

中山大学数据科学与计算机学院本科生实验报告

(2020学年秋季学期)

课程名称：数据库系统 任课教师：刘玉葆

年级&班级	18级 计科一班	专业(方向)	计科 (大数据与人工智能方向)
学号	18340014	姓名	陈嘉宁
电话	18475934419	Email	734311072@qq.com
开始日期	2020/12/20	完成日期	2020/12/20

一、实验题目

SQL事务控制语言

二、实验目的

1. 熟悉SQL Server的**事务控制语言**，能够熟练使用事务控制语言来编写事务处理程序。

三、实验过程

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：

实现该过程的SQL语句为：

```
1  begin tran T1
2      update STUDENTS
3      set email = '20201218'
4      where sid = '10001'
5
6      begin tran T2
7          insert into TEACHERS
8          values('200202973','gdgcxjhj','3529@knnbo.edu',3138)
9
10     if @@ERROR<>0
11         begin
12             rollback tran T1
13             print '在教师表中插入失败'
14             return
15         end
16     commit tran T2
17 commit tran T1
```

在事务T1中对STUDENTS表进行更新操作，在T2中对TEACHERS表进行插入操作。在这里，我尝试向TEACHERS表中插入一个已存在的元组，这势必会执行失败并返回错误，接着执行if语句下的过程，回滚事务T1。

执行结果如下：

```
SQLQuery1.sql - (L:\KLS675B\deli (55))*
begin tran T1
    update STUDENTS
    set email = '20201218'
    where sid = '10001'

    begin tran T2
        insert into TEACHERS
        values ('200202973', 'gdgcxjhj', '3529@knnbo.edu', 3138)

    if @@ERROR <> 0
    begin
        rollback tran T1
        print '在教师表中插入失败'
        return
    end
    commit tran T2
commit tran T1
```

消息

(0 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。
在教师表中插入失败

可以发现，执行结果中显示“0行受影响”，说明事务T1中更新的操作已被回滚。

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：

实现该过程的SQL语句如下：

```
1  select * from STUDENTS
2  where sid = '60000001'
3
4  begin tran T1
5      update STUDENTS
6      set email = '20201218'
7      where sid = '60000001'
8
9      save tran SP1
10         insert into TEACHERS
11         values('200202973', 'gdgcxjhj', '3529@knnbo.edu', 3138)
12
13     if @@ERROR <> 0
14     begin
15         rollback tran SP1
16         print '在教师表中插入失败'
17         return
18     end
19     commit tran T2
20 commit tran T1
21
22 select * from STUDENTS
23 where sid = '60000001'
```

这一步与上题中的实现思路相似，只是将回滚的目标换成了在事务T1更新完成后的SP1处。

执行结果如下：

```
SQLQuery1.sql - (L:\KLS675B\dell (55))*
select * from STUDENTS
where sid = '60000001'

begin tran T1
update STUDENTS
set email = '20201218'
where sid = '60000001'

save tran SP1
insert into TEACHERS
values('200202973','gdqcxjhj','3529@knnbo.edu',3138)

if @@ERROR<>0
begin
rollback tran SP1
print '在教师表中插入失败'
return
end

commit tran T2
commit tran T1

select * from STUDENTS
where sid = '60000001'
```

结果 消息

(1 行受影响)

(1 行受影响)

消息 2627, 级别 14, 状态 1, 第 10 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。
在教师表中插入失败

可以从输出的“1行受影响”看出，事务T1中的更新操作执行成功。

3. 编写一个包含事务的存储过程，用于插入teachers表数据。如果插入记录的tid已存在，则输出“数据已存在”，其他错误输出“插入数据失败”，如果执行成功，则输出“插入数据成功”。调用该存储过程，演示插入成功与插入失败的操作：

实现该过程的SQL语句如下：

```
1 create procedure insertteachersinfo
2     @tmptid char(10),
3     @tmptname char(30),
4     @tmpemail char(30),
5     @tmpsalary int,
6     @returnString varchar(100) out
7 as
8 begin tran
9     if exists(select tid from TEACHERS where tid = @tmptid)
10         begin
11             select @returnString='课程信息已存在'
12             goto onerror
13         end
14     insert into TEACHERS
15     values(@tmptid,@tmptname,@tmpemail,@tmpsalary)
16     if @@ERROR<>0
17         begin
18             select @returnString='添加失败'
19             goto onerror
20         end
21     select @returnString='添加成功'
22     print @returnString
23 commit tran
24
25 onerror:
26     print @returnString
27     rollback tran
```

