# 实验六 存储过程实现

**16337341 朱志儒**

## 实验6.1 存储过程实验

### 实验目的

掌握数据库PL/SQL编程语言，以及数据库存储过程的设计和使用方法。

### 实验要求

存储过程定义、存储过程运行，存储过程更名，存储过程删除，存储过程的参数传递。掌握PL/SQL编程语言和编程规范，规范设计存储过程。

### 实验重点和难点

实验重点：存储过程定义和运行。

实验难点：存储过程的参数传递方法。

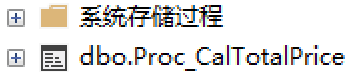
### 实验内容

1. **无参数的存储过程**

* 定义一个存储过程，更新所有订单的总价。

1. **CREATE** **PROCEDURE** Proc\_CalTotalPrice
2. **AS**
3. **UPDATE** Orders **SET** totalprice = (
4. **SELECT** SUM(extendedprice \* (1 - discount) \* (1 + tax))
5. **FROM** Lineitem
6. **WHERE** Orders.orderkey = Lineitem.orderkey);

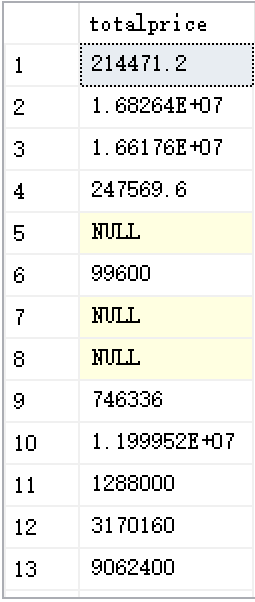
结果：



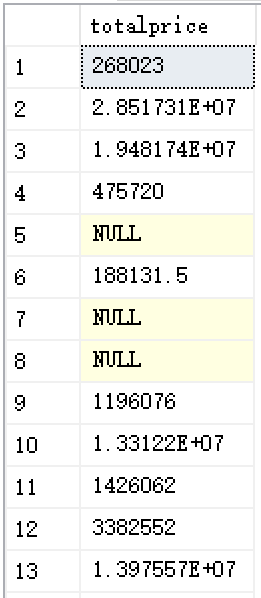
* 执行存储过程。

1. **EXEC** Proc\_CalTotalPrice;

执行前：



执行后：

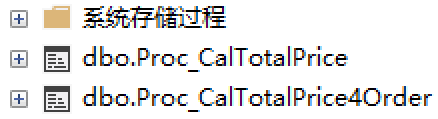


1. **有参数的存储过程**

* 定义一个存储过程，更新给定订单的总价。

1. **CREATE** **PROCEDURE** Proc\_CalTotalPrice4Order(@okey **INT**)
2. **AS**
3. **UPDATE** Orders **SET** totalprice = (
4. **SELECT** SUM(extendedprice \* (1 - discount) \* (1 + TAX))
5. **FROM** Lineitem
6. **WHERE** Orders.orderkey = Lineitem.orderkey AND
7. Lineitem.orderkey = @okey);

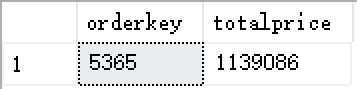
结果：



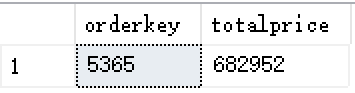
* 执行存储过程。

1. **EXEC** Proc\_CalTotalPrice4Order 5365

执行前：



执行后：

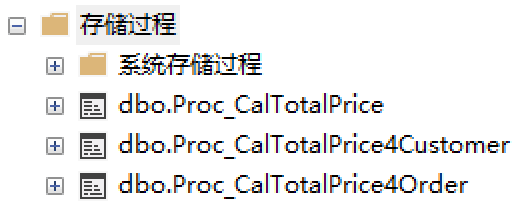


1. **有局部变量的存储过程**

* 定义一个存储过程，更新某个顾客的所有订单的总价。

1. **CREATE** **PROCEDURE** Proc\_CalTotalPrice4Customer(@p\_custname **CHAR**(25))
2. **AS**
3. **DECLARE** @L\_custkey **INT**;
4. **SELECT** @L\_custkey = custkey
5. **FROM** Customer
6. **WHERE** **name** = LTRIM(@p\_custname);
7. **UPDATE** Orders **SET** totalprice = (
8. **SELECT** SUM(extendedprice \* (1 - discount) \* (1 + tax))
9. **FROM** Lineitem
10. **WHERE** Orders.orderkey = Lineitem.orderkey AND
11. Orders.custkey = @L\_custkey);

