

Oracle8i和9i中PLSQL程序不同运行结果 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/275/2021_2022_Oracle8i_E5_92_c67_275707.htm 在把Oracle的数据库从8i升级到9i及以上的时候，一般认为原有的PLSQL程序应该完全兼容，即运行过程和运行结果完全一致。遗憾的是，事实并非如此，由于ORACLE PLSQL引擎的升级，它对某些代码解释做了更改，导致某些代码会有不同的运行结果。各位在升级数据库时必须重视，否则将导致无法估量的损失和难以恢复的灾难。

1、PLSQL表作为参数传递 先看以下代码，在ORACLE 8i和9i中的运行结果。

```
declare type test_rec is record(col_1
varchar2(100)).type test_tbl is table of test_rec index by
binary_integer.l_tbl test_tbl.procedure
change_value is begin l_tbl(1).col_1 := I am changed!.end.procedure
sub_test(pi_str in varchar2) is begin dbms_output.put_line(before:
||pi_str).change_value.dbms_output.put_line(after :
||pi_str).end.begin l_tbl(1).col_1 := I am
ok!.sub_test(l_tbl(1).col_1).exception when others
then dbms_output.put_line(sqlerrm).end.
```

示例代码非常简单，即将PLSQL表的某个成员变量当作参数给另一过程，此过程改变了原PLSQL表的值，但未改变传入参数的值（当然不能改，表示为IN的参数的值是不能改的），观看改变前后，传入参数的值在Oracle 8i和9i的变化。运行结果：Oracle 8i Oracle 9i before: I am ok! before: I am ok! after : I am ok! after : I am changed! 显然运行结果不一样！在Oracle 8i中，传入参数的值在原PLSQL表的值改变前后未变化，而在Oracle 9i中，传入参

数的值被改动了。 Oracle给出的解释是：在Oracle 8i中，所有表示为IN的参数传递都是传值的，包括PLSQL表类型的参数。而到了Oracle 9i，他们觉得PLSQL表类型的参数传递应该传引用，在PLSQL引擎上做了这样的修改，而导致这个问题。我们来回忆以下，传值意味2个变量传递的是真实的数值，各自有不同的内存空间，相当于变量被拷贝了一份，各为其主，互不相干。传引用意味2个变量传递的是内存空间的地址，指向同一块内存空间，如果此内存空间里放的数值被改变了，那么2个变量的值都会被改变。了解问题产生的原因，回头再读前面的示例代码，就比较容易理解了。同样的程序，在数据库升级后产生了不同的运行结果，这个问题的危险程度相信大家一定能明白，必须重视。

2、PLSQL表类型返回值NO_DATA_FOUND意外 看以下代码：

```
DECLARE l_test
VARCHAR2(10).type test_rec is record (col_a
varchar2(100)).TYPE test_tab IS TABLE OF test_rec INDEX BY
BINARY_INTEGER.l_test_tab test_tab.FUNCTION return_tbl (
pi_dummy IN VARCHAR2 )RETURN test_tab IS l_tbl
test_tab.BEGIN l_tbl.DELETE.l_tbl(1).col_a := I am ok!.RETURN
l_tbl.EXCEPTION WHEN OTHERS
THEN l_tbl.DELETE.RETURN l_tbl.END.BEGIN test_tab :=
return_tbl().l_test :=
l_test_tab(1).col_a.DBMS_OUTPUT.PUT_LINE ( before: ||
l_pol_num ).l_test :=
return_tbl()(1).col_a.DBMS_OUTPUT.PUT_LINE ( after : || l_test
).EXCEPTION WHEN NO_DATA_FOUND
THEN DBMS_OUTPUT.PUT_LINE ( NO_DATA_FOUND
```

exception!).END. 这段代码意味某函数返回一个PLSQL表类型的值，然后不同的引用方式，在屏幕上显示。运行结果：
Oracle 8i Oracle 9i before: I am ok! before: I am ok! after : I am ok!
NO_DATA_FOUND exception! 在Oracle 8i中，直接用函数名和下标访问PLSQL表的成员变量是合法的，但到了Oracle 9i，这种方式会导致一个运行期NO_DATA_FOUND意外，而产生不同的运行结果。这个问题Oracle没有给出严格解释，只是指出这样的方式不再合法而已。大家同样需要重视这个问题，以免掉入这个陷阱。数据库的升级导致PLSQL程序有不同的运行结果，这样的问题让人担忧，ORACLE的行事方式让人头疼。若有很多的FORM、REPORT和PLSQL存储过程，这个问题导致的代码检查修改和产生的工作量是非常巨大的，而且后期测试也需要消耗大量的资源。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