**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： 6.2 自定义函数实验**

* + - 1. **实验目的**

掌握数据库PL/SQL编程语言以及数据库自定义函数的设计和使用方法。

* + - 1. **实验内容和要求**

自定义函数定义，自定义函数运行，自定义函数删除，自定义函数的参数传递。

* + - 1. **实验重点和难点**

实验重点：自定义函数的定义和运行。

实验难点：自定义函数的参数传递方法。

* + - 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)无参数的自定义函数

①定义一个自定义函数，更新所有订单的(含税折扣价)总价，并返回所有订单的总价之和。

/\*该自定义函数与实验6.1中Proc\_CalTotalPrice() 存储过程类似， 区别在于该自定义函数具有一个REAL类型的返回值。\*/

set global log\_bin\_trust\_function\_creators=TRUE;

CREATE FUNCTION FUN\_CalTotalPrice() RETURNs REAL

BEGIN

DECLARE res REAL;

UPDATE Orders

SET totalprice=/\*更新所有订单的含税折扣价总价\*/

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

FROM Lineitem

WHERE Orders.orderkey=Lineitem.orderkey) ;

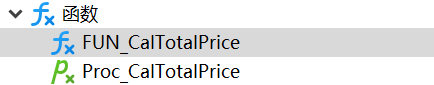
SELECT SUM(totalprice) INTO res/\*计算所有订单的含税折扣价总价之和\*/

FROM Orders;

RETURN res;/\*返回总价之和\*/

END;

验证截图如下：

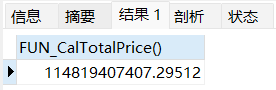


②执行自定义函数 FUN\_CalTotalPrice() 。

SELECT FUN\_CalTotalPrice() ;

/\*执行自定义函数，其返回值以结果集的方式返回和显示。\*/

验证截图如下：



(2)有参数的自定义函数

①定义一个自定义函数，更新并返回给定订单的总价。

CREATE FUNCTION FUN\_CalTotalPrice4Order(p\_okey INTEGER) RETURNs REAL

BEGIN

DECLARE res REAL;

UPDATE Orders

SET totalprice=/\*更新给定编号的订单的含税折扣价总价\*/

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

FROM Lineitem

WHERE Orders.orderkey=Lineitem.orderkey)

WHERE orderkey=p\_okey ;

SELECT totalprice INTO res/\*查找给定订单的总价\*/

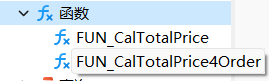
FROM Orders

WHERE orderkey=p\_okey;

RETURN res;/\*返回给定订单的总价\*/

END;

验证截图如下：

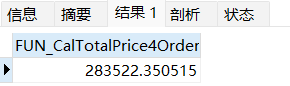


② 执行自定义函数FUN\_CalTotalPrice4Order()

/\*更新并返回5365号订单的总价\*/

select FUN\_CalTotalPrice4Order(5365) ;

验证截图如下：



(3)有局部变量的自定义函数

①定义一个自定义函数，计算并返回某个顾客的所有订单的总价。

CREATE FUNCTION FUN\_CalTotalPrice4Customer(p\_custname CHAR(25) ) RETURNs REAL

BEGIN

DECLARE L\_custkey INTEGER;/\*局部变量L\_custkey\*/

DECLARE res REAL;

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

FROM Customer

WHERE name=trim(p\_custname) ;

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

/\*提示客户编号信息\*/

/\*更新指定客户编号的所有订单的含税折扣价总价\*/

UPDATE Orders

SET totalprice=

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

FROM Lineitem

WHERE Orders.orderkey=Lineitem.orderkey)

WHERE Orders.custkey=L\_custkey ;

/\*计算指定客户编号的所有订单的含税折扣价总价之和\*/

SELECT SUM(totalprice) INTO res

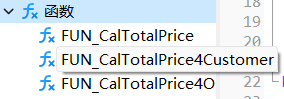
FROM Orders

WHERE custkey=L\_custkey;

RETURN res;/\*返回总价之和\*/

END;

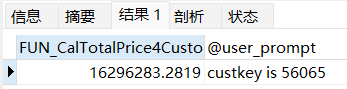
验证截图如下：



②执行自定义函数FUN\_CalTotalPrice4Customer() 。

SELECT FUN\_CalTotalPrice4Customer('符少静') ,@user\_prompt;

验证截图如下：



(4)有输出参数的自定义函数

①定义一个自定义函数,计算并返回某个顾客的所有订单的总价。

/\*该函数定义一个输入参数p\_custname, 一个输出参数p\_totalprice, 还有一个返回值类型REAL, 通过输出参数的定义, 该函数可以返回两个或者两个以上的值。该函数与FUN\_CalTotalPrice4Customer()基本类似,区别只在于该函数多了一个输出参数。\*/

CREATE FUNCTION FUN\_CalTotalPrice4Customer2(p\_custname CHAR(25) , out p\_totalprice REAL) RETURNs REAL

BEGIN

DECLARE L\_custkey INTEGER;

DECLARE res REAL;

