

# 上机实验报告

课程名： 数据库安全

指导教师： 熊光彩

姓名	周诚远	班级	SC011801	实验得分	
		学号	2018302191		
实验时间	2021. 3. 29	实验地点	启翔楼 323		
实验主题	触发器				
实验目的	<p>(1) 掌握触发器的概念。</p> <p>(2) 理解触发器的优点。</p> <p>(3) 理解触发器与存储过程的区别。</p> <p>(4) 掌握触发器的类型。①DML 触发器与 DDL 触发器；②AFTER/FOR 触发器与 INSTEADOF 触发器。</p> <p>(5) 深入掌握两个专用表：Inserted 表、Deleted 表</p>				
实验内容	<p>(1)在 SchoolManagement 数据库中,为“学生表 S”创建触发器 tri_upd_S,其作用是当更改了学生的学号,其选课记录仍然与这个学生相关(即同时更改选课表相应的学号)。{请分别用 AFTER/FOR 触发器及 INSTEAD OF 触发器予以实现}</p> <p>(2) 在 SchoolManagement 数据库中,为“学生表 S”创建一个名为 tri_no_updSname_S 的 UPDATE 触发器,其作用是当修改“S”表中的“Sname”字段时,提示不能修改,并取消修改操作。</p> <p>(3) 在 SchoolManagement 数据库中,为“学生表 S”创建一个名为 tri_no_upd_S 的 UPDATE 触发器,其作用是当修改“S”表中的任意字段时,提示不能修改,并取消修改操作。</p> <p>(4)在 SchoolManagement 数据库中,为“学生表 S”创建触发器 tri_del_S,其作用是当该学生已经转校,在“S”表删除他的记录的同时,也删除他在“SC”表中的所有选课记录。</p> <p>(5) 在 SchoolManagement 数据库中,为“教学计划 C_Plan”创建触发器一名 tri_ins_jxjh 触发器,其作用是当在“教学计划 C_Plan”表中插入一条新记录时,同时在“教师任课 C_Teacher”表中自动添加相关的任课记录(假设:教师编号=‘T2020’+C_Plan.课程号)。{请分别用 AFTER/FOR 触发器及 INSTEAD OF 触发器予以实现}</p>				

	<p>(6) 在 SchoolManagement 数据库中, 为“教学计划 C_Plan”创建一名为 del_tri_CourPlan 的触发器, 其作用是: 当删除“C_Plan”表中的某一记录时, 同时在“任课教师表 C_Teacher”表中删除与课程计划表中的“C_Plan. 课程号”相关的任课教师记录。{请分别用 AFTER/FOR 触发器及 INSTEAD OF 触发器予以实现}</p> <p>(7) 在 SchoolManagement 数据库中, 为“教学计划 C_Plan”表创建一名为 tri_no_updCno 的触发器, 其作用是当修改“C_Plan”表中“课程号”字段时, 提示不能修改, 并取消修改操作。{请分别用 AFTER/FOR 触发器及 INSTEAD OF 触发器予以实现}</p> <p>(8) 修改上例中建立在“教学计划 C_Plan”表上的触发器 tri_no_updCno, 使其既不能修改“C_Plan”表中“课程号”字段的值, 也不能修改该表中“学期”字段的值。</p> <p>(9) 对“教学计划 C_Plan”表上的触发器 tri_no_updCno 进行禁止, 之后再启用该触发器, 最后再删除该触发器。</p> <p>(10) 在 SchoolManagement 数据库范围内创建一个触发器, 禁止删除或修改该数据库中的任何表。</p>
<p>实验 结果 / 实验 结论</p>	<p>(1) 建表并插入数据</p> <pre> CREATE TABLE S(Sno NVARCHAR(10) NOT NULL PRIMARY KEY,                 Sname NVARCHAR(10), Ssex nvarchar(1),                 Sage SMALLINT, Sdept NVARCHAR(50), Schoolship money null); CREATE TABLE C(Cno nvarchar(10) NOT NULL PRIMARY KEY,                 Cname nvarchar(15), Pno nvarchar(15), Ccredit decimal(4,1) ); CREATE TABLE SC(Sno nvarchar(10) not null foreign key references S(Sno),                   Cno nvarchar(10) not null foreign key references C(Cno) Primary Key(Sno,Cno),                   Grade decimal(5,1)); CREATE TABLE C_Plan(Cno nvarchar(10) NULL ,                      Mclass nvarchar(4) null,Mno nvarchar(10) null,Syear varchar(4) ,                      Term nvarchar(3) null , StudentNum int null); CREATE TABLE C_Teacher(Tno nvarchar(15) NULL,                         Cno nvarchar(10) null,                         Mclass varchar(4) null ,Mno nvarchar(10) null,Syear varchar(4) null,StudentNum int null);  declare @i int declare @c int set @i=1 set @c=69  while @i&lt;@c begin insert into S select top (1) Sno,sname,ssex,Sage,Sdept,FLOOR(RAND()*1000) from SM1.dbo.S where Sno not in (select set @i=@i+1 END  insert into C select * from SM1.DBO.C  delete from SC  declare @i int set @i=1 </pre> <p>(2) 在 SchoolManagement 数据库中, 为“学生表 S”创建触发器 tri_upd_S, 其作用是当更改了学生的学号, 其选课记录仍然与这个学生相关 (即同时更改选课表相应的学号)。{请分别用 AFTER/FOR 触发器及 INSTEAD OF 触发器予以实现}</p>

```

Create TRIGGER [dbo].[tri_upd_S]
ON [dbo].[S]
AFTER update
AS
BEGIN

    SET NOCOUNT ON;

    Declare @sno varchar(10);
    Declare @Osno varchar(10);
    Select @sno=sno From inserted
    Select @Osno=sno From deleted
    update SC SET Sno=@sno WHERE Sno=@Osno
END

Create TRIGGER [dbo].[tri_upd_S1]
ON [dbo].[S]
instead of update
AS
BEGIN

    SET NOCOUNT ON;

    Declare @sno varchar(10);
    Declare @Osno varchar(10);
    Select @sno=sno From inserted
    Select @Osno=sno From deleted
    update SC SET Sno=@sno WHERE Sno=@Osno
    update S SET Sno=@sno WHERE Sno=@Osno
END

