

19335262 张航悦 实验 13

练习1

编写一个嵌套事务。外层修改students表某记录，内层在teachers表插入一条记录。演示内层插入操作失败后，外层修改操作回滚。

- 代码

```
select * from STUDENTS where sid='1000'
begin tran tran_upd_students
    update STUDENTS
    set grade=2021
    where sid='1000'
    begin tran tran_upd_teachers
        insert into teachers values('1234567890','zs','my@zsu.edu.cn',3000)
    if @@ERROR!=0
    begin
        rollback tran tran_upd_students
        print '更新教师表失败'
        print '更新学生表失败'
        return
    end

    commit tran tran_upd_teachers
    commit tran tran_upd_students
go
select * from STUDENTS where sid='1000'
```

100 %

结果 消息

(1 行受影响)

(1 行受影响)

消息 2627, 级别 14, 状态 1, 第 7 行
违反了 PRIMARY KEY 约束 'PK_TEACHERS'。不能在对象 'dbo.TEACHERS' 中插入重复键。
语句已终止。
更新教师表失败
更新学生表失败

(1 行受影响)

- 演示内层插入失败

外层学生表修改操作回滚，并未真实修改。

结果		消息			
	sid	sname	email	grade	
1	1000	Tom	NULL	NULL	
	sid	sname	email	grade	
1	1000	Tom	NULL	NULL	

练习2

编写一个带有保存点的事务。更新teachers表中数据后，设置事务保存点，然后在表courses中插入数据，如果courses插入数据失败，则回滚到事务保存点。演示courses插入失败，但teachers表更新成功的操作。

- 代码

```
select * from TEACHERS
where tid='1234567890'

begin tran tran_upd_teachers
    update TEACHERS
    set salary=999
    where tid='1234567890'
    save tran tran_upd_courses_done
    insert into COURSES
    values ('10001','english',10)
    if @@ERROR!=0 or @@ROWCOUNT>1
    begin
        rollback tran tran_upd_courses_done
        print '更新课程表信息失败'
        return
    end
commit tran tran_upd_teachers

go
select * from TEACHERS
where tid='1234567890'
```

100 %

结果 消息

(1 行受影响)

(1 行受影响)

消息 2627, 级别 14, 状态 1, 第 9 行
违反了 PRIMARY KEY 约束 'PK_COURSES'。不能在对象 'dbo.COURSES' 中插入重复键。
语句已终止。
更新课程表信息失败

(1 行受影响)

- 演示courses插入失败

虽然courses表插入失败，但是teachers表仍成功更新。

结果 消息				
	tid	tname	email	salary
1	1234567890	zs	my@zsu.edu.cn	3000

	tid	tname	email	salary
1	1234567890	zs	my@zsu.edu.cn	999

练习3

编写一个包含事务的存储过程，用于更新courses表的课时。如果更新记录的cid不存在，则输出“课程信息不存在”，其他错误输出“修改课时失败”，如果执行成功，则输出“课时修改成功”。调用该存储过程，演示更新成功与更新失败的操作。

- 定义存储过程

```
create procedure updatecourseinfo
    @courseid char(10),
    @hour int,
    @returnString varchar(100) out
as
begin tran
    if not exists (select cid from courses where cid=@courseid)
    begin
        select @returnString='课程信息不存在'
        goto onerror
    end
    update COURSES set hour=@hour where cid=@courseid
    if @@ERROR > 0
    begin
        select @returnString='修改课时失败'
        goto onerror
    end

    select @returnString='修改课时成功'
    print @returnString
    commit tran
```

100 %

消息

命令已成功完成。

- 演示更新失败

```
declare @courseid char(10)
declare @hour int
declare @returnString varchar(10)
exec updatecourseinfo '1', 10, @returnString out
```

100 %

消息

课程信息不存在

- 演示更新成功

```
declare @courseid char(10)
declare @hour int
declare @returnString varchar(10)
exec updatecourseinfo '10001', 15, @returnString out
```

100 %

消息

(1 行受影响)
修改课时成功
修改课时成功