



中山大学数据科学与计算机学院

移动信息工程专业-数据库系统

本科生实验报告

(2017-2018 学年秋季学期)

课程名称：数据库系统实验

教学班级	15M1	专业（方向）	移动互联网
学号	15352408	姓名	张镓伟

一、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。

二、实验过程及结果

(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。

先查询一下 students 表的记录，准备对第一条记录作修改：

```
select * from STUDENTS order by sid
```

sid	sname	email	grade
1	800001216	gfrigs	hhce4@qhldj.gov

再查询一下 teachers 表的记录，准备插入一个与第一条记录 tid 相同的新纪录以造成冲突。

```
select * from teachers order by tid
```

tid	tname	email	salary
1	200003125	fqmmyi	wcjcg@glq.net

编写嵌套事务：

```
set XACT_ABORT ON
begin tran update_stu
    update students
    set sname='haha'
    where sid='800001216'
    begin tran insert_tea
        insert into TEACHERS
        values('200003125','newTeacher','123@123.com',3000)
    commit tran insert_tea
commit tran update_stu
```

(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。

发现 teachers 表的插入操作失败了
重新查询 students 表：



```
select * from STUDENTS order by sid
```

结果		消息		
sid	sname	email	grade	
1	800001216	gfwrgs	hhce4@qhldj.gov	1992

可以看到第一条记录并没有被修改，说明之前因为内层事务操作失败，外层事务也回滚了。

(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。

Courses 数据：

```
select * from COURSES
```

结果	消息		
cid	cname	hour	
1	10001	database	96

Teachers 数据，更新第一条数据：

```
sqlquery1.sql - (10001)freedom (55)
select * from teachers order by tid
```

<

结果 消息

	tid	tname	email	salary
1	200003125	fqmmys	wcjcq@glq.net	3928

带保存点的事物处理，先更新 teacher 表第一条记录的工资，再往 courses 表插入一条重复的记录造成插入失败：

```
set XACT_ABORT OFF
begin tran update_tea
update TEACHERS
set salary=4000
where tid='200003125'
save tran update_tea_done
insert into COURSES
values('10001', 'database', 96)
if @@ERROR!=0 or @@ROWCOUNT>1
begin
rollback tran update_tea_done
print '插入courses表失败'
return
end
commit tran update_tea
```

消息

(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_COURSES'。不能在对象 'dbo.COURSES' 中插入重复键。
语句已终止。
插入courses表失败

重新查询 teacher 表

```
select * from TEACHERS
```

结果	消息			
tid	tname	email	salary	
1	200003125	fqmmyi	wcjcq@glq.net	4000

发现第一条记录的工资被成功更新



(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。

`Courses` 表信息，准备对第一条记录作更新：

```
select * from COURSES
```

	cid	cname	hour
1	10001	database	96

创建存储过程：

```
create procedure updatecourses
    @course_cid char(10),
    @course_hour int,
    @returnString varchar(100) out
as
begin tran t1
    if not exists(select cid from COURSES where cid=@course_cid)
    begin
        select @returnString='课程信息不存在'
        goto onError
    end
    update COURSES set hour=@course_hour where cid=@course_cid
    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 t1
onError:
    print @returnString
    rollback tran
go
```

消息
命令已成功完成。

更新失败操作：

```
SQLQuery1.sql - (\\...UTE\freedom (55))*
declare @course_cid char(4)
declare @course_hour int
declare @returnString varchar(100)
exec updatecourses '10000',12,@returnString out
```

消息
课程信息不存在

更新成功操作：

```
SQLQuery1.sql - (\\...UTE\freedom (55))
declare @course_cid char(4)
declare @course_hour int
declare @returnString varchar(100)
exec updatecourses '10001',12,@returnString out
```

消息
(1 行受影响)
修改课时成功
修改课时成功
消息 3903, 级别 16, 状态 1, 过程 updatecourses, 第 23 行
ROLLBACK TRANSACTION 请求没有对应的 BEGIN TRANSACTION。



重新查询 `courses`，发现第一条数据的课时确被修改了

```
select * from COURSES
```

结果			
	cid	cname	hour
1	10001	database	12

三、 实验感想

这次实验我们熟悉了事物的操作，包括嵌套操作，设置保存点和创建一个过程。虽然只有 3 题，但是由于不熟悉，还是花了很长时间，另外这是第一次知道创建过程的操作，可以省不少事。