# 实验八 事务的创建与管理实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 1181103114 | 姓名： | 章磊 |
| 班级： | 计算机1181 | 指导老师： | 冯万利 |
| 实验时间： | 5月24日 | 实验地点： | 家 |

# 一、实验目的和要求

要求掌握事务的定义、使用，能够使用和和事务对数据库进行简单的并发操作。

# 二、实验内容

(1)掌握定义事务的方法。

(2)掌握事务的提交方法。

(3)利用事务处理并发操作。

# 三、实验步骤

## 一、事务的基本概念

事物一般可分为两类，即系统事务和用户定义的事物。系统事务又称为隐式事务，指某些特定的SQL语句由系统单独作为一个事务处理，包括主要语句如下：

所有的CREATE语句。

所有的DROP语句。

INSERT,UPDATE,DELETE语句

例如，执行如下创建表语句：

CREATE TABLE xx

(

f1 int not null,

f2 char(10),

f3 varchar(30),

)

这条语句本身就构成一个事务，它要么建立含三列的表结构，要么对数据库没有任何影响。

1. 开始事务

语句格式：

BEGIN TRANSACTION | TRAN

1. 结束事务

语句格式：

COMMIT

1. 事务的回滚

语句格式：

ROLLBACK

## 二、事务处理

## 7.2.1简单事务的处理。

(1)定义一个事物，将表S中的姓名“李小刚”改成“李晓岚”，并提交该事务

USE JXGL

GO

BEGIN TRANSACTION

UPDATE S

SET sname='李晓岚'

WHERE sname='李小刚'

COMMIT

GO

(2)定义一个事务，将表SC中所有选了B1号课程的学生分数增加5%，并提交该事务

USE JXGL

GO

DECLARE @TranName varchar(20)

SELECT @TranName= 'Add\_Grade'

BEGIN TRAN @TranName

UPDATE SC SET grade=grade\*105%

WHERE cno='B1'

COMMIT TRAN @TranName

GO

(3)定义一个事务，将删除表SC中“S9”的学生成绩和S表中学号为“S9”的学生记录定义为一个事务，执行该事务并提交。

USE JXGL

GO

DECLARE @TranName varchar(20)

SELECT @TranName='Del\_Grade'

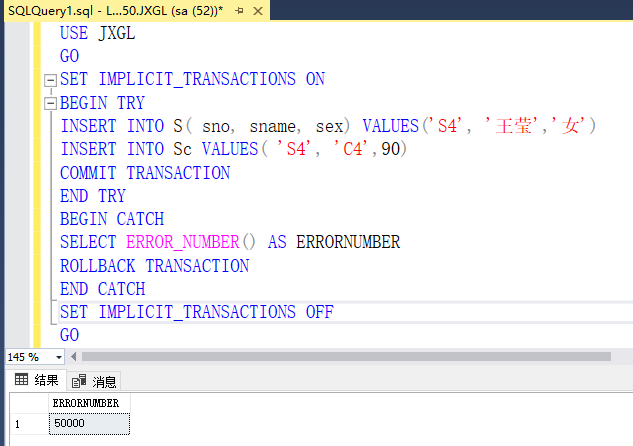
Begin TRAN @TranName

DELETE FROM SC where sno='S9'

DELETE FROM S where sno='S9'

COMMIT TRAN @TranName

GO



对表S进行插入操作

## 实验7.2.2

(1)显式事务。当对表SC进行插入操作时S表中没有“S10”记录，所以插入失败。

USE JXGL

GO

BEGIN TRY

BEGIN TRANSACTION

INSERT INTO SC( sno, cno) VALUES('S10','C4')

