**数据库实验报告**

**实 验 人： 凌国明 学 号： 21307077 日 期： 2023/11/30**

**院（系）： 计算机学院 专业（班级）： 计算机科学与技术**

**联系方式： linggm@mail2.sysu.edu.cn**

**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：



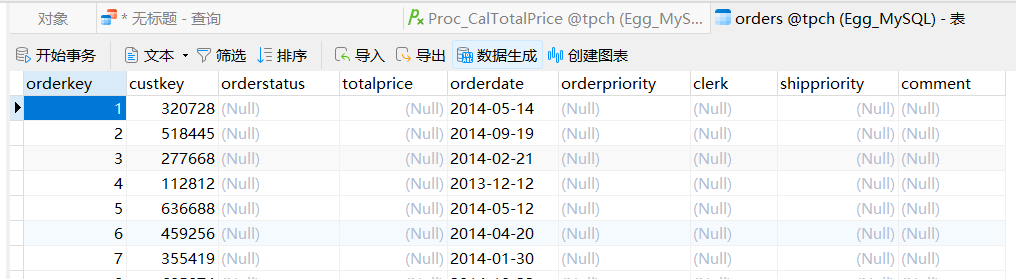
②执行存储过程Proc\_CalTotalPrice()

CALL Proc\_CalTotalPrice() ;

先清空totalprice，然后执行存储过程



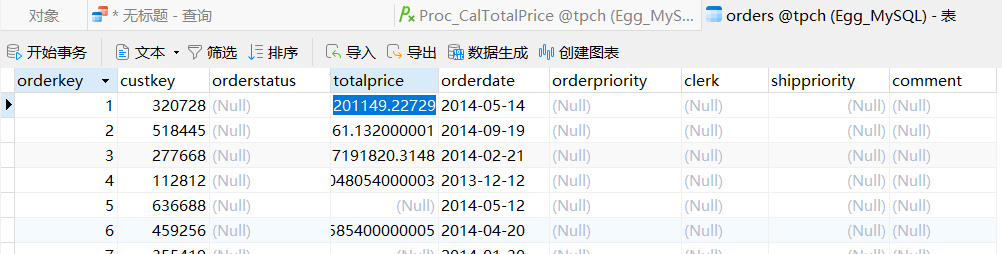
以下是清空后的表



然后执行存储过程



以下是执行存储过程后的表



(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) )

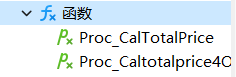
FROM Lineitem

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\*/

先清空totalprice



然后执行存储过程



以下是执行存储过程后的表



(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) )

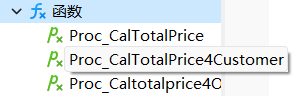
FROM Lineitem

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() 。

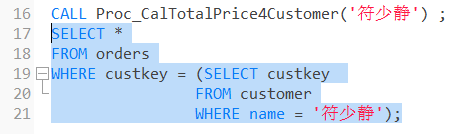
CALL Proc\_CalTotalPrice4Customer('符少静') ;

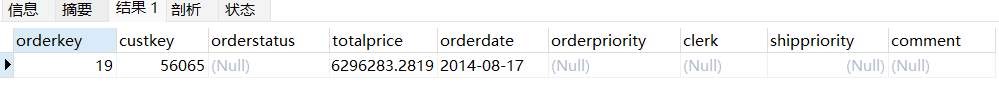
③查看存储过程执行结果。

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 ;

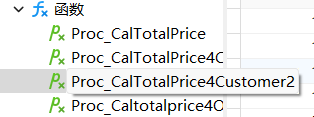
/\*查找指定客户编号的所有订单的总价的和, 并通过输出参数输出该值\*/

SELECT SUM(totalprice) INTO p\_totalprice

FROM Orders WHERE custkey=L\_custkey;

END;

验证截图如下：

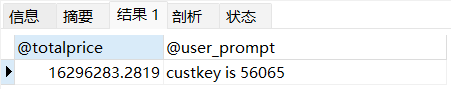


②执行存储过程Proc\_CalTotalPrice4Customer2()

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：

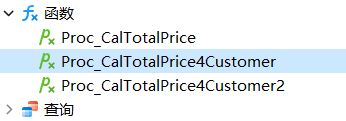
一样

(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：

异常处理：在存储过程中合理使用异常处理（如SQL中的EXCEPTION块）。当程序出错时，捕捉异常并记录错误信息（如错误代码和错误信息），有助于快速定位问题。

单元测试：编写单元测试来验证存储过程的各个部分是否按预期工作。通过为存储过程提供不同的输入参数，并检查输出结果是否正确，可以确保存储过程的可靠性和稳定性。

分段测试：将复杂的存储过程分解成若干个部分，分别进行测试。这种方法可以帮助开发者集中精力在特定的代码段，更容易发现问题。

(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?

结果集的处理：

在普通的SELECT语句中，结果直接返回给客户端或应用程序。

在存储过程中的SELECT语句，结果通常需要存储在局部变量或临时表中，或者使用游标来处理多行结果集。

上下文环境：

存储过程中的SELECT语句执行在特定的过程上下文中，可以访问和修改过程内的局部变量。

普通的SELECT语句通常没有访问存储过程内部变量的权限。

参数化和变量：

在存储过程中，SELECT语句可以使用存储过程的输入、输出参数或局部变量。

普通的SELECT语句一般不直接使用这些参数或变量。

执行方法：

存储过程中的SELECT语句可能涉及更复杂的逻辑处理，比如在SELECT之前或之后加入条件判断、循环等控制结构。

普通的SELECT语句通常是独立执行的，不涉及复杂的程序流程控制。

* + - 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。

在这次的数据库系统实验中，我深入学习了SQL编程语言及其存储过程的设计和使用。通过定义、运行、删除存储过程以及掌握参数的传递，我获得了宝贵的实践经验。特别是在编写具有不同特性的四种存储过程（无参数的过程、有参数的过程、有局部变量的过程、有输出参数的过程）的过程中，我对存储过程的运作机制有了更深刻的理解。

此外，我学到了存储过程中SELECT语句与普通SELECT语句在执行方法和用途上的不同。在存储过程中，SELECT语句可以更灵活地与过程内的其他逻辑和变量结合，实现复杂的数据操作和业务逻辑。

这次实验也让我意识到了存储过程和用户自定义函数在CALL和SELECT语句调用中的区别，尤其是在处理带有OUT或IN OUT参数的情况时。这些知识促进了我对数据库对象的实际应用的理解。