```

- (3) 在 SchoolManagement 数据库中，为“学生表 S”创建一个名为 tri\_no\_updSname\_S 的 UPDATE 触发器，其作用是当修改“S”表中的“Sname”字段时，提示不能修改，并取消修改操作

```

Create TRIGGER [dbo].[tri_no_upd_S]
ON [dbo].[S]
AFTER update
AS
BEGIN

    SET NOCOUNT ON;

    Declare @sname NVARCHAR(10);
    Declare @Osname NVARCHAR(10);
    Select @sname=sno From inserted
    Select @Osname=sno From deleted
    if @sname=@Osname
    begin
        insert into S select *from inserted
    end
    else
    begin
        print('不能修改')
        rollback transaction
    end
END

```

- (4) 在 SchoolManagement 数据库中，为“学生表 S”创建一个名为 tri\_no\_upd\_S 的 UPDATE 触发器，其作用是当修改“S”表中的任意字段时，提示不能修改，并取消修改操作。

```

Create TRIGGER [dbo].[tri_no_upd_S]
ON [dbo].[S]
AFTER update
AS
BEGIN

    print('不能修改')
    rollback transactio

END

```

- (5) 在 SchoolManagement 数据库中，为“学生表 S”创建触发器 tri\_del\_S，其作用是当该学生已经转校，在“S”表删除他的记录的同时，也删除他在“SC”表中的所有选课记录。

```

Create TRIGGER [dbo].[tri_no_upd_S]
ON [dbo].[S]
instead of delete
AS
BEGIN
    SET NOCOUNT ON;
    Declare @sno varchar(10);
    select @sno=sno from deleted
    delete from SC where Sno=@sno
    delete from S where Sno=@sno
END
|
delete from S WHERE Sno='2017300870'

```

消息

1 行受影响)

- (6) 在 SchoolManagement 数据库中，为“教学计划 C\_Plan”创建触发器一名 tri\_ins\_jxjh 触发器，其作用是当在“教学计划 C\_Plan”表中插入一条新记录时，同时在“教师任课 C\_Teacher”表中自动添加相关的任课记录（假设：教师编号= ‘T2020’ +C\_Plan. 课程号）。{请分别用 AFTER/FOR 触发器及 INSTEAD OF 触发器予以实现}

```

3 Create TRIGGER [dbo].[tri_ins_jxjh]
ON [dbo].[C_Plan]
instead of INSERT
AS
3 BEGIN
SET NOCOUNT ON;
Declare @Cno varchar(10);
Declare @Tno varchar(10);
select @Cno=cno from inserted
SET @Tno='T2020'+@Cno
INSERT INTO C_Plan select *from inserted
INSERT INTO C_teacher select @Tno,Mclass,Mno,Syear,Syear,StudentNum from inserted
- END
3 Create TRIGGER [dbo].[tri_ins_ixjh]
ON [dbo].[C_Plan]
after INSERT
AS
3 BEGIN
SET NOCOUNT ON;
Declare @Cno varchar(10);
Declare @Tno varchar(10);
select @Cno=cno from inserted
SET @Tno='T2020'+@Cno
--INSERT INTO C_Plan select *from inserted
INSERT INTO C_teacher select @Tno,Mclass,Mno,Syear,Syear,StudentNum from inserted
- END
insert into C_plan values(1,1,1,1,1,1)
-select * from C_Teacher

