

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

```
1  IF EXISTS(SELECT      *
2                      FROM      sysobjects
3                      WHERE      name = 'sc_insert' AND type = 'TR')
4      BEGIN
5          DROP TRIGGER sc_insert
6          PRINT '已删除'
7      END
8  ELSE
9      BEGIN
10         PRINT '可创建'
11     END
12  GO
13  CREATE TRIGGER sc_insert
14      ON SC
15      AFTER INSERT
16  AS
17      DECLARE @sno VARCHAR(20), @cno VARCHAR(20)
18      SELECT  @sno = Sno, @cno = Cno
19      FROM    INSERTED
20      IF NOT EXISTS(SELECT      *
21                      FROM      Student
22                      WHERE      Sno = @sno)
23          BEGIN
24              ROLLBACK TRAN
25              RAISERROR('在Student表中不存在', 16, 10)
26          END
27      IF NOT EXISTS(SELECT      *
28                      FROM      Course
29                      WHERE      Cno = @cno)
30          BEGIN
```

```

31      ROLLBACK TRAN
32      RAISERROR ('在Course表中不存在', 16, 10)
33  END
34  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
    END
> Affected rows: 0
> 时间: 0.023s

```

1.2.1.1.3 验证

1. 学号不满足条件

```

1  INSERT
2  INTO  SC(Sno, Cno, Grade)
3  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

```

1 INSERT
2 INTO SC(Sno, Cno, Grade)
3 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

```

1 INSERT
2 INTO SC(Sno, Cno, Grade)
3 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

```

1 IF EXISTS(SELECT *
2           FROM sysobjects
3           WHERE name = 'Course_del' AND type = 'TR')
4   DROP TRIGGER Course_del
5 GO
6 CREATE TRIGGER Course_del
7   ON Course
8   AFTER DELETE

```

```

9  AS
10  BEGIN TRANSACTION
11  DECLARE @cno VARCHAR(20)
12  SELECT @cno = Cno
13  FROM DELETED
14  WHILE EXISTS(SELECT *
15                FROM SC
16                WHERE Cno = @cno)
17  BEGIN
18  DELETE
19  FROM SC
20  WHERE Cno = @cno
21  END
22  COMMIT TRANSACTION
23  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

```

1  DELETE
2  FROM Course
3  WHERE Cno = '009'
4
5  DELETE
6  FROM SC
7  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
```

```
1 SELECT *
2 FROM SC
3 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

[illegible]

```

30 WHERE Cno IN (SELECT Cno FROM DELETED)
31 END
32 COMMIT TRANSACTION
33 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

```

1 SELECT *
2 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

```

1 INSERT
2 INTO SC(Sno, Cno, Grade)
3 VALUES ('20110111', '001', 100)
4
5 INSERT
6 INTO SC(Sno, Cno, Grade)
7 VALUES ('20110111', '002', 100)
8

```

```

9  INSERT
10     INTO SC(Sno, Cno, Grade)
11     VALUES ('20110111', '003', 100)
12
13     SELECT *
14     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_modify`, 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

```

1  IF EXISTS (SELECT *
2             FROM sysobjects
3             WHERE name = 'student_view' AND type = 'v')
4      BEGIN
5          DROP VIEW student_view;
6          PRINT '已删除! '
7      END
8  PRINT '可以正常建立视图! '
9  GO
10
11  CREATE VIEW student_view
12  AS
13      SELECT Student.Sno, Student.Sname, Course.Cno, Course.Cname, SC.Grade
14      FROM SC JOIN Student ON SC.Sno = Student.Sno
15             JOIN Course ON SC.Cno = Course.Cno
16      WHERE SC.Sno = Student.Sno AND Course.Cno = SC.Cno
17  GO
18
19  IF EXISTS (SELECT *
20             FROM sysobjects
21             WHERE name = 'grade_modify' AND type = 'TR')
22      BEGIN
23          DROP TRIGGER grade_modify;
24          PRINT '已删除! '
25      END
26  PRINT '可以正常建立触发器! '
27  GO
28
29  CREATE TRIGGER grade_modify
30      ON student_view
31      INSTEAD OF UPDATE

```

```

32 AS
33 BEGIN TRANSACTION
34 DECLARE @grade int, @cno VARCHAR(20), @sno VARCHAR(20)
35 SELECT @grade = Grade, @sno = Sno, @cno = Cno
36 FROM INSERTED
37 UPDATE student_view
38 SET Grade = @grade
39 WHERE Sno = @sno AND Cno = @cno
40 COMMIT TRANSACTION
41 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

