**SQL学习之触发器**

1. **DML触发器**

DML触发器是一些附加在特定表或视图上的操作代码，当数据库服务器中发生数据操作语言事件时执行这些操作。SqlServer中的DML触发器有三种：

1. insert触发器:向表中插入数据时被触发；
2. delete触发器：从表中删除数据时被触发；
3. update触发器：修改表中数据时被触发。

**当遇到下列情形时，应考虑使用DML触发器：**

1. 通过数据库中的相关表实现级联更改
2. 防止恶意或者错误的insert、update和delete操作，并强制执行check

约束定义的限制更为复杂的其他限制。

1. 评估数据修改前后表的状态，并根据该差异才去措施。
2. **DDL触发器**

DDL触发器是当服务器或者数据库中发生数据定义语言(主要是以create,drop,alter开头的语句)事件时被激活使用，使用DDL触发器可以防止对数据架构进行的某些更改或记录数据中的更改或事件操作。

1. **登录触发器**

登录触发器将为响应 LOGIN 事件而激发存储过程。与 SQL Server 实例建立用户会话时将引发此事件。登录触发器将在登录的身份验证阶段完成之后且用户会话实际建立之前激发。因此，来自触发器内部且通常将到达用户的所有消息（例如错误消息和来自 PRINT 语句的消息）会传送到 SQL Server 错误日志。如果身份验证失败，将不激发登录触发器。

**触发器的工作原理：**

触发器触发时：

1. 系统自动在内存中创建deleted表或inserted表；
2. 只读，不允许修改，触发器执行完成后，自动删除。

inserted表：

1. 临时保存了插入或更新后的记录行；
2. 可以从inserted表中检查插入的数据是否满足业务需求；
3. 如果不满足，则向用户发送报告错误消息，并回滚插入操作。

deleted表：

1. 临时保存了删除或更新前的记录行；
2. 可以从deleted表中检查被删除的数据是否满足业务需求；
3. 如果不满足，则向用户报告错误消息，并回滚插入操作。



创建触发器的语法：

CREATE TRIGGER trigger\_name

ON table\_name

[WITH ENCRYPTION]

FOR | AFTER | INSTEAD OF [DELETE, INSERT, UPDATE]

AS

T-SQL语句

GO

--with encryption 表示加密触发器定义的sql文本

--delete,insert,update指定触发器的类型

测试数据

--创建学生表

create table student(

stu\_id int identity(1,1) primary key,

stu\_name varchar(10),

stu\_gender char(2),

stu\_age int

)

创建insert触发器：

--创建insert触发器

create trigger trig\_insert

on student

after insert

as

begin

if object\_id(N'student\_sum',N'U') is null--判断student\_sum表是否存在

create table student\_sum(stuCount int default(0));--创建存储学生人数的student\_sum表

declare @stuNumber int;

select @stuNumber = count(\*)from student;

if not exists (select \* from student\_sum)--判断表中是否有记录

insert into student\_sum values(0);

update student\_sum set stuCount =@stuNumber; --把更新后总的学生数插入到student\_sum表中

End

--测试触发器trig\_insert-->功能是向student插入数据的同时级联插入到student\_sum表中，更新stuCount

--因为是后触发器，所以先插入数据后，才触发触发器trig\_insert;

insert into student(stu\_name,stu\_gender,stu\_age)values('吕布','男',30);

select stuCount 学生总人数 from student\_sum;

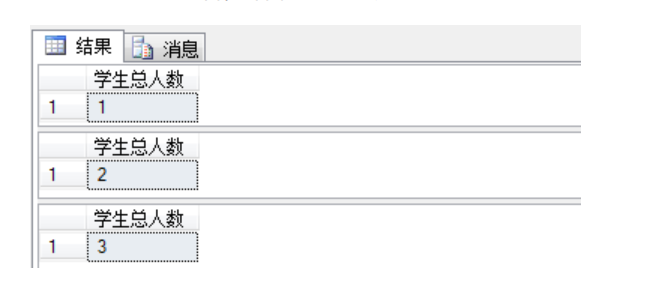
insert into student(stu\_name,stu\_gender,stu\_age)values('貂蝉','女',30);

select stuCount 学生总人数 from student\_sum;

insert into student(stu\_name,stu\_gender,stu\_age)values('曹阿瞒','男',40);

select stuCount 学生总人数 from student\_sum;