结果：



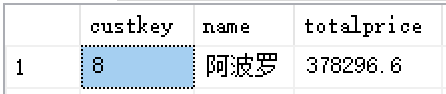
* 执行存储过程。

1. **EXEC** Proc\_CalTotalPrice4Customer '阿波罗'

执行前：



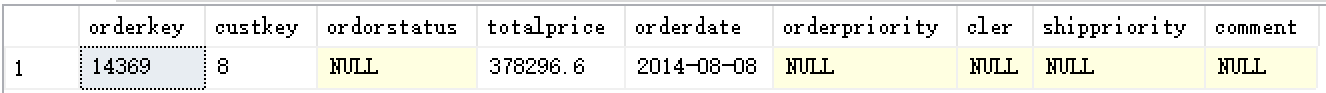
执行后：



* 查看存储过程执行结果

1. **SELECT** \*
2. **FROM** Orders
3. **WHERE** custkey = (
4. **SELECT** custkey
5. **FROM** Customer
6. **WHERE** **name** = '阿波罗');

结果：



1. **有输出参数的存储过程**

* 定义一个存储过程，更新某个顾客的所有订单的总价。

1. **CREATE** **PROCEDURE** Proc\_CalTotalPrice4Customer2(@p\_custname **CHAR**(25), @p\_totalprice **REAL** **OUT**)
2. **AS**
3. **DECLARE** @L\_custkey **INT**;
4. **SELECT** @L\_custkey = custkey
5. **FROM** Customer
6. **WHERE** **name** = LTRIM(@p\_custname);
7. PRINT N'custkey is ';
8. PRINT @L\_custkey;
9. **UPDATE** Orders **SET** totalprice = (
10. **SELECT** SUM(extendedprice \* (1 - discount) \* (1 + tax))
11. **FROM** Lineitem
12. **WHERE** Orders.orderkey = Lineitem.orderkey AND
13. Orders.custkey = @L\_custkey);
14. **SELECT** @p\_totalprice = SUM(totalprice)
15. **FROM** Orders
16. **WHERE** custkey = @L\_custkey;

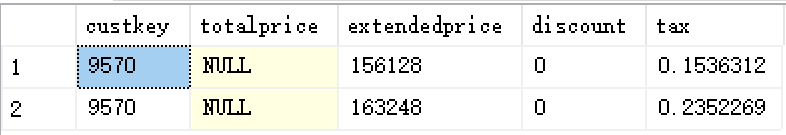
结果：



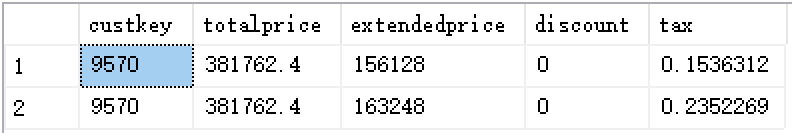
* 执行存储过程。

1. **EXEC** Proc\_CalTotalPrice4Customer2 '曹操', NULL;

执行前：



执行后：





* 查看存储过程执行结果。

1. **SELECT** SUM(totalprice)
2. **FROM** Orders
3. **WHERE** custkey = (
4. **SELECT** custkey
5. **FROM** Customer
6. **WHERE** **name** = '曹操');

结果：



结果与第二步的结果相同。

1. **修改存储过程**

* 修改存储过程。

1. **EXEC** sp\_rename 'Proc\_CalTotalPrice4Order', 'CalTotalPrice4Order'

修改前：



修改后：



* 重新编译存储过程。

1. **EXEC** sp\_recompile N'CalTotalPrice4Order';

结果：



1. **删除存储过程**

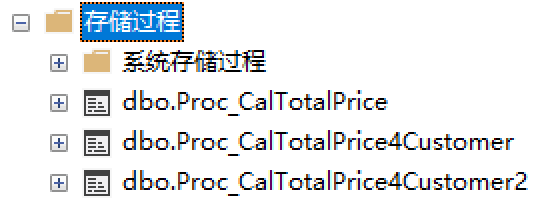
* 删除存储过程。

1. **DROP** **PROCEDURE** CalTotalPrice4Order;

删除前：



删除后：



### 实验总结

存储过程可以通过CALL和SELECT语句调用。

1. 存储过程如果带有OUT或INOUT参数，则参数对应位置在调用时必须使用NULL或其他常量占位。运行所得是一个结果集，结果集有一条或多条RECORD组成，每条RECORD中字段的顺序是OUT或INOUT参数对应的字段在前，最后返回RETURN语句对应的字段。
2. SELECT调用，就是执行普通的SELECT语句。对于存储过程，不能和其他任何常量、函数、存储过程等一并构成表达式使用，只能单独作为一个表达式出现在SELECT语句中。