1.实验七 触发器

1.1 一、实验目的

（1）理解触发器的用途、类型和工作原理

（2）掌握利用T-SQL语句创建和维护触发器的方法

（3）掌握利用企业管理器创建、维护触发器的方法

1.2 二、实验内容

1.2.1 创建after触发器并测试

1.2.1.1 第一题

创建一个在插入时触发的触发器`sc_insert`,当向`sc`表插入数据时，须确保插入的学号已在`Student`表中存在，并且还须确保插入的课程号在`Course`表中存在；若不存在，则给出相应的提示信息，并取消插入操作，提示信息要求指明插入信息是学号不满足条件还是课程号不满足条件。（注：先取消相关外键）

1.2.1.1.1 第一题SQL语句

```
IF EXISTS(SELECT    *
            FROM    sysobjects
           WHERE    name = 'sc_insert' AND type = 'TR')
    BEGIN
        DROP TRIGGER sc_insert
        PRINT '已删除'
    END
ELSE
    BEGIN
        PRINT '可创建'
    END
GO
CREATE TRIGGER sc_insert
    ON SC
    AFTER INSERT
AS
    DECLARE @sno VARCHAR(20), @cno VARCHAR(20)
    SELECT  @sno = Sno, @cno = Cno
    FROM    INSERTED
    IF NOT EXISTS(SELECT    *
                    FROM    Student
                   WHERE    Sno = @sno)
        BEGIN
            ROLLBACK TRAN
            RAISERROR('在Student表中不存在', 16, 10)
        END
    IF NOT EXISTS(SELECT    *
                    FROM    Course
                   WHERE    Cno = @cno)
        BEGIN
```

```
            ROLLBACK TRAN
            RAISERROR ('在Course表中不存在', 16, 10)
        END
GO
```

1.2.1.1.2 执行结果

```
                PRINT '已删除'
        END
ELSE
        BEGIN
                PRINT '可创建'
        END
> 已删除
> OK
> 时间: 0.029s


CREATE TRIGGER sc_insert
        ON SC
        AFTER INSERT
AS
        DECLARE @sno VARCHAR(20), @cno VARCHAR(20)
        SELECT  @sno = Sno, @cno = Cno
        FROM    INSERTED
        IF NOT EXISTS(SELECT    *
                        FROM    Student
                       WHERE    Sno = @sno)
                BEGIN
                        ROLLBACK TRAN
                        RAISERROR('在Student表中不存在', 16, 10)
                END
        IF NOT EXISTS(SELECT    *
                        FROM    Course
                       WHERE    Cno = @cno)
                BEGIN
                        ROLLBACK TRAN
                        RAISERROR ('在Course表中不存在', 16, 10)
                END
> Affected rows: 0
> 时间: 0.023s
```

1.2.1.1.3 验证

1. 学号不满足条件

```
INSERT
  INTO  SC(Sno, Cno, Grade)
VALUES  ('20110050', '001', 50);
```

```
INSERT
  INTO  SC(Sno, Cno, Grade)
VALUES  ('20110050', '001', 50);
> Msg 50000, Level 16, State 10, Server iZwz9b9up93dfbcq3xo029Z, Procedure sc_insert, Line 13
在Student表中不存在
> [42000] [Microsoft][SQL Server Native Client 11.0][SQL Server]在Student表中不存在 (50000)

> 时间: 0.031s
```

