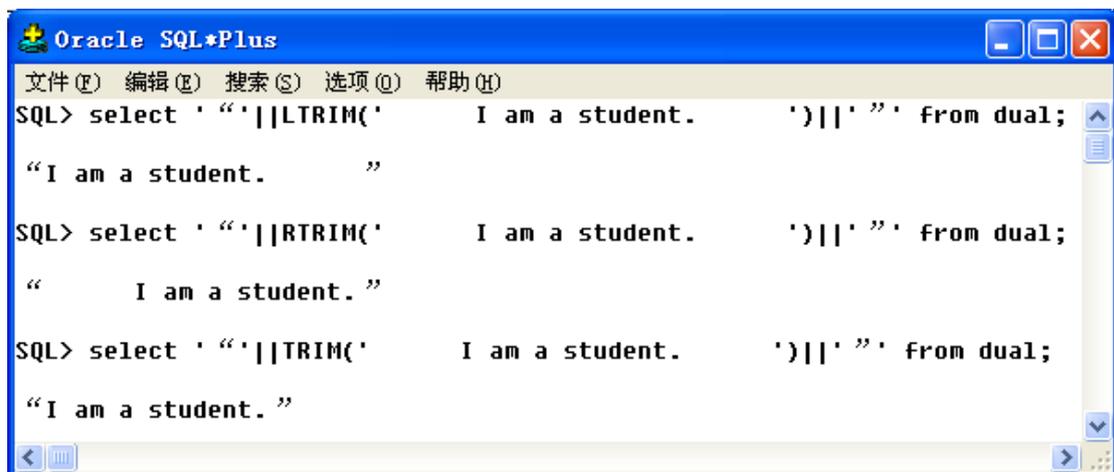
以上函数功能演示如下:



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select ASCII('A'), ASCII('9'), ASCII('&'), ASCII('人') from dual;
           65           57           38           51403
SQL> select CHR(65), CHR(57), CHR(38), CHR(51403) from dual;
A 9 & 人
```

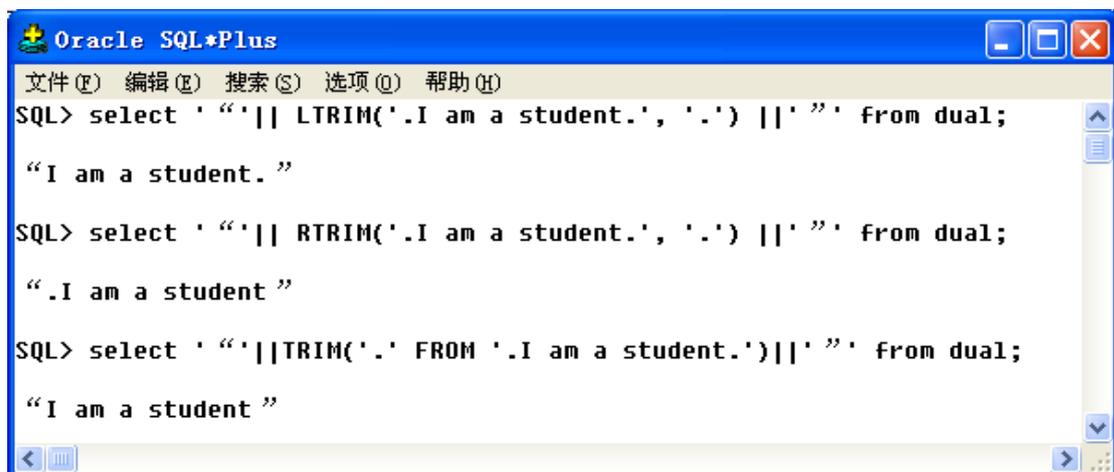
- ④ LTRIM (去除左侧空格/指定字符)
- ⑤ RTRIM (去除右侧空格/指定字符)
- ⑥ TRIM (去除两侧的空格/去除指定字符)

以上函数去除空格功能演示如下:



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select ' ' || LTRIM(' I am a student. ') || ' ' from dual;
" I am a student. "
SQL> select ' ' || RTRIM(' I am a student. ') || ' ' from dual;
" I am a student. "
SQL> select ' ' || TRIM(' I am a student. ') || ' ' from dual;
" I am a student. "
```

