# 实验五 存储过程和触发器

一、实验目的

(1) 通过实践理解存储过程和触发器的概念、作用及优点；

(2) 掌握存储过程的定义与调用，实现存储过程中带有不同参数的应用；

(3) 掌握创建触发器。

二、实验原理

**1.存储过程**

一个被命名的存储在服务器上的T-SQL语句的集合，是封装重复性工作的一种方法。

（1）创建存储过程

CREATE PROC[DURE]procdure\_name [{@parament data\_type}

[VARYING][=default][OUTPUT]] [,…n1]

AS sql\_statement

**procedure\_name**:新存储过程的名称，必须符合标识符规则且唯一。

**@parameter**:过程中的参数。可以声明一个或多个参数。用户必须在执行过程时提供每个所声明参数的值（除非定义了该参数的默认值）。使用 @ 符号作为第一个字符来指定参数名称。参数名称须符合标识符规则。每个过程的参数仅用于该过程本身；相同的参数名称可用在其它过程中。默认情况下参数只能代替常量，不能代替表名、列名或其它数据库对象名称。

**data\_type**:参数的数据类型。

**default:**参数的默认值。如果定义了默认值，不必指定该参数的值即可执行过程。默认值必须是常量或 NULL。

**OUTPUT**:表明参数是返回参数。该选项的值可以返回给 EXEC[UTE]。使用 OUTPUT 参数可将信息返回给调用过程。

（2）执行存储过程

SQL Server系统中，可以使用EXECUTE语句执行存储过程。EXECUTE语句也可以简写为EXEC。如果将要执行的存储过程需要参数，那么应该在存储过程名称后面带上参数值。

[EXEC[UTE]]{[@return\_status=]

{procedure\_name[;number]|@procedure\_name\_var}

[@parameter={value|@variable[OUTPUT]|[DEFAULT]}

[,…n]

(3) 删除存储过程

使用DROP PROCEDURE语句可永久地删除存储过程。

DROP PROCEDURE{procedure}[,…n]

**2.触发器**

一种特殊类型的存储过程，可实现自动化操作，主要是实现主键和外键所不能保证的复杂的参照完整性和数据一致性。触发器与存储过程主要的区别在于触发器的运行方式。存储过程必须由用户、应用程序或者触发器来显式地调用并执行，而触发器是当特定事件出现的时候自动执行的，与连接到数据库中的用户或者应用程序无关。

(1)创建触发器

CREATE TRIGGER trigger\_name ON { table|view }

{{{FOR|AFTER|INSTEAD OF}{[INSERT][,][UPDATE]}

AS sql\_statement [ ...n ]}}

**trigger\_name**:触发器的名称。

**Table | view:**在其上执行触发器的表或视图，有时称为触发器表或触发器视图。

AFTER:指定触发器只有在触发SQL语句中指定的所有操作都已成功执行后才激发。如果仅指定FOR关键字，则AFTER是默认设置。

INSTEAD OF:指定执行触发器而不是执行触发 SQL 语句，从而替代触发语句的操作。

**{ [DELETE] [,] [INSERT] [,] [UPDATE] }**:是指定在表或视图上执行哪些数据修改语句时将激活触发器的关键字。必须至少指定一个选项。在触发器定义中允许使用以任意顺序组合的这些关键字。如果指定的选项多于一个，需用逗号分隔这些选项。

**AS**:是触发器要执行的操作。

**sql\_statement**:是触发器的条件和操作。触发器条件指定其它准则，以确定DELETE、INSERT或UPDATE语句是否导致执行触发器操作。当尝试DELETE、INSERT或UPDATE操作时，Transact-SQL语句中指定的触发器操作将生效。

**n**:是表示触发器中可以包含多条Transact-SQL语句的占位符。

(2)删除触发器

DROP TRIGGER {trigger}[,…n]

三、实验示例

**1.存储过程的应用**

【例1】使用不带任何参数的存储过程。从GZGL库中的三个表中查询，返回学生学号、姓名、课程名、成绩、学分。

use xscj

go

create procedure student\_info as

select a.学号,姓名,课程名,成绩,学分 from xs a inner join xs\_kc b

on a.学号= b.学号 inner join kc c on b.课程号= c.课程号

在GZGL库中创建好该存储过程后，可以通过下面的语句执行该存储过程，则会返回查询结果。

exec student\_info

【例2】使用带参数的存储过程。从XSCJ的三个表中查询某个人指定课程的成绩和学分。

create procedure student\_info1 @name char(8),@cname char(20) as

select a.学号,姓名,课程名,成绩,c.学分 from xs a inner join xs\_kc b

on a.学号= b.学号 inner join kc c on b.课程号= c.课程号

where a.姓名=@name and c.课程名=@cname

可以采取下面两种方式运行此存储过程：

exec student\_info1 '王林','网络编程' 或

exec student\_info1 @name='李丽',@cname='网络数据库'