2. 课程号不满足条件

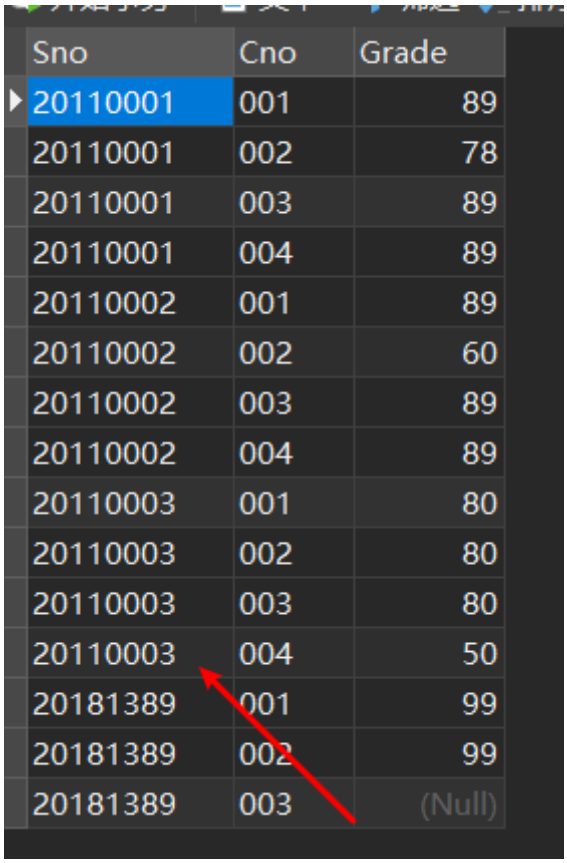
```
INSERT
  INTO  SC(Sno, Cno, Grade)
VALUES  ('20110001', '050', 50);
```

```
INSERT
  INTO  SC(Sno, Cno, Grade)
VALUES  ('20110001', '050', 50);
> Msg 50000, Level 16, State 10, Server iZwz9b9up93dfbcq3xo029Z, Procedure sc_insert, Line 20
在Course表中不存在
> [42000] [Microsoft][SQL Server Native Client 11.0][SQL Server]在Course表中不存在 (50000)

> 时间: 0.029s
```

3. 满足条件

```
INSERT
  INTO  SC(Sno, Cno, Grade)
VALUES  ('20110003', '004', 50)
```



Sno	Cno	Grade
20110001	001	89
20110001	002	78
20110001	003	89
20110001	004	89
20110002	001	89
20110002	002	60
20110002	003	89
20110002	004	89
20110003	001	80
20110003	002	80
20110003	003	80
20110003	004	50
20181389	001	99
20181389	002	99
20181389	003	(Null)

1.2.1.2 第二题

为`Course`表创建一个触发器`Course_del`，当删除了`Course`表中的一条课程信息时，同时将表`sc`表中相应的学生选课记录删除掉。

1.2.1.2.1 第二题SQL代码

```
IF EXISTS(SELECT    *
            FROM    sysobjects
           WHERE    name = 'Course_del' AND type = 'TR')
    DROP TRIGGER Course_del
GO
CREATE TRIGGER Course_del
    ON Course
    AFTER DELETE
```

```
AS
    BEGIN TRANSACTION
        DECLARE @cno VARCHAR(20)
        SELECT  @cno = Cno
          FROM  DELETED
         WHILE  EXISTS(SELECT   *
                         FROM   SC
                        WHERE   Cno = @cno)
            BEGIN
                DELETE
                  FROM  SC
                 WHERE  Cno = @cno
            END
    COMMIT TRANSACTION
GO
```

1.2.1.2.2 执行结果

```
IF EXISTS(SELECT        *
            FROM        sysobjects
           WHERE        name = 'Course_del' AND type = 'TR')
        DROP TRIGGER Course_del
> OK
> 时间: 0.032s


CREATE TRIGGER Course_del
        ON Course
        AFTER DELETE
AS
        BEGIN TRANSACTION
                DECLARE @cno VARCHAR(20)
                SELECT  @cno = Cno
                  FROM  DELETED
                 WHILE  EXISTS(SELECT   *
                                 FROM   SC
                                WHERE   Cno = @cno)
                        BEGIN
                                DELETE
                                  FROM  SC
                                 WHERE  Cno = @cno
                        END
        COMMIT TRANSACTION
> Affected rows: 0
> 时间: 0.023s
```

1.2.1.2.3 验证

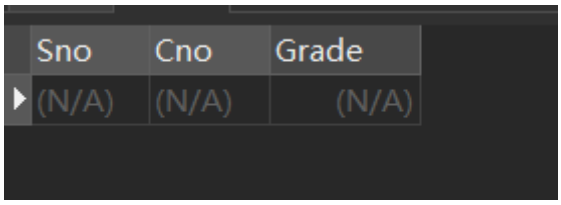
```
DELETE
  FROM  Course
 WHERE  Cno = '009'

DELETE
  FROM  SC
 WHERE  Cno = '009'
```

```
DELETE
  FROM  Course
 WHERE  Cno = '009'

DELETE
  FROM  SC
 WHERE  Cno = '009'
> Affected rows: 0Affected rows: 2Affected rows: 1Affected rows: 0Affected rows: 0
> 时间: 0.063s
```

```sql
SELECT  *
  FROM  SC
 WHERE  Cno = '009'
```



Sno	Cno	Grade
(N/A)	(N/A)	(N/A)