以上函数去除指定字符功能演示如下:



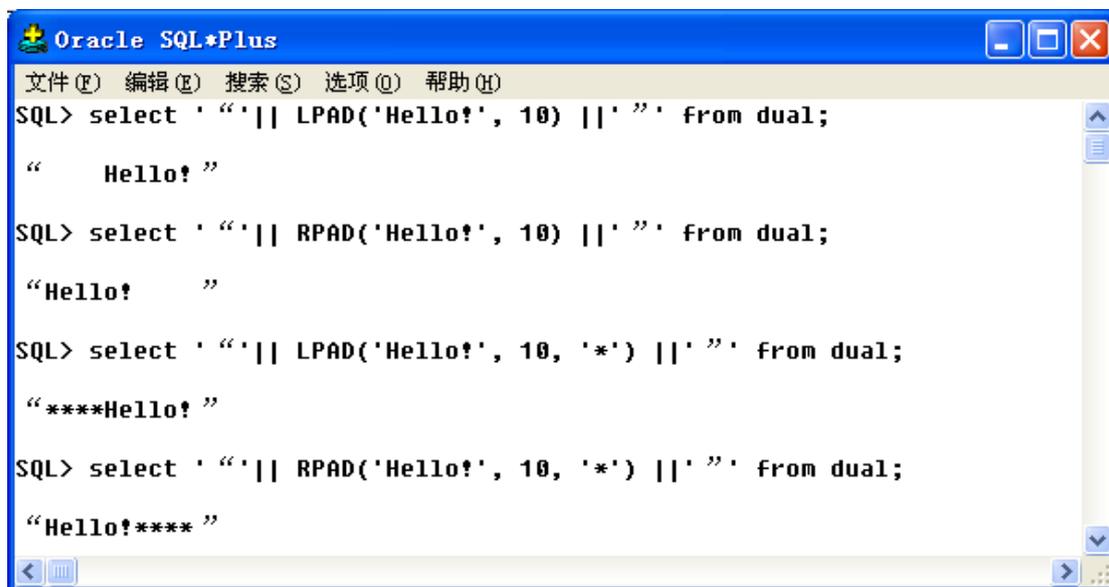
```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select ' ' || LTRIM('.I am a student.', '.') || ' ' from dual;
" I am a student. "
SQL> select ' ' || RTRIM('.I am a student.', '.') || ' ' from dual;
".I am a student "
SQL> select ' ' || TRIM('.') FROM '.I am a student.' || ' ' from dual;
" I am a student "
```

