

第十三次实验

(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。

```
1 select top 1 * from students
2 SET XACT_ABORT ON--如果产生错误自动回滚
3 begin tran
4     update STUDENTS set sname='sty' where sid='1'
5     select top 1 * from students
6     begin tran
7         insert into TEACHERS values('200003125','','','')
8     commit tran
9 commit tran
10 select top 1 * from students
```

(1 行受影响)

(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_TEACHERS”。不能在对象“dbo.TEACHERS”中插入重复键。重复键值为 (200003125)。

完成时间: 2021-12-06T15:10:55.8423560+08:00

	sid	sname	email	grade
1	1	hhh	NULL	NULL

	sid	sname	email	grade
1	1	sty	NULL	NULL

	sid	sname	email	grade
1	1	hhh	NULL	NULL

(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。

```
1 select top 1 * from teachers
2 begin tran tran1
3     update teachers set tname='sty' where tid='200001000'
4     select top 1 * from teachers
5     save tran tran1_done
6     insert into courses values('10001','','')
7     if @@error!=0 or @@ROWCOUNT>1 begin
```

```

8         rollback tran tran1_done
9         return
10    end
11    commit tran tran1
12    go
13    select top 1 * from teachers
14    go

```

(1 行受影响)

(1 行受影响)

(1 行受影响)

消息 2627, 级别 14, 状态 1, 第 8 行
违反了 PRIMARY KEY 约束“PK_COURSES”。不能在对象“dbo.COURSES”中插入重复键。重复键值为 (10001)。
语句已终止。

(1 行受影响)

完成时间: 2021-12-06T15:28:46.4991432+08:00

	tid	tname	email	salary
1	200001000	LXL	s4zrck@pew.net	3024

	tid	tname	email	salary
1	200001000	sty	s4zrck@pew.net	3024

	tid	tname	email	salary
1	200001000	sty	s4zrck@pew.net	3024

(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。

```

1    go
2    create procedure updatecourseINFO
3        @courseid char(10),
4        @hour int,
5        @returnString varchar(100) out
6    as
7    begin tran
8        if not exists(select cid from courses where cid=@courseid)
9            begin
10                select @returnString='课程信息不存在'
11                goto onerror
12            end
13        update COURSES set hour=@hour where cid=@courseid
14        if @@error<>0
15            begin
16                select @returnString='修改课时失败'
17                goto onerror

```

```

18         end
19         select @returnString='课程修改成功'
20         print @returnString
21     commit tran
22 onerror:
23         print @returnString
24         rollback tran
25 go

```

命令已成功完成。

完成时间: 2021-12-06T15:37:22.8021163+08:00

```

1 declare @courseid char(10)
2 declare @hour int
3 declare @returnString varchar(100)
4 exec updatecourseINFO '10001',90,@returnString out

```

(1 行受影响)

课程修改成功

课程修改成功

消息 3903, 级别 16, 状态 1, 过程 updatecourseINFO, 行 23 [批起始行 0]

ROLLBACK TRANSACTION 请求没有对应的 BEGIN TRANSACTION。

完成时间: 2021-12-06T15:42:23.8033693+08:00

```

1 declare @courseid char(10)
2 declare @hour int
3 declare @returnString varchar(100)
4 exec updatecourseINFO '100010',90,@returnString out

```

课程信息不存在

完成时间: 2021-12-06T15:44:49.7943834+08:00