**2.触发器的应用**

【例3】创建一个名为tr\_age 的触发器，要求在插入和更新时检查AGE 是否在15 到60 之间，如不在15 到60，则弹出“年龄不合法”提示信息，SQL 代码如下所示：

CREATE TRIGGER tr\_age ON members

FOR INSERT,UPDATE

AS

DECLARE @age INT

SELECT @age = YEAR(GETDATE())-YEAR(m\_birth)

FROM inserted

IF @age NOT BETWEEN 15 AND 60

BEGIN

ROLLBACK TRANSACTION

PRINT '年龄不合法'

END

上面代码中的ROLLBACK TRANSACTION表示撤销插入或更新操作。

创建完触发器tr\_age 后，再按F5 执行以下更新数据表memebers 的SQL 代码：

UPDATE members SET m\_birth='2000-6-4' WHERE m\_account='Lfz'

执行此段SQL 更新记录代码后，将在结果框中显示以下消息，表示触发了数据表memebers 中的触发器tr\_age：



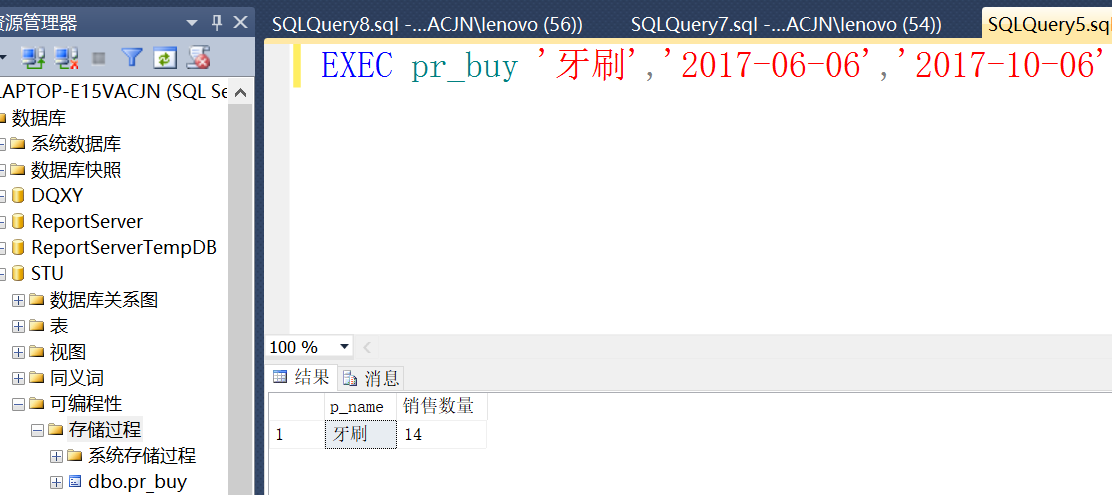
图6-1 更新操作执行后提示错误消息

四、实验内容

**1. 创建存储过程pr\_buy， 返回指定商品（p\_name ）在某指定时间段（o\_date）的销售量（o\_quantity）。**

**2. 查看存储过程。**

**3. 执行存储过程pr\_buy 查询某商品在指定日期之间的销售量。（统计商品的名称、日期和数量不必与下图一致，查询合理即可）**

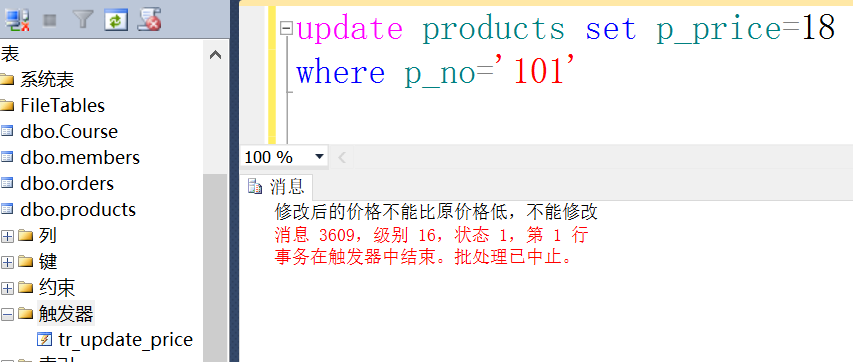


**4. 在表products中建立更改触发器tr\_update\_price，实现修改后的商品价格不能低于原价格。如果修改后的价格低于原价格，则输出’修改后的价格不能比原价格低，不能修改’的提示信息，修改失败。**

**提示：更新的动作实际分2步，先delete ,再insert，所以更新前的值通过 deleted获取，更新后的值通过 inserted获取，如SELECT @price = p\_price FROM inserted获取更新后的值。**

**5. 查看触发器。**

**6. 通过sql语句更新商品价格，测试tr\_update\_price 触发器的功能。（测试商品及价格不必与下图一致）**



五、思考题

1. 存储过程与触发器的区别是什么？