请注意: 在用 TRIM 去除字符串中指定字符 (非空格) 时的特殊用法。

⑦ LPAD（左侧添充空格/指定字符）

⑧ RPAD（右侧添充空格/指定字符）

以上函数功能演示如下：



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select ' ' || LPAD('Hello!', 10) || ' ' from dual;
      "      Hello!      "
SQL> select ' ' || RPAD('Hello!', 10) || ' ' from dual;
      "Hello!          "
SQL> select ' ' || LPAD('Hello!', 10, '*') || ' ' from dual;
      "****Hello!      "
SQL> select ' ' || RPAD('Hello!', 10, '*') || ' ' from dual;
      "Hello!****      "
```

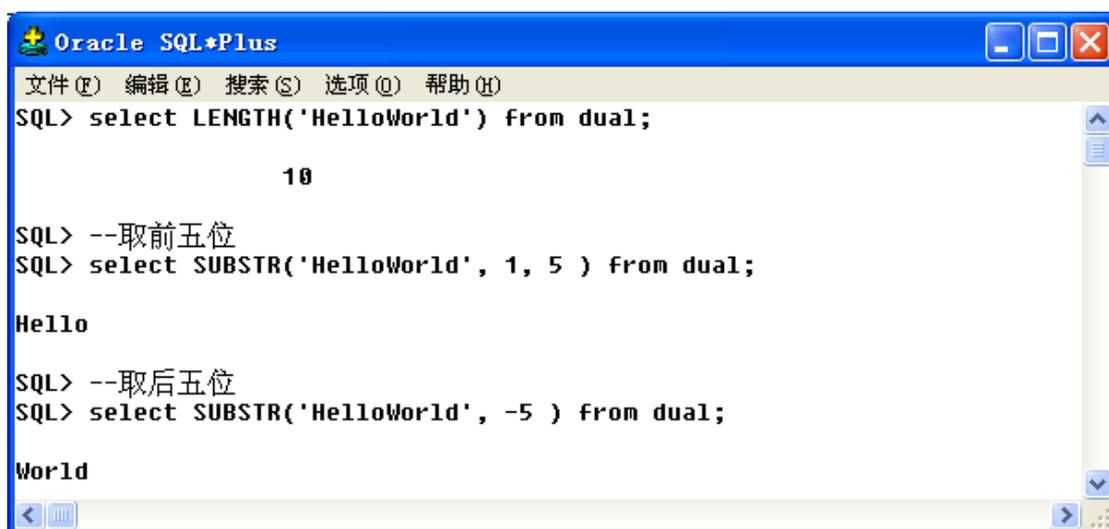
⑨ LENGTH（取字符串的长度）

⑩ SUBSTR（截取子字符串）

用法格式：SUBSTR(string, start [,length])

功能说明：从字符串 string 的 start 位置开始向后截取 length 长度的子串

以上函数功能演示如下：



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select LENGTH('HelloWorld') from dual;
          10
SQL> --取前五位
SQL> select SUBSTR('HelloWorld', 1, 5 ) from dual;
      Hello
SQL> --取后五位
SQL> select SUBSTR('HelloWorld', -5 ) from dual;
      World
```

(1) INSTR（查找子字符串的位置）

用法格式：INSTR(string, search_string [,n [,m]])

功能说明：从字符串 string 的第 n 个字符开始查找 search_string 第 m 次出现的位置，如果按条件没有找到字符串，则返回 0

(注意：此函数返回的位置是以原字符串首字符所在位置为起始点的，首字符位置为1)

函数功能演示如下：



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> --查字符a第一次出现的位置
SQL> select INSTR('I am a student, and I like study.', 'a' ) from dual;

          3

SQL> --查字符a第二次出现的位置
SQL> select INSTR('I am a student, and I like study.', 'a', 1, 2) from dual;

          6

SQL> --从字符串第5个字符位置开始查找第一次出现a的位置
SQL> select INSTR('I am a student, and I like study.', 'a', 5 ) from dual;

          6

SQL> --从字符串第5个字符位置开始查找第二次出现a的位置
SQL> select INSTR('I am a student, and I like study.', 'a', 5, 2) from dual;

         17

SQL> --从后向前查第一次出现字符a的位置(即:最后一次出现字符a的位置)
SQL> select INSTR('I am a student, and I like study.', 'a', -1 ) from dual;

         17

SQL> --从后向前查第二次出现字符a的位置
SQL> select INSTR('I am a student, and I like study.', 'a', -1, 2) from dual;

          6

SQL> --查找字符串是否存在字符w
SQL> select INSTR('I am a student, and I like study.', 'W' ) from dual;

          0
```

(12) REPLACE (子串替换)

用法格式: REPLACE(string, search_string [,replace_string])

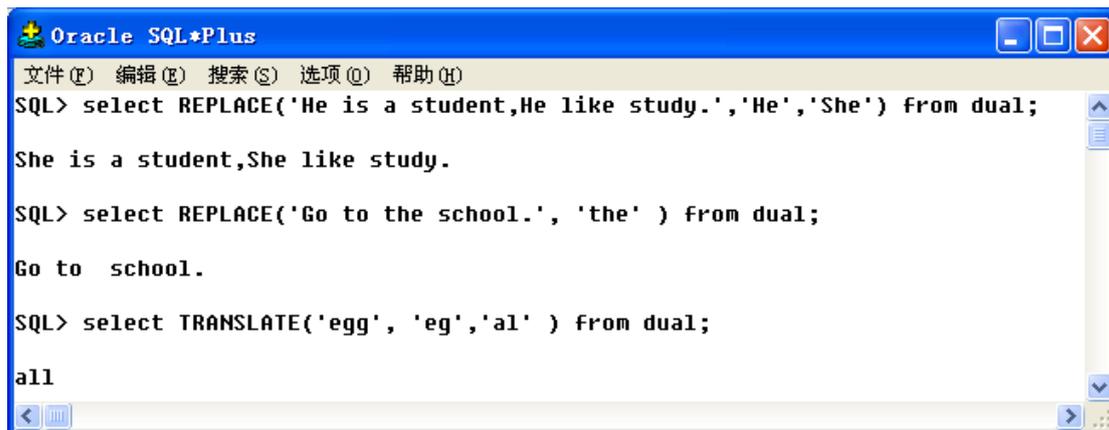
功能说明: 将字符串 string 中的所有 search_string 都替换为 replace_string, 如果省略 replace_string, 则将字符串 string 中的所有 search_string 都去掉。

(13) TRANSLATE (字符转换)

用法格式: TRANSLATE(string, search_set, replace_set)

功能说明: 将 string 中出现在 search_set 中的字符转换为 replace_set 相应位置的字符。

以上函数功能演示如下：



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> select REPLACE('He is a student,He like study.','He','She') from dual;
She is a student,She like study.
SQL> select REPLACE('Go to the school.','the' ) from dual;
Go to school.
SQL> select TRANSLATE('egg', 'eg','al' ) from dual;
all
```