创建该存储过程后，通过向TEACHERS表插入数据来验证其功能：

- 插入已存在的主键:

```
1 declare @tmptid char(10)
2 declare @tmptname char(30)
3 declare @tmpemail char(30)
4 declare @tmpsalary int
5 declare @returnString varchar(100)
6 exec insertteachersinfo
   '200202973','gdgcxjhj','3529@knnbo.edu',3138,@returnString out
```

执行结果如下:



```
SQLQuery1.sql - (L...KLS675B\dell (54))
declare @tmptid char(10)
declare @tmptname char(30)
declare @tmpemail char(30)
declare @tmpsalary int
declare @returnString varchar(100)
exec insertteachersinfo '200202973','gdgcxjhj','3529@knnbo.edu',3138,@returnString out
```

消息
课程信息已存在

- 插入新的元组:

```
1 declare @tmptid char(10)
2 declare @tmptname char(30)
3 declare @tmpemail char(30)
4 declare @tmpsalary int
5 declare @returnString varchar(100)
6 exec insertteachersinfo
   '20201218_1','gdgcxjhj','3529@knnbo.edu',3138,@returnString out
```

执行结果如下:



```
SQLQuery1.sql - (L...KLS675B\dell (54))
declare @tmptid char(10)
declare @tmptname char(30)
declare @tmpemail char(30)
declare @tmpsalary int
declare @returnString varchar(100)
exec insertteachersinfo '20201218_1','gdgcxjhj','3529@knnbo.edu',3138,@returnString out
```

消息
(1 行受影响)
添加成功
添加成功
消息 3503, 级别 16, 状态 1, 过程 insertteachersinfo, 第 27 行
ROLLBACK TRANSACTION 请求没有对应的 BEGIN TRANSACTION。

- 插入tname为null的元组 (设定tname not null) :

```
1 declare @tmptid char(10)
2 declare @tmptname char(30)
3 declare @tmpemail char(30)
4 declare @tmpsalary int
5 declare @returnString varchar(100)sql
6 exec insertteachersinfo
   '20201218_2','null','3529@knnbo.edu',3138,@returnString out
```

执行结果如下:

```
SQLQuery1.sql - (L:\KLS675B\dell (54))
declare @tmptid char(10)
declare @tmptname char(30)
declare @tmpemail char(30)
declare @tmpsalary int
declare @returnString varchar(100)
exec insertteachersinfo '20201218_2',null,'3529@knnbo.edu',3138,@returnString out
```

消息
消息 518, 级别 16, 状态 2, 过程 insertteachersinfo, 第 14 行
不能将值 NULL 插入列 'tname', 表 'School.dbo.TEACHERS'; 列不允许有 Null 值。INSERT 失败。
语句已终止。
添加失败

以上三个实例可以说明所创建的存储过程是有效的。但是这其中还有一个问题，由于SQL语句是顺序执行的，在调用该存储过程时，无论插入成功与否，均会执行“Onererror”的过程，也就导致在上述实例中插入成功的例子里出现报错信息。简要修改代码即可避免该问题，修改后的代码如下：

```
1  create procedure insertteachersinfo
2      @tmptid char(10),
3      @tmptname char(30),
4      @tmpemail char(30),
5      @tmpsalary int,
6      @returnString varchar(100) out
7  as
8  begin tran
9      if exists(select tid from TEACHERS where tid = @tmptid)
10         begin
11             select @returnString='课程信息已存在'
12             goto onerror
13         end
14     insert into TEACHERS
15     values(@tmptid,@tmptname,@tmpemail,@tmpsalary)
16     if @@ERROR<>0
17         begin
18             select @returnString='添加失败'
19             goto Onererror
20         end
21     select @returnString='添加成功'
22 commit tran
23 goto End
24
25 Onererror:
26     print @returnString
27     rollback tran
28
29 End:
30     print @returnString
```

四、实验感想

本次实验工作量较小，按照PPT中的引导能够较快地完成，且从中可以将理论课上学到的关于事务的知识很好地融会贯通。

