

实验六 触发器

一、实验目的

- 1、了解 SQL Server 的触发器类型
- 2、掌握创建各种触发器的方法
- 3、掌握使用每种触发器的方法
- 4、掌握简单系统的实现方法

二、预备知识

触发器是一种特殊的存储过程，它与我们前面介绍的存储过程是不同的。触发器主要是通过事件进行触发而被执行的，而存储过程可以通过存储过程名而被直接调用。当对某一基本表进行诸如 insert、update、delete 这些操作时，SQL Server 就会自动执行触发器所在、定义的 SQL 语句，从而保证对数据的处理必须符合由这些 SQL 语句所定义的规则。对应的触发器有三种：insert 触发器、update 触发器、delete 触发器。

触发器的主要作用就是其能够实现由主码和外码所不能保证的复杂的参照完整性和数据的一致性。除此之外，触发器还有其它许多不同的功能：

- 强化约束：实现比 check 语句更为复杂的约束
- 跟踪变化：通过检测数据库内的操作，拒绝未经许可的数据库更新。
- 级联运行：自动级联影响整个数据库的各项内容。例如，某个表上的触发器中包含有对另外一个表的数据操作（如删除、插入或更新），而该操作又导致该表上触发器被触发。
- 调用存储过程：触发器可以调用一个或多个存储过程。这样就可以解决高级形式的业务规则或复杂行为限制以及实现定制记录等一些方面的问题。例如，触发器能够找出某一表在数据修改前后状态发生的差异，并根据这种差异执行一定的处理。

在使用触发器的过程中，SQL Server 的每个触发器有两个特殊的表：插入表 (inserted 表) 和删除表 (deleted 表)。这是两个由系统管理的临时表，它存储在内存中，而不是存储在数据库中，所以不允许用户直接对其修改。而且当触发器工作完成，这两个表也被删除。

这两个表的结构总是与该触发器作用的表有相同的表结构。这两个主要保存因用户操作而被影响到的原数据值或新数据值。这两个表是只读的，即不允许用户直接修改这两个表的内容，但可以引用两个表的数据。

在这两个表中存放的数据：

- 对一个定义了插入类型触发器的表来讲，一旦对该表执行了插入操作，那么对向该表插入的所有行来说，都有一个相应的副本存放到插入表中。
- 对一个定义了删除类型触发器的表来讲，一旦对该表执行了删除操作，则所有的删除行的存放至删除表中。这样当需要时，可以从删除表中恢复。
- 更新操作包括两个部分，即先将更新的内容去掉，然后将新值插入。因此对一个定义了更新类型触发器的表来讲，一旦对该表执行了更新操作时，在删除表中存放了旧值，然后在插入表中存放新值。

三、实验示例

【例 15-1】创建一个触发器 t_updae，当表 S_1 发生修改时，显示修改前后的记录

```
use Stu_Cou
if exists (select name from sysobjects
           where name = 't_update' and type = 'tr')
    drop trigger t_update
go
create trigger t_update
on S_1 for update
as
begin
    select * from inserted
    select * from deleted
end
--表 inserted 包含修改后的记录；表 deleted 包含修改前的记录
go
```

【例 15-2】将 S_1 表中学号为 '95001' 的学生的已修学分修改为 29，查看触发器的执行情况

```
use Stu_Cou
update S_1
set Spre=29 where Sno='95001'
```

【例 15-3】创建一个触发器 t_insert，向表 SC_1 中插入新的一行，显示新插入的记录

```
use Stu_Cou
if exists (select name from sysobjects
           where name = 't_insert' and type = 'tr')
    drop trigger t_insert
go
create trigger t_insert
on SC_1 for insert
as
select * from inserted
--使用 inserted 来引用一个临时驻留在内存的表，该表包括了新插入的记录
go
```

【例 15-4】向表 SC_1 中插入 ('95001', '7', 80)，查看触发器的执行情况

```
use Stu_Cou
insert into SC_1
values( '95001', '7', 80 )
```

【例 15-5】建立一个触发器 t_delete，删除表 SC_1 中某些记录时显示被删除的记录

```
use Stu_Cou
if exists (select name from sysobjects
           where name = 't_delete' and type = 'tr')
```

```

drop trigger t_delete
go
create trigger t_delete
on SC_1 for delete
as
select * from deleted
--使用 deleted 来引用一个临时驻留在内存的表，该表包括被删除的记录
go

```

【例 15-6】删除 SC_1 表中学号为 ‘95001’ 的学生记录，查看触发器的情况

```

use Stu_Cou
delete from SC_1
where Sno='95001'

```

【例 15-7】为了保证参照完整性，建立触发器 s_sc，删除 S_1 表中某些记录的同时也删除 SC_1 表中的相应记录(为什么?)

```

use Stu_Cou
if exists (select name from sysobjects
           where name = 's_sc' and type = 'tr')
drop trigger s_sc
go
create trigger s_sc
on S_1 for delete
as
begin
    -- delete SC_1 表示从表 SC_1 中删除记录
    delete SC_1 from SC_1, deleted
    where SC_1.Sno = deleted.Sno

    select * from deleted
    select * from SC_1
end

```

【例 15-8】删除表 S_1 中计算机系学生的记录，检查触发器的情况

```

use Stu_Cou
delete from S_1
where Sdept='计算机'

```

[例 15-9] 创建一个触发器，当向 SC 表插入有一元组时，检查该元组的学号在 Student 表中是否存在，且课程号在 Course 表中是否存在，若有一项不存在，则不允许插入。

```

use Stu_Cou
if exists (select name from sysobjects)
    where name='check_insert' and type='tr')
drop trigger check_insert

```

```

go
create trigger check_insert
on SC for insert
as
if exists(select * from inserted A
where A.Sno NOT IN
(select B.Sno from student B) or A.Cno NOT IN (select c.cno from course C))
begin
RAISERROR('违背数据的完整性',16,1)
ROLLBACK TRANSACTION
--回滚到旧状态
end
go

```

如果执行语句:

insert SC values('96003', '6', 88), 执行结果是什么? 多测试几条数据。

[例 15-10]创建一触发器 check_score, 检查插入的成绩是否在 0 到 100 之间。

```

use Stu_Cou
if exists(select name from sysobjects
where name='check_score' and type='tr')
drop trigger check_score
go
create trigger check_score
on SC for insert, update --为什么写 insert, update
as
declare @score int
select @score=grade from inserted
if @score<0 or @score>100
begin
RAISERROR('成绩必须在 0 到 100 之间', 16, 1)
ROLLBACK TRANSACTION
end
go

```

如果执行语句:

insert SC values('96002' , '6', 120), 执行结果是什么?

[例 15-11] 创建一 delete 触发器 delete_trigger, SC1 表中包含了学生的选课情况。如果存在一个 S1 表, 其中包含了学生的基本情况, 如果要删除 S1 中的学生记录, 那么其相应的学生选课也就随之删除。

```

use Stu_Cou
select * into S1
from Student
go

```

```
select * into SC1  
from SC  
go
```

```
if exists (select name from objects  
          where name='delete_trigger' and type='tr')  
  drop trigger delete_trigger  
go  
create trigger delete_trigger  
on S2 for delete  
as  
  delete SC2 from sc2,deleted where SC2.Sno=deleted.Sno  
go
```

如果执行语句：

就可以看到 SC2 中对应的记录也被删除了。

```
delete S2 where Sno='95001'  
go  
select * from SC2
```

四、习题

- 1、触发器是什么？有什么用途？
- 2、创建一个触发器，如果对不及格的成绩进行修改，给出提示信息并取消操作。