COMMIT TRANSACTION

END TRY

BEGIN CATCH

SELECT ERROR\_NUMBER() AS ERRORNUMBER

ROLLBACK TRANSACTION

END CATCH

GO

(2)隐式设置事务。当对表S进行插入操作时S表中有“S4”，所以插入失败。如图7.2.2.1所示

use jxgl

go

set implicit\_transactions on

begin try

insert into S(sno,sname,sex)values('S4','王莹','女')

insert into SC values('S4','C4','90')

commit transaction

end try

begin catch

select error\_number() as errornumber

rollback transaction

end catch

set implicit\_transactions off

go

## 实验7.2.3 SQL事务处理实验。

(1)当第一条有效，第二条有误时，会保证两条事务都不写入数据库。如图

USE JXGL

GO

BEGIN TRAN

INSERT INTO S( sno, sname, sex, age, sdept) VALUES('S20','王荣格','女',21, 'CS')

INSERT INTO S( sno, sname, sex, age, sdept) VALUES( 'S20','王连福', '男',22, 'MA')

IF @@ERROR<> 0 ROLLBACK TRAN

COMMIT TRAN

GO

(2)当第一条有误，第二条有效时，第一条不写入数据库，第二条写入。

USE JXGL

GO

INSERT INTO s( sno, sname, sex, age, sdept) VALUES('S20','王荣格','女',21, 'CS')

BEGIN TRAN

INSERT INTO S( sno, sname, sex, age, sdept) VALUES( 'S20','王连福','男',22, 'MA')

IF @@ERROR<> 0 ROLLBACK TRAN

INSERT INTO S( sno, sname, sex, age, sdept) VALUES ('S21', '钱云华', '男',21, 'MA')

IF @@ERROR<> 0 ROLLBACK TRAN

COMMIT TRAN

GO

(3)不管哪条数据有问题，都保证不写入数据库。

USE JXGL

GO

BEGIN TRY

BEGIN TRAN tempTran

INSERT INTO S( sno,sname,sex,age,sdept) VALUES('S21', '王金霞','女',22, 'IS')

INSERT INTO C(cno,cname, credit,pcno) VALUES('C14','汇编语言','CS','马政')

INSERT INTO SC(sno,cno,grade) VALUES ('S20','C14',78)

COMMIT TRAN tempTran

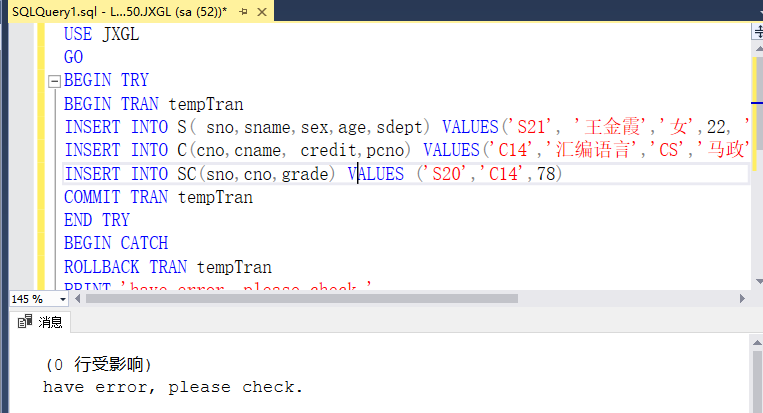
END TRY

BEGIN CATCH

ROLLBACK TRAN tempTran

PRINT 'have error, please check.'

END CATCH

GO

## 7.2.4简单银行转账程序设计。

创建数据库BANK及数据表。

CARDINFO(cardid char(19),pass char(6),balance meney);

TRANINFO(transdate date,cardid char(19),transtype char(4),transmoney meney)

1. 创建存储过程proc\_takemoney,完成转账过程中的账户余额变换及存储支取和存入记录。

USE BANK

GO

CREATE PROCEDURE proc\_takemoney

@cardID1 varchar(20),

@pwd char(6),

@takemoney money,

@cardID2 varchar(20)

AS

SET NOCOUNT ON

DECLARE @err INT

SET @err=0

IF EXISTS(SELECT\*FROM CARDINFO WHERE cardid= @cardID1 AND pass=@pwd)