```

1 IF EXISTS (SELECT *
2             FROM sysobjects
3             WHERE name = 'ins_credit' AND type = 'TR')
4 BEGIN
5     DROP TRIGGER ins_credit;
6     PRINT '已删除！'
7 END
8 PRINT '可以正常建立触发器！'
9 GO
10
11 CREATE TRIGGER ins_credit
12 ON SC
13 INSTEAD OF UPDATE, DELETE, INSERT
14 AS
15 BEGIN TRANSACTION
16 DECLARE @cno VARCHAR(20), @sno VARCHAR(20), @grade TINYINT,
17         @cno_old VARCHAR(20), @sno_old VARCHAR(20)
18 IF UPDATE(Grade)
19 BEGIN
20     IF NOT EXISTS (SELECT * FROM DELETED)
21     BEGIN
22         SELECT @cno = Cno, @sno = Sno, @grade = Grade
23         FROM INSERTED
24         INSERT
25             INTO SC(Sno, Cno, Grade)
26         SELECT Sno, Cno, Grade FROM INSERTED
27         UPDATE Student
28             SET getcredit = (SELECT COUNT(*)
29                             FROM SC
30                             WHERE Grade >= 60 AND Sno =
31                                 @sno)
32         WHERE Sno = @sno
33     END
34 ELSE
35 BEGIN

```



```

35         SELECT @cno = Cno, @sno = Sno, @grade = Grade
36         FROM INSERTED
37         UPDATE Student
38             SET getcredit = (SELECT COUNT(*)
39                             FROM SC
40                             WHERE Grade >= 60 AND Sno =
@sno)
41             WHERE Sno = @sno
42         END
43     END
44 ELSE
45     BEGIN
46         SELECT @cno = Cno, @sno = Sno, @grade = Grade
47         FROM DELETED
48         DELETE
49             FROM SC
50             WHERE Sno = @sno AND Cno = @cno
51         UPDATE Student
52             SET getcredit = (SELECT COUNT(*)
53                             FROM SC
54                             WHERE Grade >= 60 AND Sno = @sno)
55             WHERE Sno = @sno
56     END
57 COMMIT TRANSACTION
58 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

```
1 | sp_helptrigger SC
2 | 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

```
1 | sp_helptext grade_modify
2 | GO
```

1.2.3.2.2 运行结果

```
Text
▶ CREATE TRIGGER gra
ade WHERE Sno =
```

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

```
1 | SELECT text
2 | FROM syscomments
3 | WHERE text LIKE '%grade_modify%';
4 | GO
```

1.2.3.3.2 运行结果

```
text
▶ CREATE TRIGGER gra
```

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

```
1 | sp_depends grade_modify
2 | 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

```

1  IF EXISTS (SELECT *
2             FROM sysobjects
3             WHERE name = 'ins_credit' AND type = 'TR')
4      DROP TRIGGER ins_credit
5  GO
6
7  ALTER TRIGGER sc_insert
8      ON SC
9      INSTEAD OF INSERT
10 AS
11     DECLARE @sno VARCHAR(20), @cno VARCHAR(20), @grade tinyint
12     SELECT @sno = Sno, @cno = Cno, @grade = Grade
13     FROM INSERTED
14     IF NOT EXISTS(SELECT *
15                  FROM Student
16                  WHERE Sno = @sno)
17     BEGIN
18         ROLLBACK TRAN
19         RAISERROR('学号在Student表中不存在!', 16, 10)
20     END
21     IF NOT EXISTS(SELECT *
22                  FROM Course
23                  WHERE Cno = @cno)
24     BEGIN
25         ROLLBACK TRAN
26         RAISERROR('课程号在Course中不存在!', 16, 10)
27     END
28     INSERT
29     INTO SC(Sno, Cno, Grade)
30     VALUES (@sno, @cno, @grade)
31 GO
32

```

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

```

1 DROP TRIGGER SC_insert
2

```

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

```
1 BEGIN
2   DECLARE @sno VARCHAR(20), @cno VARCHAR(20)
3   SELECT @sno = Sno, @cno = Cno
4   FROM   INSERTED
5   IF NOT EXISTS(SELECT *
6                 FROM   Student
7                 WHERE  Sno = @sno)
8   BEGIN
9     ROLLBACK TRAN
10    RAISERROR('在Student表中不存在', 16, 10)
11  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

名	触发	插入	更新	删除	启用	注释
sc_insert	After	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

定义 高级

```
1 BEGIN
2   DECLARE @sno VARCHAR(20), @cno VARCHAR(20)
3   SELECT @sno = Sno, @cno = Cno
4   FROM   INSERTED
5   IF NOT EXISTS(SELECT *
6                 FROM   Student
7                 WHERE  Sno = @sno)
8   BEGIN
9     ROLLBACK TRAN
10    RAISERROR('在Student表中不存在', 16, 10)
11  END
```

字段数: 3 触发器数: 1

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
<hr/>	
	添加触发器
	删除触发器