**数据库实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实 验 人：** | **伍建霖** | | **学 号：** | **20337251** | **日 期：** | **2022.12.18** |
| **院（系）：** | **计算机学院** | | | **专业（班级）：** | **网络空间安全** | |
| **联系方式：** |  | | |  |  | |
| **实验题目：** | | **6.1 存储过程实验** | | | | |

* + - 1. **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。

* + - 1. **实验内容和要求**

存储过程定义，存储过程运行，存储过程删除，存储过程的参数传递。

* + - 1. **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。

* + - 1. **实验工具**

MySQL、SQL Server、Navicat

* + - 1. **实验过程**
* 在MySQL环境下，执行以下操作

(1)无参数的存储过程

①定义一个存储过程，更新所有订单的(含税折扣价)总价。

/\*即根据订单明细表， 计算每个订单的总价， 更新ORDER表\*/

CREATE PROCEDURE Proc\_CalTotalPrice()

BEGIN

UPDATE Orders

SET totalprice= (SELECT SUM(extendedprice\*(1-discount) \*( 1+tax) )

FROM Lineitem

WHERE Orders.orderkey=Lineitem.orderkey) ;

END;

验证截图如下：



②执行存储过程Proc\_CalTotalPrice()

CALL Proc\_CalTotalPrice() ;

(2)有参数的存储过程

①定义一个存储过程，更新给定订单的(含税折扣价)总价。

CREATE PROCEDURE Proc\_Caltotalprice4Order(okey INTEGER)

BEGIN

UPDATE Orders

SET totalprice= (SELECT SUM(extendedprice\*(1-discount) \*(1+tax) )

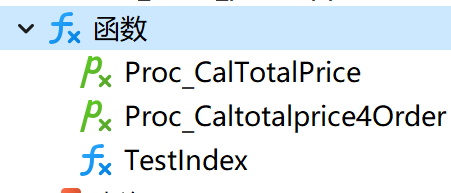
FROM Lineitem

WHERE Orders.orderkey=Lineitem.orderkey)

WHERE orderkey=okey;

END;

验证截图如下：



②执行存储过程。

CALL Proc\_Caltotalprice4Order(5365) ;/\*带参数的调用\*/

(3)有局部变量的存储过程

①定义一个存储过程，更新某个顾客的所有订单的(含税折扣价)总价。

CREATE PROCEDURE Proc\_CalTotalPrice4Customer(p\_custname CHAR(25) )

BEGIN

DECLARE L\_custkey INTEGER;

SELECT custkey INTO L\_custkey/\*查找给定客户名对应的客户编号\*/

FROM Customer

WHERE name=TRIM(p\_custname) ;

/\*TRIM是系统函数， 截去字符串前后空格\*/

UPDATE Orders

SET totalprice=/\*修改指定客户编号的所有订单的总价\*/

(SELECT SUM(extendedprice\*(1-discount) \*(1+tax) )

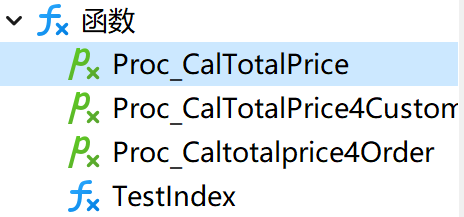
FROM Lineitem

WHERE Orders.orderkey=Lineitem.orderkey)

where custkey=L\_custkey;

END;

验证截图如下：



②执行存储过程Proc\_CalTotalPrice4Customer() 。

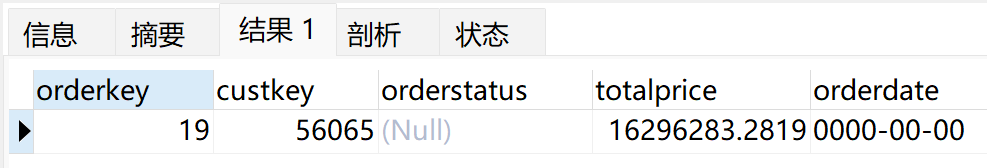
CALL Proc\_CalTotalPrice4Customer('符少静') ;

③查看存储过程执行结果。

SELECT \* FROM Orders

WHERE custkey=(SELECT custkey FROM Customer WHERE name='符少静');