```

结果 消息

Tno	Cno	Mclass	Mno	Syear	StudentNum
T20201	1	1	1	1	1

(7) 在 SchoolManagement 数据库中，为“教学计划 C\_Plan”创建一名为 del\_tri\_CourPlan 的触发器，其作用是：当删除“C\_Plan”表中的某一记录时，同时在“任课教师表 C\_Teacher”表中删除与课程计划表中的“C\_Plan. 课程号”相关的任课教师记录。{请分别用 AFTER/FOR 触发器及 INSTEAD OF 触发器予以实现}

```

3 Create TRIGGER [dbo].[del_tri_CourPlan]
ON [dbo].[C_Plan]
after delete
AS
3 BEGIN
SET NOCOUNT ON;
Declare @Cno varchar(10);
Declare @Tno varchar(10);
select @Cno=cno from deleted
SET @Tno='T2020'+@Cno
delete from C_teacher where Tno=('T2020'+@Cno)
- END
3 Create TRIGGER [dbo].[del_tri_CourPlan1]
ON [dbo].[C_Plan]
instead of delete
AS
3 BEGIN
SET NOCOUNT ON;
Declare @Cno varchar(10);
Declare @Tno varchar(10);
select @Cno=cno from deleted
SET @Tno='T2020'+@Cno
delete from C_teacher where Tno=('T2020'+@Cno)
delete from C_Plan where Cno=@Cno
- END
delete from C_plan where cno='1'
delete from C_teacher
insert into C_plan values(1,1,1,1,1,1)
delete from C_Plan where cno=1
select*from C_Plan
select*from C_Teacher

```

结果 消息

Tno	Cno	Mclass	Mno	Syear	StudentNum
-----	-----	--------	-----	-------	------------

Cno	Mclass	Mno	Syear	Term	StudentNum
-----	--------	-----	-------	------	------------

- (8) 在 SchoolManagement 数据库中, 为“教学计划 C\_Plan”表创建一名为 tri\_no\_updCno 的触发器, 其作用是当修改“C\_Plan”表中“课程号”字段时, 提示不能修改, 并取消修改操作。[请分别用 AFTER/FOR 触发器及 INSTEAD OF 触发器予以实现]

```
3 Create TRIGGER [dbo].[tri_no_updCno]
  ON [dbo].[C_Plan]
  instead of update
AS
3 BEGIN
  SET NOCOUNT ON;
  Declare @Cno varchar(10); Declare @Ocno varchar(10);
  select @Ocno=cno from deleted; select @Cno=cno from inserted
  if @Cno=@Ocno
  begin
    insert into c_Plan select *from inserted
  end
  else
  begin
    print('不可修改')
  end
END
3 Create TRIGGER [dbo].[tri_no_updCnoII]
  ON [dbo].[C_Plan]
  instead of update
AS
3 BEGIN
  SET NOCOUNT ON;
  Declare @Cno varchar(10); Declare @Ocno varchar(10);
  select @Ocno=cno from deleted; select @Cno=cno from inserted
  if @Cno=@Ocno
  begin
    insert into c_Plan select *from inserted
  end
  else
  begin
    print('不可修改'); rollback transaction
  end
END
```

- (9) 修改上例中建立在“教学计划 C\_Plan”表上的触发器 tri\_no\_updCno, 使其既不能修改“C\_Plan”表中“课程号”字段的值, 也不能修改该表中“学期”字段的值。

```
3 Create TRIGGER [dbo].[tri_no_updCno]
  ON [dbo].[C_Plan]
  instead of update
AS
3 BEGIN
  SET NOCOUNT ON;
  Declare @Cno varchar(10); Declare @Ocno varchar(10);
  Declare @term varchar(10); Declare @Oterm varchar(10);
  select @Ocno=cno from deleted; select @Cno=cno from inserted
  select @Oterm=term from deleted; select @term=term from inserted
  if @Cno=@Ocno and @Oterm=@term
  begin
    insert into c_Plan select *from inserted
  end
  else
  begin
    print('不可修改')
  end
END
```

	<p>(10) 对“教学计划 C_Plan”表上的触发器 tri_no_updCno 进行禁止，之后再启用该触发器，最后再删除该触发器。</p> <pre> ALTER TABLE C_Plan DISABLE TRIGGER tri_no_updCno ALTER TABLE C_Plan ENABLE TRIGGER tri_no_updCno DROP TRIGGER tri_no_updCno </pre> <p>(11) 在 SchoolManagement 数据库范围内创建一个触发器，禁止删除或修改该数据库中的任何表。在 SchoolManagement 数据库范围内创建一个触发器，禁止删除或修改该数据库中的任何表。</p> <pre> CREATE TRIGGER DDL_TableTrigger ON DATABASE FOR DROP_TABLE, ALTER_TABLE AS BEGIN     PRINT ('禁止操作')     ROLLBACK ; end drop table C_Teacher; </pre> <div data-bbox="288 1108 1275 1193" style="background-color: #f0f0f0; border: 1px solid #ccc; padding: 5px;"> <p>消息</p> <p>禁止操作</p> <p>消息 3726, 级别 16, 状态 1, 过程 DDL_TableTrigger, 第 255 行 无法删除对象 'S', 因为该对象正由一个 FOREIGN KEY 约束引用。</p> <p>消息 3609, 级别 16, 状态 2, 第 256 行 事务在触发器中结束。批处理已中止。</p> </div>
实验心得	<p>通过本次实验，我了解了什么是触发器和触发器的使用方式，通过触发器我们可以防止一些误操作，也可以在数据更改时对其他的表的数据进行修改，既方便了用户又保证了数据完整性。</p>