3. 日期函数

- ① SYSDATE (取当前的数据库系统时间)
- ② ADD_MONTHS(加减指定的月份)
- ③ MONTHS_BETWEEN (取两个日期之间相隔的月数)
- ④ LAST_DAY (取指定日期所在月的最后一天)

以上函数功能演示如下：

```

Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> --直接调用SYSDATE函数,得到当前数据库系统的时间
SQL> select SYSDATE from dual;

16-10月-07

SQL> --由于这种环境的默认时间显示格式,不符合我们的习惯,因此可以用如下方法
SQL> select TO_CHAR(SYSDATE, 'YYYY-MM-DD HH24:MI:SS') from dual;

2007-10-16 15:50:52

SQL> --也可以采用更改当前会话(session)的方式,来统一设置当前会话的日期显示格式
SQL> ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';

会话已更改。

SQL> select SYSDATE from dual;

2007-10-16 15:51:14

SQL> --显示后一个月的时间
SQL> select ADD_MONTHS(SYSDATE, 1) from dual;

2007-11-16 15:51:20

SQL> --显示前一个月的时间
SQL> select ADD_MONTHS(SYSDATE, -1) from dual;

2007-09-16 15:51:26

SQL> --显示后一个月和前一个月之间的月数
SQL> select MONTHS_BETWEEN(ADD_MONTHS(SYSDATE, 1), ADD_MONTHS(SYSDATE, -1)) from dual;

                2

SQL> --显示本月最后一天的时间
SQL> select LAST_DAY(SYSDATE) from dual;

2007-10-31 15:53:25
    
```

- ⑤ ROUND(date, 'format_string')
功能：按日期掩码 format_string 的格式对给定的日期 date 进行舍/入。
- ⑥ TRUNC(date, 'format_string')
功能：按日期掩码 format_string 的格式对给定的日期 date 进行截取。

以上函数功能演示如下：

```

Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> --查看当前的日期
SQL> select SYSDATE from dual;

2007-10-16 20:11:36

SQL> --ROUND(日期)
SQL> select ROUND(sysdate, 'YYYY'), ROUND(sysdate, 'MM'), ROUND(sysdate, 'DD') from dual;

2008-01-01 00:00:00 2007-11-01 00:00:00 2007-10-17 00:00:00

SQL> --TRUNC(日期)
SQL> select TRUNC(sysdate, 'YYYY'), TRUNC(sysdate, 'MM'), TRUNC(sysdate, 'DD') from dual;

2007-01-01 00:00:00 2007-10-01 00:00:00 2007-10-16 00:00:00
    
```

4. 类型转换函数

功能：在数值和字符，以及字符和日期之间进行数据类型的转换

说明：在 ORACLE 数据服务器，可以对数值形式的字符串隐式地转换为数值类型的数据，也可以将数值类型的数据隐式地转换为数值形式的字符串。但也可以进行如下的显式转换。

- ① TO_CHAR(n) (将数值 n 转换为字符类型)
- ② TO_NUMBER('string') (将字符串 string 转换为数值类型)

