[**Sql Server触发器**](http://www.cnblogs.com/sydeveloper/archive/2012/12/03/2799905.html)

一、触发器是一种特殊的存储过程，不能被显式调用，只能在对表进行insert、update、delete操作时被自动激活。所以触发器可以用来实现对表进行复杂的完整性约束。

二、 Sql Server为每个触发器都创建了两个专用表：Inserted表和Deleted表。这两个表由系统来维护，它们存在于内存中而不是数据库中。这两个表的结构总是与被该触发器作用的表的结构相同，触发器执行完成后，与该触发器相关的这两个表也被删除。

|  |  |  |
| --- | --- | --- |
| 对表的操作 | Inserted逻辑表 | Deleted逻辑表 |
| 增加记录（insert） | 存放增加的记录 | 无 |
| 删除记录（delete） | 无 | 存放被删除的记录 |
| 修改记录（update） | 存放更新后的记录 | 存放更新前的记录 |

三、for、after、instead of触发器

after：触发器在触发它们的语句完成后执行。如果该语句因错误而失败，触发器将不会执行。不能为视图指定after触发器，只能为表指定该触发器。可以为每个触发操作（insert、update、delete）指定多个after触发器。如果表有多个after触发器，可使用sp\_settriggerorder定义哪个after触发器最先激发，哪个最后激发。除第一个和最后一个触发器外，所有其他的after触发器的激发顺序不确定，并且无法控制。

for：等同于after

instead of：该触发器代替触发操作执行。可在表和视图上指定instead of触发器。只能为每个触发操作（insert、update、delete）定义一个instead of触发器。instead of触发器可用于对insert和update语句中提供的数据值执行增强的完整性检查。

四、 触发器的使用

1、创建触发器：

create trigger trigger\_name

on {table\_name | view\_name}

{for| After | Instead of }

[ insert, update,delete ]

as

sql\_statement

2、删除触发器：

drop trigger trigger\_name

3、查看数据库中已有的触发器：

 select \* from sysobjects where xtype='TR'

4、查看单个触发器：

exec sp\_helptext '触发器名'

5、修改触发器：

alter trigger trigger\_name

on {table\_name | view\_name}

{for| After | Instead of }

[ insert, update,delete ]

as

sql\_statement

五、触发器的实例

1、在Orders表中建立触发器，当向Orders表中插入一条记录时，检查goods表的货品状态status是否为1。是，则不能往Orders表加入该订单。

if (object\_id('tgr\_orders\_insert', 'tr') is not null)  
    drop trigger [tgr\_orders\_insert];  
go  
create trigger [tgr\_orders\_insert]  
on [orders]  
after insert  
as   
if (select [status] from [goods],[inserted] where [goods].name=[inserted].name)=**1**  
begin  
    print 'the goods is being processed'  
    print 'the order cannot be committed'      
    rollback transaction   --回滚﹐避免加入

end