[SQL\*Plus中替换变量与定义变量](https://www.cnblogs.com/kerrycode/p/3967275.html)

**替换变量**

SQL\*Plus中的替换变量又叫替代变量，它一般用来临时存储相关数据；在SQL语句之间传递值。一般使用&或&&前缀来指定替换变量. 关于使用替换变量，一般是利用其创建通用的脚本或达到和用户交换目的。如下所示：

例如，我需要查看表的相关信息，如果不使用替换变量，每次查询我都要修改脚本。非常不便，如果使用替换变量，我们可以将下面脚本存放在tab.sql脚本中,每次运行时，只需要输入替换变量的值就可以了。

COL LOGGING FOR A7;

COL OWNER FOR A12;

COL TABLE\_NAME FOR A30

COL TABLESPACE\_NAME FOR A30

SELECT OWNER, TABLE\_NAME, TABLESPACE\_NAME, STATUS, COMPRESSION,

LOGGING, PARTITIONED, NUM\_ROWS,

TO\_CHAR(LAST\_ANALYZED,'YY-MM-DD HH24:MI:SS') LAST\_ANALYZED

FROM DBA\_TABLES

WHERE TABLE\_NAME LIKE '&table\_name\_like%'

ORDER BY TABLE\_NAME;

epps> @tab.sql

Enter value for table\_name\_like: EMP

old 4: WHERE TABLE\_NAME LIKE '&table\_name\_like%'

new 4: WHERE TABLE\_NAME LIKE 'EMP%'

OWNER TABLE\_NAME TABLESPACE\_NAME STATUS COMPRESS LOGGING PAR NUM\_ROWS LAST\_ANALYZED

------- --------------- ---------------------- -------- -------- ------- --- ---------- -----------------

SCOTT EMP USERS VALID DISABLED YES NO 14 09-07-20 22:00:12

WIPOWNER EMPLOYEE\_MASTER WIPOWNER\_DATA VALID DISABLED YES NO 12084 14-09-07 14:08:22

epps> @tab.sql

Enter value for table\_name\_like: DEPT

old 4: WHERE TABLE\_NAME LIKE '&table\_name\_like%'

new 4: WHERE TABLE\_NAME LIKE 'DEPT%'

OWNER TABLE\_NAME TABLESPACE\_NAME STATUS COMPRESS LOGGING PAR NUM\_ROWS LAST\_ANALYZED

------- -------------- ------------------ -------- -------- ------- --- ---------- -----------------

SCOTT DEPT USERS VALID DISABLED YES NO 4 09-07-20 22:00:12

epps>

[](https://images0.cnblogs.com/blog/73542/201409/112136356372330.png)

如果替换变量为数字类型，那么可以直接是&+变量名称，如下所示&DEPTNO，

SQL> SELECT deptno, dname, loc

2 FROM SCOTT.DEPT

3 WHERE DEPTNO= &DEPTNO;

Enter value for deptno: 10

old 3: WHERE DEPTNO= &DEPTNO

new 3: WHERE DEPTNO= 10

DEPTNO DNAME LOC

---------- -------------- -------------

10 ACCOUNTING NEW YORK

如果是字符串类型或日期类型，最好使用单引号将其包裹起来，否则在输入变量值时则必须加上单引号，不那样做就会报错，如下所示。

SQL> SELECT deptno, dname, loc

2 FROM SCOTT.DEPT

3 WHERE DNAME= &DNAME;

Enter value for dname: SALES

old 3: WHERE DNAME= &DNAME

new 3: WHERE DNAME= SALES

WHERE DNAME= SALES

\*

ERROR at line 3:

ORA-00904: "SALES": invalid identifier

SQL> SELECT deptno, dname, loc

2 FROM SCOTT.DEPT

3 WHERE DNAME= &DNAME;

Enter value for dname: 'SALES'

old 3: WHERE DNAME= &DNAME

new 3: WHERE DNAME= 'SALES'

DEPTNO DNAME LOC

---------- -------------- -------------

30 SALES CHICAGO

**关于替换变量中&与&&的区别,两者区别在于作用范围不一样，&引用的替换变量只在当前SQL有效，而&&引用的替换变量则在当前会话有效**。下面以人们经常引用的例子来说明

SQL> SELECT 2+&NUM FROM DUAL;

Enter value for num: 2

old 1: SELECT 2+&NUM FROM DUAL

new 1: SELECT 2+2 FROM DUAL

2+2

----------

4

SQL> SELECT 3+&NUM FROM DUAL;

Enter value for num: 3

old 1: SELECT 3+&NUM FROM DUAL

new 1: SELECT 3+3 FROM DUAL

3+3

----------

6

SQL> SELECT 2+&&NUM FROM DUAL;

Enter value for num: 2

old 1: SELECT 2+&&NUM FROM DUAL

new 1: SELECT 2+2 FROM DUAL

2+2

----------

4

SQL> SELECT 3+&&NUM FROM DUAL;

old 1: SELECT 3+&&NUM FROM DUAL

new 1: SELECT 3+2 FROM DUAL

3+2

----------

5

**小知识1：**在存储过程或包体里面，经常有在字符串中使用&的情况，执行脚本时，经常会将这些字符串视为替换变量，要求输入值，这样烦不甚烦，其实只需要设置一下SQL\*PLUS的环境变量即可避免这种情况。通常通过SET DEFINE OFF

**小知识2：**如果要显示SQL\*Plus使用替换值替换后的脚本文件，可以使用SET VERIFY ON/OFF 命令

SQL> SET VERIFY OFF

SQL> SELECT 2+&NUM FROM DUAL;