知识扩展：常用进制的转换

```

Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> --将十进制转换为十六进制
SQL> select TO_CHAR(65535,'xxxx'), TO_CHAR(65535,'XXXX') from dual;

      ffff      FFFF

SQL> --将十六进制转换为十进制
SQL> select TO_NUMBER('f','x'), TO_NUMBER('ffff','xxxx') from dual;

           15           65535

SQL> --将二进制转换为十进制
SQL> select BIN_TO_NUM(1,0), BIN_TO_NUM(1,0,1,0) from dual;

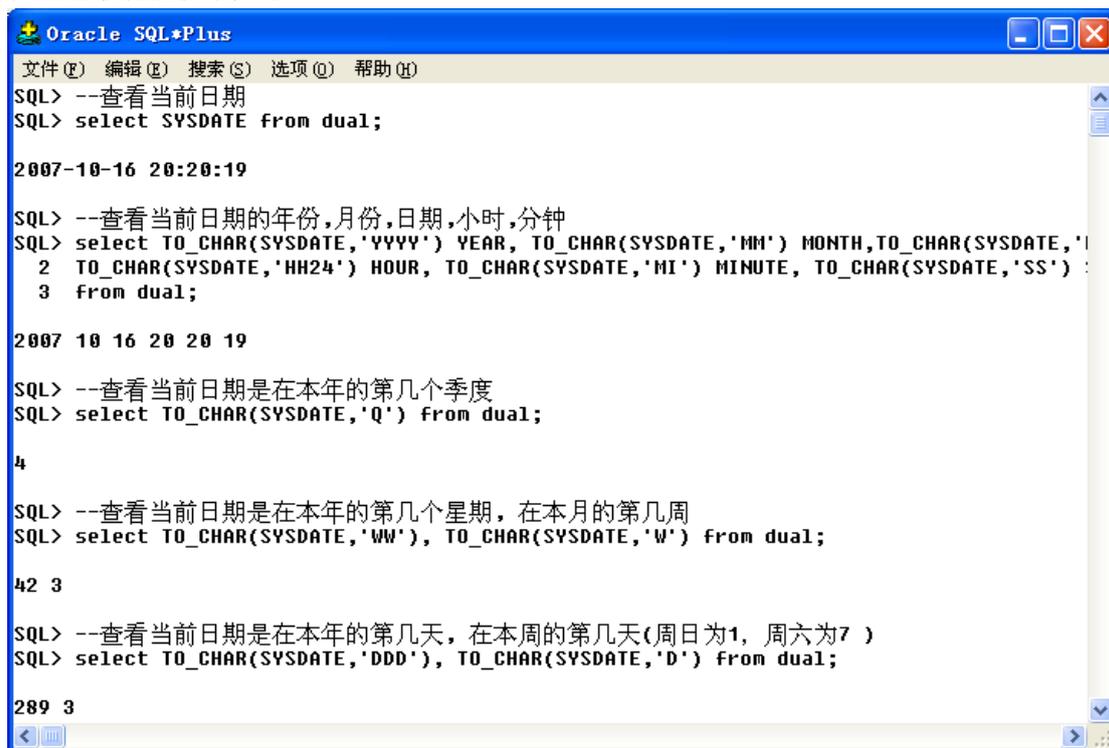
           2           10
    
```

- ③ TO_CHAR(date, 'format_string') (按照时间掩码 format_string 的格式要求，将日期型数据 date 转换为字符型数据)
- ④ TO_DATE('string', 'format_string')(将字符型数据 string 按照时间掩码 format_string 的格式要求，转换为相应的日期型数据)

常用的时间格式掩码如下：

掩码元素	含义
YYYY	四位数年份 (如: 2005)
YY	二位数年份 (如 05)
Q	季度 (1-4)
MM	月份 (01-12)
WW	年的星期数 (1-53)，其中第一星期为年的第一天至第七天
W	月的星期数 (1-5)，其中第一星期为月的第一天至第七天
DDD	年的日 (1-366)
DD	月的日 (1-31)
D	周的日 (1-7)，其中周日为 1，周六为 7
HH24	24 小时制 (0-23)
MI	分钟 (0-59)
SS	秒 (0-59)
SSSSS	自午夜之后的秒 (0-86399)