执行上面的语句，结果如图所示



 既然定义了学生总数表student\_sum表是向student表中插入数据后才计算学生总数的，所以学生总数表应该禁止用户向其中插入数据

--创建insert\_forbidden,禁止用户向student\_sum表中插入数据

create trigger insert\_forbidden

on student\_sum

after insert

as

begin

RAISERROR('禁止直接向该表中插入记录，操作被禁止',1,1)--raiserror 是用于抛出一个错误

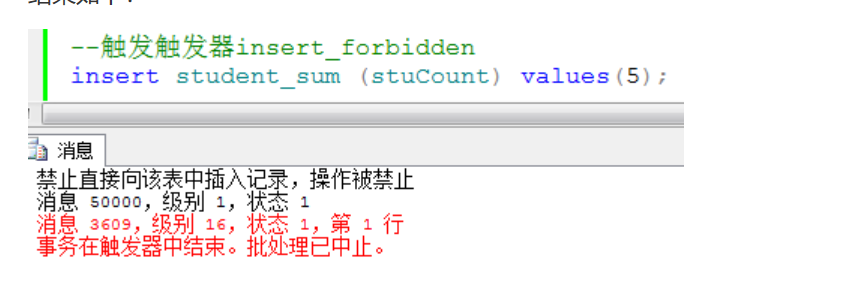
rollback transaction

end

--触发触发器insert\_forbidden

insert student\_sum (stuCount) values(**5**);

结果如下：



创建delete触发器

用户执行delete操作，就会激活delete触发器，从而控制用户能够从数据库中删除数据记录，触发delete触发器后，用户删除的记录会被添加到deleted表中，原来表的相应记录被删除，所以在deleted表中查看删除的记录。

--创建delete触发器

create trigger trig\_delete

on student

after delete

as

begin

select stu\_id as 已删除的学生编号,stu\_name stu\_gender,stu\_age

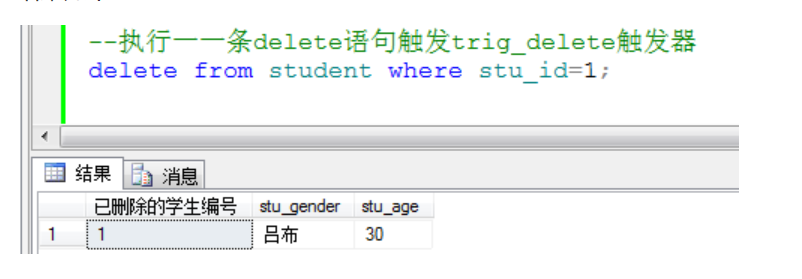
from deleted

end;

--执行一一条delete语句触发trig\_delete触发器

delete from student where stu\_id=**1**;

结果如下所示：



创建update触发器

update触发器是当用户在指定表上执行update语句时被调用被调用，这种类型的触发器用来约束用户对数据的修改。update触发器可以执行两种操作：更新前的记录存储在deleted表中，更新后的记录存储在inserted表中。

--创建update触发器

create trigger trig\_update

on student

after update

as

begin

declare @stuCount int;

select @stuCount=count(\*) from student;

update student\_sum set stuCount =@stuCount;