1.2.1.3 第三题

在`Course`表中添加一个平均成绩`avg_Grade`字段（记录每门课程的平均成绩），创建一个触发器`Grade_modify`，当SC表中的某学生的成绩发生变化时，则`Course`表中的平均成绩也能及时相应的发生改变。

1.2.1.3.1 第三题SQL

```sql
IF EXISTS (SELECT   *
             FROM   sysobjects
            WHERE   name = 'Grade_modify' AND type = 'TR')
    BEGIN
        DROP TRIGGER Grade_modify;
        PRINT '已删除！'
    END
PRINT '可以正常建立触发器！'
GO

CREATE TRIGGER Grade_modify
    ON SC
    after INSERT, DELETE, UPDATE
AS
    BEGIN TRANSACTION
        IF UPDATE(Grade)
            BEGIN
                UPDATE Course
                   SET avg_grade = (SELECT  SUM(Grade)/COUNT(*)
                                      FROM  SC
                                     WHERE  Course.Cno = SC.Cno)
                 WHERE Cno IN (SELECT Cno FROM INSERTED)
             END
         ELSE
            BEGIN
                UPDATE  Course
                   SET  avg_grade = (SELECT SUM(Grade)/COUNT(*)
                                       FROM SC
                                      WHERE Course.Cno = SC.Cno)
```

```
                WHERE  Cno IN (SELECT Cno FROM DELETED)
            END
    COMMIT TRANSACTION
 GO
```

1.2.1.3.2 执行结果

```
            PRINT '已删除！'
        END
PRINT '可以正常建立触发器！'
> 已删除！
> 可以正常建立触发器！
> OK
> 时间: 0.051s


CREATE TRIGGER Grade_modify
        ON SC
        after INSERT, DELETE, UPDATE
AS
        BEGIN TRANSACTION
                IF UPDATE(Grade)
                        BEGIN
                                UPDATE Course
                                   SET avg_grade = (SELECT      SUM(Grade)/COUNT(*)
                                                    FROM      SC
                                                   WHERE      Course.Cno = SC.Cno)
                                 WHERE Cno IN (SELECT Cno FROM INSERTED)
                        END
                ELSE
                        BEGIN
                                UPDATE  Course
                                   SET  avg_grade = (SELECT     SUM(Grade)/COUNT(*)
                                                     FROM     SC
                                                    WHERE     Course.Cno = SC.Cno)
                            WHERE      Cno IN (SELECT Cno FROM DELETED)
                        END
        COMMIT TRANSACTION
> Affected rows: 0
> 时间: 0.028s
```

1.2.1.3.3 验证

```
SELECT  *
  FROM  Course;
```

Cno	Cname	Total_perior	Credit	avg_Grade
001	高数	96	6	(Null)
002	C语言程序设计	64	4	(Null)
003	JAVA语言程序设计	48	3	(Null)
004	Visual Basic	48	4	76

```
INSERT
  INTO  SC(Sno, Cno, Grade)
VALUES  ('20110111', '001', 100)

INSERT
  INTO  SC(Sno, Cno, Grade)
VALUES  ('20110111', '002', 100)

```

```sql
INSERT
  INTO  SC(Sno, Cno, Grade)
VALUES  ('20110111', '003', 100)

SELECT  *
  FROM  Course;
```

Cno	Cname	Total_perior	Credit	avg_Grade
001	高数	96	6	91
002	C语言程序设计	64	4	83
003	JAVA语言程序设计	48	3	71
004	Visual Basic	48	4	76

1.2.2 创建instead of 触发器并测试

1.2.2.1 第一题

创建一视图`Student_view`,包含学号、姓名、课程号、课程名、成绩等属性，在`Student_view`上创建一个触发器`Grade_moidfy`，当对`Student_view`中的学生的成绩进行修改时，实际修改的是`sc`中的相应记录。