以上函数功能演示如下:



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> --查看当前日期
SQL> select SYSDATE from dual;

2007-10-16 20:20:19

SQL> --查看当前日期的年份,月份,日期,小时,分钟
SQL> select TO_CHAR(SYSDATE,'YYYY') YEAR, TO_CHAR(SYSDATE,'MM') MONTH, TO_CHAR(SYSDATE,'DD') DAY,
2 TO_CHAR(SYSDATE,'HH24') HOUR, TO_CHAR(SYSDATE,'MI') MINUTE, TO_CHAR(SYSDATE,'SS') SECOND
3 from dual;

2007 10 16 20 20 19

SQL> --查看当前日期是在本年的第几个季度
SQL> select TO_CHAR(SYSDATE,'Q') from dual;

4

SQL> --查看当前日期是在本年的第几个星期,在本月的第几周
SQL> select TO_CHAR(SYSDATE,'WW'), TO_CHAR(SYSDATE,'W') from dual;

42 3

SQL> --查看当前日期是在本年的第几天,在本周的第几天(周日为1,周六为7)
SQL> select TO_CHAR(SYSDATE,'DDD'), TO_CHAR(SYSDATE,'D') from dual;

289 3
```

5. 空值转换函数

① NVL(expr1, expr2)

功能: 如果 expr1 的计算结果为 null 值, 则返回 expr2 的值。如果 expr1 的计算结果不是 null 值, 则返回 expr1 的值

② NVL2(expr1, expr2, expr3)

功能: 如果 expr1 非空, 则返回 expr2 的值; 如果 expr1 为空, 则返回 expr3 的值。

③ NULLIF(expr1, expr2)

功能: 比较两个表达式, 如果相等返回空; 如果不相等, 返回第一个表达式的值。

④ COALESCE(expr1, expr2, ..., exprn)

功能: 返回表达式列表中的第一个非空表达式的值。

6. 查看环境参数函数

(1) USER 函数 (返回当前用户名)

(2) USERENV 函数

① USERENV('ISDBA') (查看当前用户是否是 DBA)

② USERENV('LANGUAGE') (查看环境所用的字符集)

③ USERENV('TERMINAL') (查看当前用户终端的标识)

以上函数用法演示如下:

```
select USERENV('ISDBA'), USERENV('LANGUAGE'), USERENV('TERMINAL') from
dual;
```

(3) SYS_CONTEXT 函数

说明: SYS_CONTEXT 函数是在 ORACLE 8i 中新增的, 在功能上比 USERENV 函数更加全面。

用法格式：SYS_CONTEXT('USERENV', '<parameter>')

根据参数 parameter 来返回相应的信息

① 登录验证类型

```
select SYS_CONTEXT('USERENV', 'AUTHENTICATION_TYPE') from dual;
```

② 当前登录数据库的用户名

```
select SYS_CONTEXT('USERENV', 'CURRENT_USER') from dual;
```

③ 当前登录客户端的操作系统用户名

```
select SYS_CONTEXT('USERENV', 'OS_USER') from dual;
```

④ 当前所在客户机名（包括所在工作组名）

```
select SYS_CONTEXT('USERENV', 'HOST') from dual;
```

⑤ 当前所在客户机/终端名

```
select SYS_CONTEXT('USERENV', 'TERMINAL') from dual;
```

⑥ 当前所在客户端的 IP

```
select SYS_CONTEXT('USERENV', 'IP_ADDRESS') from dual;
```

⑦ 当前是否为 DBA 帐户

```
select SYS_CONTEXT('USERENV', 'ISDBA') from dual;
```

⑧ 当前的字符集

```
select SYS_CONTEXT('USERENV', 'LANGUAGE') from dual;
```

⑨ 当前所使用的网络协议

```
select SYS_CONTEXT('USERENV', 'NETWORK_PROTOCOL') from dual;
```

⑩ 当前所使用的日历

```
select SYS_CONTEXT('USERENV', 'NLS_CALENDAR') from dual;
```

(11) 当前的货币单位

```
select SYS_CONTEXT('USERENV', 'NLS_CURRENCY') from dual;
```

(12) 当前的日期格式

```
select SYS_CONTEXT('USERENV', 'NLS_DATE_FORMAT') from dual;
```

(13) 当前的日期语言