SELECT custkey INTO L\_custkey

FROM Customer

WHERE name=trim(p\_custname) ;

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

UPDATE Orders

SET totalprice=

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

FROM Lineitem

WHERE Orders.orderkey=Lineitem.orderkey)

WHERE custkey=L\_custkey;

SELECT SUM(totalprice) IN TOp\_totalprice

FROM Orders

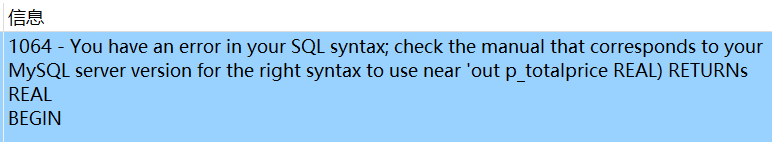
WHERE custkey=L\_custkey;

set Res = p\_totalprice;

RETURN res;

END;

验证截图如下：

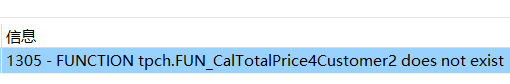


根据出错信息，可知：在MySql中，自定义函数不能拥有输出参数，这是因为自定义函数自身就是输出参数。

②执行自定义函数FUN\_CalTotalPrice4Customer2() 。

注：在MySql中，以下语句失败。

SELECT FUN\_CalTotalPrice4Customer2('符少静', @totalprice) ;

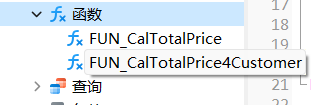


(5)删除自定义函数

删除自定义函数FUN\_CalTotalPrice4Order。

DROP FUNCTION FUN\_CalTotalPrice4Order;

验证截图如下：



* + - 1. **与实验结果相关的文件**

无

* + - 1. **思考题**

(1)试分析自定义函数与存储过程的区别与联系。

答：

目的相似：它们都旨在封装数据库逻辑，使代码更加模块化和可重用。通过将复杂的操作封装在函数或过程中，可以简化数据库操作，并在整个数据库系统中重复使用这些逻辑。

以下是他们的区别：

功能和用途：

自定义函数通常用于计算并返回单个值，它可以嵌入在SQL语句中，如在SELECT、WHERE或CASE语句中使用。

存储过程更加复杂，可以执行一系列的操作，包括更新数据、处理事务等，并且可以返回多个值和结果集。

返回值：

自定义函数必须返回一个值，这是其定义的一部分。

存储过程可以不返回值，或者通过OUT参数返回多个值。

使用场景：

自定义函数适用于需要计算并返回结果的场景，例如，计算某个字段的转换值。

存储过程适用于更复杂的业务逻辑，如数据的批量处理、复杂的计算和逻辑判断。

调用方式：

自定义函数可以直接在SQL语句中调用。

存储过程通常需要独立调用，使用CALL语句或特定的程序调用接口。

(2)如何使得自定义函数可以返回多个值?如何利用?

答：

从“有输出参数的自定义函数”部分的失败案例，得知MySQL中的自定义函数不支持OUT或IN OUT类型的参数，这是因为函数本身就是用来返回一个值的。

可以采用以下方法来实现

使用复合类型返回值：如果数据库支持，可以定义一个复合类型（如记录或对象）作为函数的返回类型，从而可以包含多个值。

拆分为多个函数：如果逻辑允许，可以将一个复杂的函数拆分为多个简单的函数，每个函数返回一部分所需数据。这样可以保持函数的单一职责原则，同时获得所需的所有信息。

使用临时表或表变量：在一些数据库系统中，可以利用临时表或表变量来存储并返回多个值。函数可以向这些表中插入数据，然后在函数之外查询这些表以获取结果。

转换为存储过程：如果需要返回多个值，可以考虑将逻辑转换为存储过程，因为存储过程可以通过OUT或IN OUT参数返回多个值。

* + - 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。

在这次实验中，我深入学习了PL/SQL编程语言和数据库自定义函数的设计与使用。实验内容涉及自定义函数的定义、运行、删除以及参数传递，特别是在编写四种不同类型的函数（无参数的自定义函数、有参数的自定义函数、有局部变量的自定义函数、有输出参数的自定义函数）过程中，我对自定义函数的灵活性和功能性有了更加深刻的理解。

通过实验，我认识到自定义函数在数据库编程中的重要性。它们不仅提高了代码的复用性，还使得数据操作逻辑更加清晰和集中。我特别感兴趣的是参数传递的方法，这对于编写高效且可维护的函数至关重要。通过实践，我学会了如何合理地使用参数来提高函数的灵活性和通用性。

此外，我还学习了自定义函数和存储过程之间的区别与联系。虽然它们在很多方面有共同点，但在使用场景、返回值和性能上有明显的不同。我意识到，在实际应用中，根据需求选择合适的工具非常重要。

总的来说，这次“自定义函数实验”不仅增强了我的数据库编程技能，也让我对函数在数据库系统中的作用有了更全面的认识。通过实际操作和探索，我对如何在复杂的数据库环境中有效使用自定义函数有了更清晰的了解。