验证截图如下：



(4)有输出参数的存储过程

①定义一个存储过程，更新某个顾客的所有订单的(含税折扣价)总价。

CREATE PROCEDURE Proc\_CalTotalPrice4Customer2(p\_custname CHAR(25) , OUT p\_totalprice REAL)

BEGIN

DECLARE L\_custkey INTEGER;

SELECT custkey INTO L\_custkey/\*查找给定客户名对应的客户编号\*/

FROM Customer

WHERE name=trim(p\_custname) ;

SELECT CONCAT('custkey is ', L\_custkey) into @user\_prompt;

/\*提示客户编号信息\*/

UPDATE Orders

SET totalprice=/\*修改指定客户编号的所有订单的总价\*/

(SELECT SUM(extendedprice\*(1-discount) \*(1+tax) )

FROM Lineitem

WHERE Orders.orderkey=Lineitem.orderkey)

WHERE orderkey=L\_custkey ;

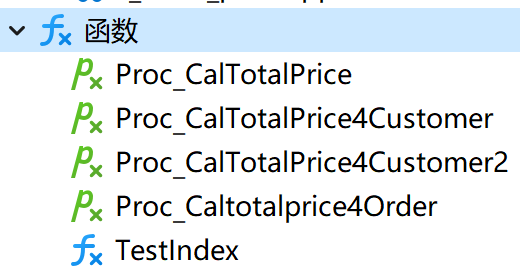
/\*查找指定客户编号的所有订单的总价的和, 并通过输出参数输出该值\*/

SELECT SUM(totalprice) INTO p\_totalprice

FROM Orders WHERE custkey=L\_custkey;

END;

验证截图如下：

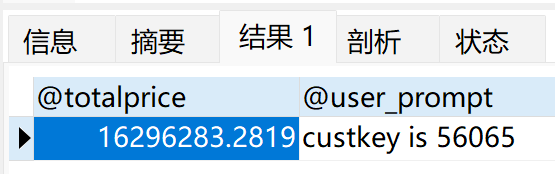


②执行存储过程Proc\_CalTotalPrice4Customer2()

CALL Proc\_CalTotalPrice4Customer2('符少静', @totalprice) ;

select @totalprice,@user\_prompt;

验证截图如下：



③查看存储过程执行结果

/\*检查下列SQL语句执行结果与上述存储过程执行结果是否一致\*/

SELECT SUM(totalprice)

FROM Orders

WHERE custkey=(SELECT custkey FROM Customer WHERE name='符少静') ;

验证截图如下：

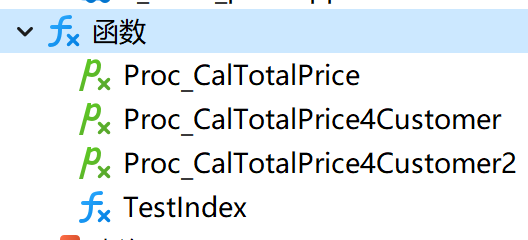


(5)删除存储过程

删除存储过程Proc\_Caltotalprice4Order。

DROP PROCEDURE Proc\_Caltotalprice4Order;

验证截图如下：

****

* + - 1. **与实验结果相关的文件**

无

* + - 1. **思考题**

(1)试总结几种调试存储过程的方法。

答：

1. Select 输出数据调试
2. Debug调试

(2) 存储过程中的SELECT语句与普通的SELECT语句格式有何不同?执行方法有何不同?

答：存储过程中的SELECT语句，可以将查询的结果赋给存储过程中的变量。

* + - 1. **实验总结**

存储过程、用户自定义函数可以通过CALL和SELECT语句调用。需要说明的是：

①存储过程、用户自定义函数如果带有OUT或IN OUT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， 结果集由一条或多条RECORD组成， 每条RECORD中字段的顺序是OUT或IN OUT参数对应的字段在前，最后返回RETURN语句对应的字段。

②SELECT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。对于用户自定义函数， 如果没有OUT或IN OUT参数，可以和其他常量、变量、对象名如字段名等组合成表达式使用。带有OUT或IN OUT参数的函数不可以参与表达式的计算。