

# 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:

+ 系统存储过程  
+ `dbo.Proc_CalTotalPrice`

- 执行存储过程。

1. `EXEC Proc_CalTotalPrice;`

执行前:

	totalprice
1	214471.2
2	1.68264E+07
3	1.66176E+07
4	247569.6
5	NULL
6	99600
7	NULL
8	NULL
9	746336
10	1.199952E+07
11	1288000
12	3170160
13	9062400

执行后:

	totalprice
1	268023
2	2.851731E+07
3	1.948174E+07
4	475720
5	NULL
6	188131.5
7	NULL
8	NULL
9	1196076
10	1.33122E+07
11	1426062
12	3382552
13	1.397557E+07

## 2) 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:

```

+ 系统存储过程
+  dbo.Proc_CalTotalPrice
+  dbo.Proc_CalTotalPrice4Order

```

- 执行存储过程。

```
1. EXEC Proc_CalTotalPrice4Order 5365
```

执行前:

	orderkey	totalprice
1	5365	1139086

执行后:

	orderkey	totalprice
1	5365	682952

### 3) 有局部变量的存储过程

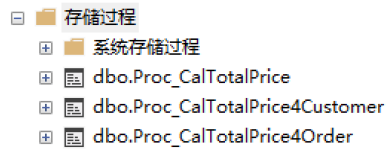
- 定义一个存储过程，更新某个顾客的所有订单的总价。

```

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:

	custkey	name	totalprice
1	8	阿波罗	NULL

执行后:

	custkey	name	totalprice
1	8	阿波罗	378296.6

- 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:

	orderkey	custkey	ordorstatus	totalprice	orderdate	orderpriority	cler	shippriority	comment
1	14369	8	NULL	378296.6	2014-08-08	NULL	NULL	NULL	NULL

#### 4) 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_total
   price 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:

	custkey	totalprice	extendedprice	discount	tax
1	9570	NULL	156128	0	0.1536312
2	9570	NULL	163248	0	0.2352269

执行后:

	custkey	totalprice	extendedprice	discount	tax
1	9570	381762.4	156128	0	0.1536312
2	9570	381762.4	163248	0	0.2352269

	(无列名)
1	381762.4375

- 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：

	(无列名)
1	381762.4375

结果与第二步的结果相同。

## 5) 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：

已成功地标记对象 'CalTotalPrice4Order'，以便对它重新进行编译。

## 6) 删除存储过程

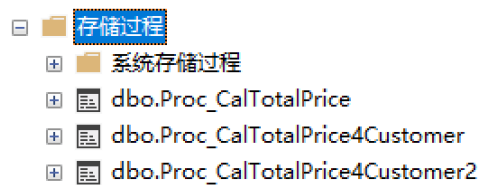
- 删除存储过程。

```
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。