```
select SYS_CONTEXT('USERENV', 'NLS_DATE_LANGUAGE') from dual;
```

(14) 当前所在的地域

```
select SYS_CONTEXT('USERENV', 'NLS_TERRITORY') from dual;
```

(15) 当前代理用户

```
select SYS_CONTEXT('USERENV', 'PROXY_USER') from dual;
```

7. DECODE 函数

DECODE 函数用法格式与说明：

格式：

```
DECODE(col|expression,  
        search1, result1  
        [, search2, result2 ]  
        [, search3, result3 ]  
        [, ....., ..... ]  
        [, default]  
    )
```

说明：

第一个参数可是是表的列，也可以是一个表达式。

如果第一参数的值与 search1 相等，则函数返回 result1 的值；

如果第一参数的值与 search2 相等，则函数返回 result2 的值；以此类推。

如果第一参数的值与所有的 search 都不相等，则函数返回 default 的值。

此函数用法示例如下：

```

Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> --对员工的月薪按每1000元为一个梯度进行分档
SQL> select ename, sal,
  2  DECODE( TRUNC( sal/1000, 0 ),
  3           0, 1,
  4           1, 2,
  5           2, 3,
  6           3, 4,
  7           4, 5,
  8           5, 6,
  9           6, 7,
10          7, 8 ) grade
11  from emp order by grade;

ENAME          SAL      GRADE
-----
SMITH           800         1
JAMES           950         1
ALLEN          1600         2
WARD           1250         2
ADAMS           1100         2
TURNER          1500         2
MARTIN          1250         2
MILLER          1300         2
JONES           2975         3
BLAKE           2850         3
CLARK           2450         3
SCOTT           3000         4
FORD            3000         4
KING            5000         6

已选择14行。
    
```

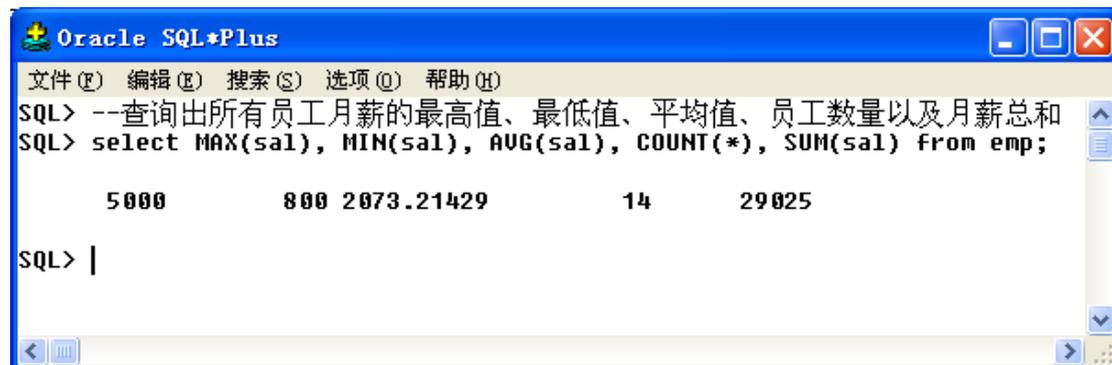
二、聚组函数

功能：对一组数据进行运算。

函数	功能描述
MAX	最大值
MIN	最小值
AVG	平均值
COUNT	数量值

SUM	总计值
-----	-----

以上函数功能演示如下：



```
Oracle SQL*Plus
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
SQL> --查询出所有员工月薪的最高值、最低值、平均值、员工数量以及月薪总和
SQL> select MAX(sal), MIN(sal), AVG(sal), COUNT(*), SUM(sal) from emp;

          5000          800 2073.21429          14          29025

SQL> |
```

三、分析函数

分析函数简介：ORACLE 从 8.1.6 版开始提供分析函数，分析函数用于计算基于组的某种聚合值，它和聚组函数的不同之处是对于每个组返回多行值，而聚组函数对于每个组只返回一行值。

（ORACLE 有一整套分析函数，功能极其强大，用法也非常灵活，关于这一块的功能演示，请待续……）