1.2.2.1.1 第一题SQL语句

```sql
IF EXISTS (SELECT   *
             FROM   sysobjects
            WHERE   name = 'student_view' AND type = 'v')
    BEGIN
        DROP VIEW student_view;
        PRINT '已删除！'
    END
PRINT '可以正常建立视图！'
GO

CREATE VIEW student_view
AS
    SELECT  Student.Sno, Student.Sname, Course.Cno, Course.Cname, SC.Grade
      FROM  SC  JOIN Student ON SC.Sno = Student.Sno
                JOIN Course  ON SC.Cno = Course.Cno
     WHERE  SC.Sno = Student.Sno AND Course.Cno = SC.Cno
GO

IF EXISTS (SELECT   *
             FROM   sysobjects
            WHERE   name = 'grade_modify' AND type = 'TR')
    BEGIN
        DROP TRIGGER grade_modify;
        PRINT '已删除！'
    END
PRINT '可以正常建立触发器！'
GO

CREATE TRIGGER grade_modify
    ON student_view
    INSTEAD OF UPDATE
```

```
AS
    BEGIN TRANSACTION
        DECLARE @grade int, @cno VARCHAR(20), @sno VARCHAR(20)
         SELECT @grade = Grade, @sno = Sno, @cno = Cno
           FROM INSERTED
         UPDATE student_view
            SET Grade = @grade
          WHERE Sno = @sno AND Cno = @cno
     COMMIT TRANSACTION
GO
```

1.2.2.2 第二题

在SC表中插入一个`getcredit`字段（记录某学生，所选课程所获学分的情况），创建一个触发器`ins_credit`，当更改（注：含插入时）`SC`表中的学生成绩时，如果新成绩大于等于60分，则该生可获得这门课的学分，且该学分须与`Course`表中的值一致；如果新成绩小于60分，则该生未能获得学分，修改值为0。

1.2.2.2.1 第二题SQL语句

```
IF EXISTS (SELECT   *
             FROM   sysobjects
            WHERE   name = 'ins_credit' AND type = 'TR')
    BEGIN
        DROP TRIGGER ins_credit;
        PRINT '已删除！'
    END
PRINT '可以正常建立触发器！'
GO

CREATE TRIGGER ins_credit
    ON SC
    INSTEAD OF UPDATE, DELETE, INSERT
AS
    BEGIN TRANSACTION
        DECLARE @cno VARCHAR(20), @sno VARCHAR(20), @grade TINYINT,
                @cno_old VARCHAR(20), @sno_old VARCHAR(20)
        IF UPDATE(Grade)
            BEGIN
                IF NOT EXISTS (SELECT * FROM DELETED)
                    BEGIN
                        SELECT  @cno = Cno, @sno = Sno, @grade = Grade
                          FROM  INSERTED
                        INSERT
                          INTO  SC(Sno, Cno, Grade)
                        SELECT  Sno, Cno, Grade FROM INSERTED
                        UPDATE  Student
                           SET  getcredit = (SELECT COUNT(*)
                                               FROM SC
                                              WHERE Grade >= 60 AND Sno =
@sno)
                         WHERE  Sno = @sno
                    END
                ELSE
                    BEGIN
```

```
                        SELECT  @cno = Cno, @sno = Sno, @grade = Grade
                          FROM  INSERTED
                        UPDATE  Student
                           SET  getcredit = (SELECT COUNT(*)
                                               FROM SC
                                              WHERE Grade >= 60 AND Sno =
@sno)
                         WHERE  Sno = @sno
                    END
            END
        ELSE
            BEGIN
                SELECT  @cno = Cno, @sno = Sno, @grade = Grade
                  FROM  DELETED
                DELETE
                  FROM  SC
                 WHERE  Sno = @sno AND Cno = @cno
                UPDATE  Student
                   SET  getcredit = (SELECT COUNT(*)
                                       FROM SC
                                      WHERE Grade >= 60 AND Sno = @sno)
                 WHERE  Sno = @sno
            END
    COMMIT TRANSACTION
GO
```

1.2.2.2.2 运行结果

```
                                SELECT      Sno, Cno, Grade FROM INSERTED
                                    UPDATE  Student
                                       SET  getcredit = (SELECT     COUNT(*)
                                                           FROM     SC
                                                           WHERE    Grade >= 60 AND Sno = @sno)
                                     WHERE      Sno = @sno
                                END
                        ELSE
                                BEGIN
                                        SELECT  @cno = Cno, @sno = Sno, @grade = Grade
                                          FROM  INSERTED
                                    UPDATE      Student
                                           SET  getcredit = (SELECT     COUNT(*)
                                                               FROM     SC
                                                               WHERE    Grade >= 60 AND Sno = @sno)
                                     WHERE      Sno = @sno
                                END
                END
            ELSE
                BEGIN
                        SELECT  @cno = Cno, @sno = Sno, @grade = Grade
                          FROM  DELETED
                    DELETE
                        FROM  SC
                     WHERE      Sno = @sno AND Cno = @cno
                    UPDATE      Student
                           SET  getcredit = (SELECT     COUNT(*)
                                               FROM     SC
                                              WHERE     Grade >= 60 AND Sno = @sno)
                         WHERE  Sno = @sno
                END
        COMMIT TRANSACTION
> Affected rows: 0
> 时间: 0.473s
```