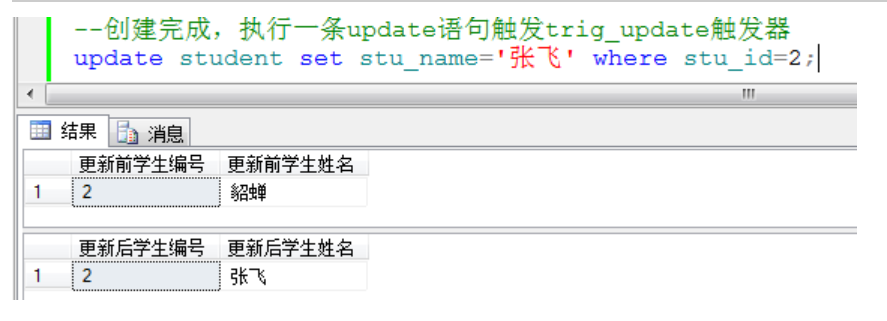
select stu\_id as 更新前学生编号,stu\_name as 更新前学生姓名 from deleted

select stu\_id as 更新后学生编号,stu\_name as 更新后学生姓名 from inserted

End

--创建完成，执行一条update语句触发trig\_update触发器

update student set stu\_name='张飞' where stu\_id=**2**;



**创建替代触发器**

**与前面介绍的三种after触发器不同，SqlServer服务器在执行after触发器的sql代码后，先建立临时的inserted表和deleted表，然后执行代码中对数据库操作，最后才激活触发器中的代码。而对于替代(instead of)触发器,SqlServer服务器在执行触发instead of 触发器的代码时，先建立临时的inserted表和deleted表，然后直接触发instead of触发器，而拒绝执行用户输入的DML操作语句。**

**--创建instead of 触发器**

**create trigger trig\_insteadOf**

**on student**

**instead of insert**

**as**

**begin**

**declare @stuAge int;**

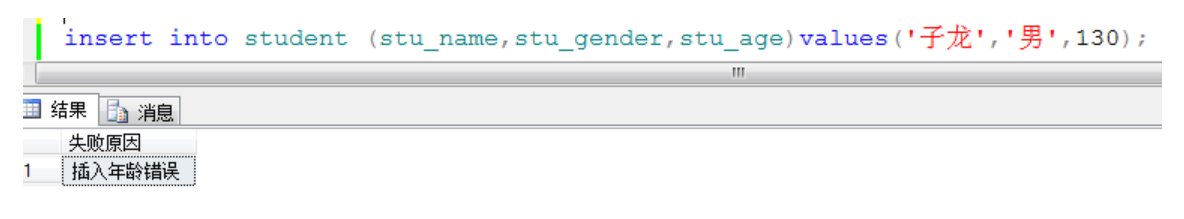
**select @stuAge=(select stu\_age from inserted)**

**if(@stuAge >120)**

**select '插入年龄错误' as '失败原因'**

**End**

创建完成，执行一条insert语句触发触发器trig\_insteadOf



查看触发器

**查看所有触发器**

--查看数据库中所有的触发器

use 数据库名

go

select \* from sysobjects where xtype='TR'

sysobjects 保存着数据库的对象，其中 xtype 为 TR 的记录即为触发器对象。在 name 一列，我们可以看到触发器名称。

**sp\_helptext 查看触发器内容**

use 数据库名

go

exec sp\_helptext '触发器名称'

将会以表的样式显示触发器内容。

 除了触发器外，sp\_helptext 还可以显示 规则、默认值、未加密的存储过程、用户定义函数、视图的文本。

**sp\_helptrigger 用于查看触发器的属性**

sp\_helptrigger 有两个参数：第一个参数为表名；第二个为触发器类型，为 char(6) 类型，可以是 INSERT、UPDATE、DELETE，如果省略则显示指定表中所有类型触发器的属性。

use 数据库名

go

exec sp\_helptrigger tableName

**禁用触发器**

禁用:alter table 表名 disable trigger 触发器名称  
　　启用:alter table 表名 enable trigger 触发器名称

　　如果有多个触发器，则各个触发器名称之间用英文逗号隔开。

如果把“触发器名称”换成“ALL”，则表示禁用或启用该表的全部触发器。

**修改触发器**

--修改触发器语法

ALTER TRIGGER trigger\_name

ON table\_name

[ WITH ENCRYPTION ]

FOR {[DELETE][,][INSERT][,][UPDATE]}

AS

sql\_statement;

**删除触发器**

--语法格式:

DROP TRIGGER { trigger } [ ,...n ]

参数: trigger: 要删除的触发器名称

n:表示可以删除多个触发器的占位符