**Oracle特殊字符转义：&amp;和&#39;**

2015-01-18      [**0**](javascript:void(0);)[个评论](http://www.2cto.com/database/201501/370906.html#SOHUCS)    来源：xuke6677的专栏

[收藏](javascript:;)    [[http://statics.2cto.com/images/dl.jpg](http://www.2cto.com/index.php?m=member&c=content&a=publish&modid=1&siteid=1)我要投稿](http://www.2cto.com/index.php?m=member&c=content&a=publish&modid=1&siteid=1)

我们在SQL\*PLUS下执行 SQL show all命令时，可以发现一个参数：define & (hex 26)，如下所示  
concat . (hex 2e)  
copycommit 0  
copytypecheck ON  
define & (hex 26)  
describe DEPTH 1 LINENUM OFF INDENT OFFecho OFF

**1、”&“ 转义**

这个是[Oracle](http://www.2cto.com/database/Oracle/)里面用来识别自定义变量的设置，现在我们在SQL\*PLUS下将其关闭：  
SQL Set define OFF;  
然后再次执行导入脚本，OK！问题搞定。  
注意：如果是在TOAD中执行，建议在每一个要导入的脚本第一行加上前面那句关闭define的话，否则当你导入第二个含有特殊字符的脚本的时候，又会出错。  
如果是在SQL\*PLUS中执行，则只需要设置一次define OFF，后面就可以连续导入了。直到你重新设置define ON为止。  
·方法二：在SQL语句中将'&'替换成chr(38)，因为chr(38)是‘&’的ASCII码  
SQL Select 'Tom' || chr(38) || 'Jerry' from dual;  
·方法三：分拆原来的字符串  
SQL Select 'Tom' || '&' || 'Jerry' from dual;  
我们可以看到，方法一最为简便，而且效率也最高。方法二因为有一个调用函数的过程，所以性能稍差。方法三需要两次连接字符串，效率最差！

**2、” ' “ 转义**

·方法一：使用转义字符  
SQL Select 'test' || '''' from dual;  
第三个'才是我们真正的内容  
·方法二：同样是使用转义字符，只不过方式不同而已  
SQL Select 'test ''' from dual;  
注意：这里的第二个，第三个'就是我们上面方法一中提到的转义符和真正的内容