第一部分

第一题

SET XACT\_ABORT ON--如果产生错误自动回滚

go -- 用来分割不同的事务

begin tran outside

update STUDENTS set sname='ysy' where sid='800001216'

begin tran inside

insert TEACHERS values('200003125','','','')

commit tran inside

commit tran outside

内层插入失败后，外层没有执行全部回滚，去students表中验证

select \* from STUDENTS where sid='800001216'

名字并不叫ysy

第二题

go -- 用来分割不同的事务

begin tran

update TEACHERS set tname='ysy2' where tid='200003125'

SAVE TRAN tran\_upd\_teachers\_done

insert COURSES values('10001','','')

if @@ERROR!=0 OR @@ROWCOUNT>1

BEGIN

------ ROLLBACK

ROLLBACK TRAN tran\_upd\_teachers\_done

PRINT 'erroR happen when insert courses'

return

END

commit tran

查看teachers表可以看到名字被改成了ysy

第三题

go

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 ONERROE

END

-- NO ERROR INSERT NEW COURSE INFO

update COURSES set hour=@hour where cid=@courseid

IF @@ERROR<>0

BEGIN

SELECT @returnString='修改课时失败'

GOTO ONERROE

END

SELECT @returnString='课程修改成功'

PRINT @returnString

commit TRAN

ONERROE:

PRINT @returnString

ROLLBACK TRAN

Go

失败的案例：

declare @courseid char(10)

declare @hour int

declare @returnString varchar(100)

exec insertcourseINFO2 '100010',90,@returnString out

输出：

课程信息不存在

成功的案例

exec insertcourseINFO2 '10001',90,@returnString out

(1 行受影响)

课程修改成功

第二部分

第一题

代码1

go

BEGIN TRAN

UPDATE STUDENTS SET sname='ysysb' where sid='800001216'

WAITFOR DELAY '00:00:05'

SELECT \* FROM STUDENTS WHERE sid='800001216'

ROLLBACK TRAN

SELECT \* FROM STUDENTS WHERE sid='800001216'

代码2

go

set transaction isolation level read uncommitted

select \* from STUDENTS where sid='800001216'

if @@ROWCOUNT<>0

BEGIN

WAITFOR DELAY '00:00:05'

SELECT \* FROM STUDENTS WHERE sid='800001216'

END

在执行代码1的时候执行代码2 得到两个sname一个是ysysb一个是ysy，读了脏数据。

第二题

修改后的代码2

go

set transaction isolation level read committed

select \* from STUDENTS where sid='800001216'

if @@ROWCOUNT<>0

BEGIN

WAITFOR DELAY '00:00:05'

SELECT \* FROM STUDENTS WHERE sid='800001216'

END

可以看到输出结果都是ysy

第三题：

代码1

set transaction isolation level repeatable read

begin tran

select \* from students where sid='800001216'

if @@ROWCOUNT<>0

BEGIN

WAITFOR DELAY '00:00:05'

SELECT \* FROM STUDENTS WHERE sid='800001216'

END

ROLLBACK TRAN

代码2

set transaction isolation level repeatable read

UPDATE STUDENTS SET sname='ysyhello' where sid='800001216'

可以看到代码1输出的结果名字都是ysy，

而查看数据库发现名字已经被修改成了ysyhello

select \* from STUDENTS where sid='800001216'

重新编写代码2，让代码2删除对应的数据项：

因为存在外键的约束，所以先重新插入一个学生：

重写代码1

insert STUDENTS values('123','test','','')

set transaction isolation level repeatable read

begin tran

select \* from students where sid='123'

if @@ROWCOUNT<>0

BEGIN

WAITFOR DELAY '00:00:05'

SELECT \* FROM STUDENTS WHERE sid='123'

END

ROLLBACK TRAN

重写代码2

set transaction isolation level repeatable read

delete STUDENTS where sid='123'

看到代码1读出了两个相同的数据，但其实在第二次读取的时候，sid为123的学生已经被删除了，但还是读取了，此时就是幻象读。

第四题：

代码一：设置可串行化连续读两次存在的id 123

set transaction isolation level serializable

begin tran

select \* from students where sid='123'

WAITFOR DELAY '00:00:05'

SELECT \* FROM STUDENTS WHERE sid='123'

ROLLBACK TRAN

代码二 插入id为123的记录

set transaction isolation level serializable

update STUDENTS set sname='hello' where sid='123'

代码1查询结果显示都一样名字都是ysy，但在数据表中查询，名字发生了改变变成了hello，因此可串行化放置了其他用户更新数据