2+2

----------

4

SQL> SET VERIFY ON

SQL> SELECT 2+&NUM FROM DUAL;

old 1: SELECT 2+&NUM FROM DUAL

new 1: SELECT 2+2 FROM DUAL

2+2

----------

4

**用户定义变量**

使用DEFINE定义了的变量（或许我们可以叫其定义常量，和C/C++中的DEFINE非常类似），可以使用&引用声明的变量。其作用范围或生命周期通常是整个会话。如果定义了变量后，需要清除变量，则可以使用UNDEFINE清除变量，使用DEFINE VARIABLE来查看变量

SQL> DEFINE NUM=2; --定义变量

SQL> DEFINE NUM --查看变量

DEFINE NUM = "2" (CHAR)

SQL> SELECT 2+&NUM FROM DUAL; --引用变量

old 1: SELECT 2+&NUM FROM DUAL

new 1: SELECT 2+2 FROM DUAL

2+2

----------

4

SQL> UNDEFINE NUM; --清除变量

SQL> SELECT 2+&NUM FROM DUAL; --清除变量后，&NUM变成替换变量了。需要输入值

Enter value for num:

**小知识点：**查看当前会话下，SQL\*Plus下定义的变量

SQL> define

DEFINE \_DATE = "11-SEP-14" (CHAR)

DEFINE \_CONNECT\_IDENTIFIER = "epps" (CHAR)

DEFINE \_USER = "SYS" (CHAR)

DEFINE \_PRIVILEGE = "AS SYSDBA" (CHAR)

DEFINE \_SQLPLUS\_RELEASE = "1002000400" (CHAR)

DEFINE \_EDITOR = "ed" (CHAR)

DEFINE \_O\_VERSION = "Oracle Database 10g Release 10.2.0.4.0 - Production" (CHAR)

DEFINE \_O\_RELEASE = "1002000400" (CHAR)

**VARIABLE变量和DEFINE变量不同的是其需要指定变量类型，类似编程语言中的强类型和弱类型变量。前者用于绑定变量，后者是用于&或&&进行变量替换**

SQL> VAR NUM NUMBER;

SQL> EXEC :NUM :=10;

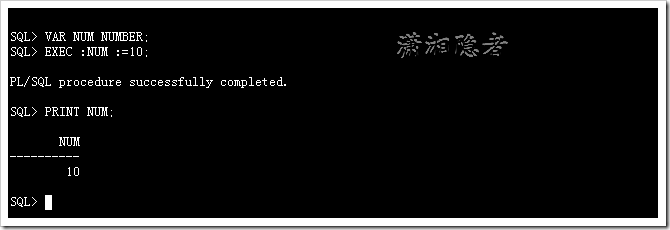
PL/SQL procedure successfully completed.

SQL> PRINT NUM;

NUM

----------

10

[](https://images0.cnblogs.com/blog/73542/201409/112136368402331.png)

DECLARE定义变量则一般用于PL/SQL中。一般大家也用得比较多。在此不做说明。

**参考资料：**

<http://www.doc88.com/p-783671200732.html>

<http://blog.itpub.net/9933980/viewspace-627702/>

<http://blog.csdn.net/haiross/article/details/15340489>

# sqlplus执行脚本文件时如何传参数

2011年11月01日 21:55:33 [yfleng2002](https://me.csdn.net/yfleng2002) 阅读数：11670 标签： [脚本](http://so.csdn.net/so/search/s.do?q=%E8%84%9A%E6%9C%AC&t=blog) [application](http://so.csdn.net/so/search/s.do?q=application&t=blog) [database](http://so.csdn.net/so/search/s.do?q=database&t=blog) [testing](http://so.csdn.net/so/search/s.do?q=testing&t=blog) [oracle](http://so.csdn.net/so/search/s.do?q=oracle&t=blog) [delete](http://so.csdn.net/so/search/s.do?q=delete&t=blog)  更多

个人分类： [ORACLE](https://blog.csdn.net/yfleng2002/article/category/539331)

d:\test.sql脚本如下:

1. select &1 from &2;
2. exit;

执行时这样传参数:sqlplus "scott/tiger@test" @d:\test.sql sysdate dual

**注意:参数必须用&[1-9]表示,不然传不进去,会提示让手动输入参数**

1. C:\>sqlplus "scott/tiger@test" @d:\test.sql sysdate dual
3. D:\>sqlplus "scott/tiger@test" @d:\test.sql sysdate dual
5. SQL\*Plus: Release 11.2.0.1.0 Production on 星期二 11月 1 21:59:00 2011
7. Copyright (c) 1982, 2010, Oracle. All rights reserved.

10. 连接到:
11. Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
12. With the Partitioning, OLAP, Data Mining and Real Application Testing options
14. 原值 1: select &1 from &2
15. 新值 1: select sysdate from dual
17. SYSDATE
18. *--------------*
19. 01-11月-11
21. 从 Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
22. With the Partitioning, OLAP, Data Mining and Real Application Testing options 断
23. 开
25. D:\>

如果脚本中有重复用到相同的值,如果&1=&2:

d:\tes2.sql

1. delete scott.emp where no=&1 or deptno=&2;
2. commit;

执行时,就必须传2个参数:

sqlplus "scott/tiger@test" @d:\test2.sql 10 10

**小窍门: 这时用procedure就可以不用传多个相同的参数,则只用传1个参数:**

d:\test3.sql

1. declare
2. var\_no number:=&1;
3. begin
4. delete scott.emp where no=var\_no or deptno=var\_no;
5. commit;
6. end;

sqlplus "[scott/tiger@test](mailto:scott/tiger@test" \t "_blank)" @d:\test3.sql 10