1.2.3 使用T-SQL语句管理和维护

1.2.3.1 第一题

用系统存储过程 `sp_helptrigger` 查看触发器 `Grade_modify` 的相关信息

1.2.3.1.1 SQL语句

```
sp_helptrigger SC
GO
```

1.2.3.1.2 运行结果

trigger_name	trigger_owner	isupdate	isdelete	isinsert	isafter	isinsteadof	trigger_schema
ins_credit	dbo	1	1	1	0	1	dbo
sc_insert	dbo	0	0	1	1	0	dbo

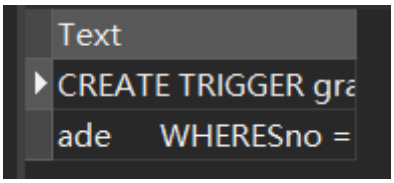
1.2.3.2 第二题

使用系统存储过程`sp_helptext`查看触发器`Grade_modify`中的定义内容。

1.2.3.2.1 SQL语句

```
sp_helptext grade_modify
GO
```

1.2.3.2.2 运行结果



1.2.3.3 第三题

使用`select`语句查看触发器`Grade_modify`的定义内容。

1.2.3.3.1 SQL语句

```
SELECT  text
  FROM  syscomments
 WHERE  text LIKE '%grade_modify%';
GO
```

1.2.3.3.2 运行结果

1.2.3.4 第四题

用系统存储过程`sp_depends`查看触发器`Grade_modify`的相关性。

1.2.3.4.1 SQL语句

```
sp_depends grade_modify
GO
```

1.2.3.4.2 运行结果

name	type		selected	column
dbo.student_view	view	no	yes	Sno
dbo.student_view	view	no	yes	Cno
dbo.student_view	view	yes	no	Grade

1.2.3.5 第五题

将`sc_insert`触发器改为`instead of`触发器，实现的功能不变。

1.2.3.5.1 SQL语句

```
IF EXISTS (SELECT   *
             FROM   sysobjects
            WHERE   name = 'ins_credit' AND type = 'TR')
    DROP TRIGGER ins_credit
GO

ALTER TRIGGER sc_insert
    ON SC
    INSTEAD OF INSERT
AS
    DECLARE @sno VARCHAR(20), @cno VARCHAR(20), @grade tinyint
     SELECT @sno = Sno, @cno = Cno, @grade = Grade
       FROM INSERTED
    IF NOT EXISTS(SELECT    *
                    FROM    Student
                   WHERE    Sno = @sno)
        BEGIN
            ROLLBACK TRAN
            RAISERROR('学号在Student表中不存在!', 16, 10)
        END
    IF NOT EXISTS(SELECT    *
                    FROM    Course
                   WHERE    Cno = @cno)
        BEGIN
            ROLLBACK TRAN
            RAISERROR('课程号在Course中不存在!', 16, 10)
        END
    INSERT
      INTO  SC(Sno, Cno, Grade)
    VALUES  (@sno, @cno, @grade)
GO

```

1.2.3.5.2 运行结果

```
> 时间: 0.049s


ALTER TRIGGER sc_insert
        ON SC
        INSTEAD OF INSERT
AS
        DECLARE @sno VARCHAR(20), @cno VARCHAR(20), @grade tinyint
         SELECT @sno = Sno, @cno = Cno, @grade = Grade
           FROM INSERTED
        IF NOT EXISTS(SELECT    *
                      FROM    Student
                                 WHERE        Sno = @sno)
                BEGIN
                        ROLLBACK TRAN
                        RAISERROR('学号在Student表中不存在!', 16, 10)
                END
        IF NOT EXISTS(SELECT    *
                      FROM    Course
                                 WHERE        Cno = @cno)
                BEGIN
                        ROLLBACK TRAN
                        RAISERROR('课程号在Course中不存在!', 16, 10)
                END
        INSERT
          INTO  SC(Sno, Cno, Grade)
        VALUES  (@sno, @cno, @grade)
> Affected rows: 0
> 时间: 0.102s
```

1.2.3.6 第六题

将触发器`sc_insert`删除。

1.2.3.6.1 SQL语句

```
DROP TRIGGER SC_insert

```

1.2.3.6.2 运行结果

```
DROP TRIGGER SC_insert
> OK
> 时间: 0.11s
```

1.2.4 使用SQL Server Management Studio管理存储过程

1.2.4.1 第一题

在`SQL Server Management Studio`中重新创建刚删除的触发器`sc_insert`

1.2.4.1.1 解答

点击新建触发器

然后再输入即可

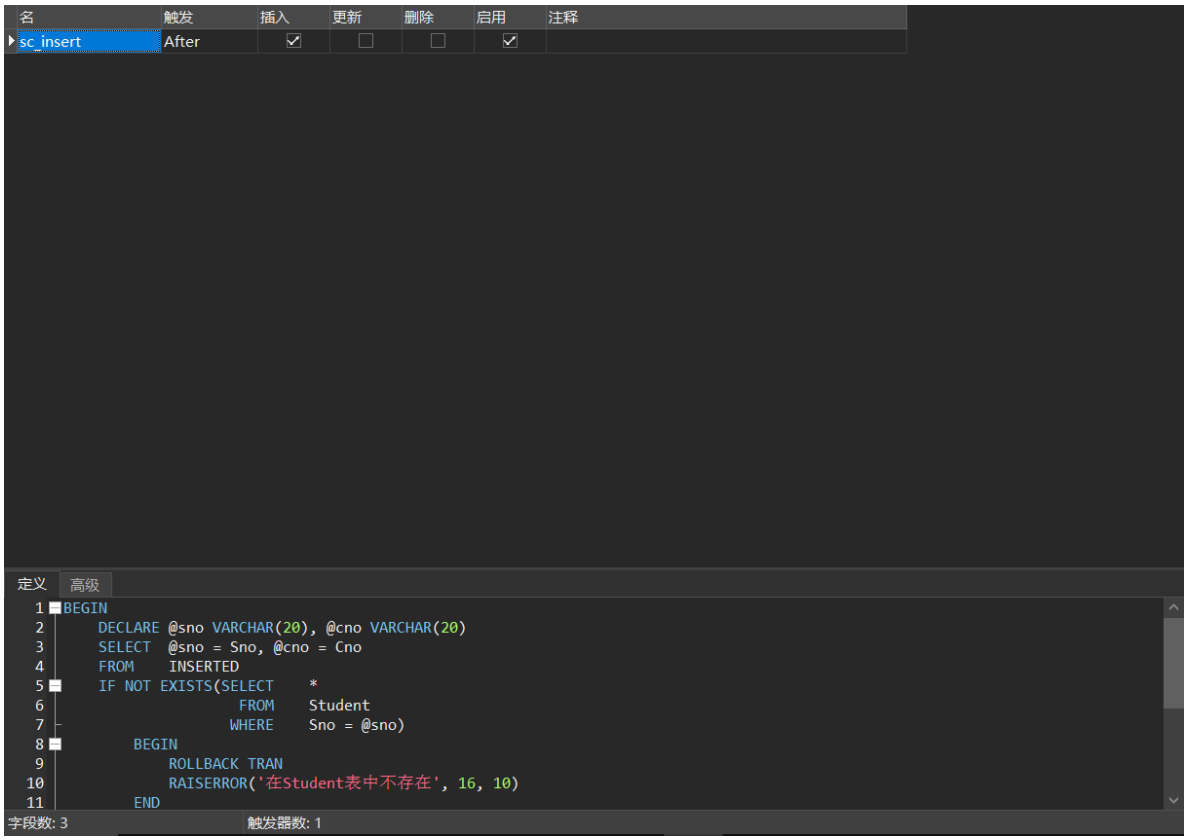
```
BEGIN
    DECLARE @sno VARCHAR(20), @cno VARCHAR(20)
    SELECT  @sno = Sno, @cno = Cno
    FROM    INSERTED
    IF NOT EXISTS(SELECT    *
                  FROM    Student
                 WHERE    Sno = @sno)
        BEGIN
            ROLLBACK TRAN
            RAISERROR('在Student表中不存在', 16, 10)
        END
```

1.2.4.2 第二题

查看触发器`sc_insert`的内容。

1.2.4.2.1 解答

双击触发器即可



1.2.4.3 第三题

删除触发器`sc_insert`

1.2.4.3.1 解答

点击删除触发器即可


复制
粘贴
添加触发器
删除触发器