BEGIN

BEGIN TRAN

UPDATE CARDINFO SET BALANCE=BALANCE + @takemoney WHERE cardID=@cardID2

SET @err=@err + @@error

UPDATE CARDINFO SET balance=balance - @takemoney WHERE cardID=@cardID1

SET @err=@err + @@error

INSERT INTO TRANINFO ( transdate,cardid,transtype,transmoney) VALUES( GETDATE(),@cardID1,'支取',@takemoney)

SET @err=@err + @@error

INSERT INTO TRANINFO ( transdate, cardid, transtype, transmoney) VALUES ( GETDATE( ),@cardID1,'存人',@takemoney)

SET @err = @err + @@error

SELECT TOP 2￥FROM TRANINFO ORDER BY TRANSDATE DESC

IF(@err<>0)

BEGIN

ROLLBACK TRAN

END

ELSE

COMMIT TRAN

END

ELSE

BEGIN

PRINT @err

END

GO

1. 调用存储过程proc\_takemoney,完成账户A向账户B转账1000元。

USE BANK

GO

DECLARE @cardID1 varchar(19),@cardID2 varchar(19),@pwd char(6),@takemoney money

SET @cardID1 = '6227300292160184140'

SET @pwd = '123456'

SET @cardID2 = '6227300274694561341'

SET @takemoney=6000

EXEC dbo.proc\_takemoney @cardID1 ,@pwd, @takemoney,@cardID2

GO

## 实验10.5复杂事务的处理。

设计并执行事务:学生“王晓霞”打算选修“离散数学”课程。根据规定,此门课程选修的人数最多为30人,该生是否可以选修此门课程,并给出提示结果。

USE JXGL

GO

BEGIN TRAN

DECLARE @person\_num tinyint, @c\_no char(4), @s\_no char(8)

SELECT @c\_no= cno FROM C WHERE cname= '离散数学'

SELECT @s\_no= sno FROM S WHERE sname= '王晓霞'

SELECT @person\_num = COUNT(\*) FROM SC WHERE cno= @c\_no

IF @person\_num < 30

BEGIN

INSERT INTO SC(sno, cno) VALUES(@s\_no, @c\_no)

COMMIT TRAN

PRINT '王晓霞同学选修离散数学课程注册成功!'

END

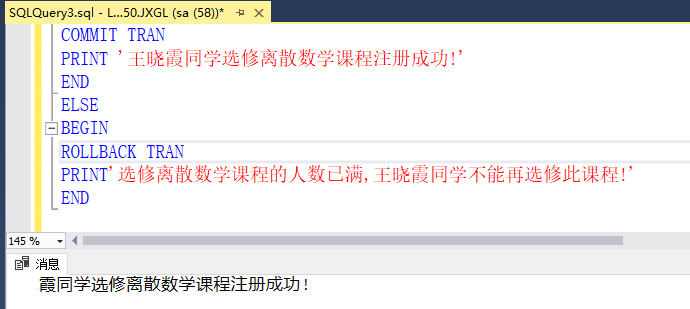
ELSE

BEGIN

ROLLBACK TRAN

PRINT'选修离散数学课程的人数已满,王晓霞同学不能再选修此课程!'

END



设计并执行事务

## 三、 事务的保存点

将SQL Server 2012中，ROLLBACK 还可以加上选项[TRAN [SACTION]<保存点名>|<保存点变量名>]，保存点名或保存点变量名可用SAVE TRANSACTION语句设置：

save tran[saction]{保存点名|保存点变量名}

## 3.3.1事务的保存点实验。

(1)定义一个事务，向JXGL数据库的S表中插入一行数据，然后删除该行。

USE JXGL

GO

BEGIN TRANSACTION

INSERT INTO S(sno, sname, sex, age, sdept)

VALUES('S14','高艳霞','女',22, 'IS')

SAVE TRAN My\_sav

DELETE FROM S

WHERE sname = '高艳霞'

ROLLBACK TRAN My\_sav

COMMIT TRAN

GO

执行上述事务后可知新插入的数据行并没有被删除，因为事务中使用rollback语句将操作回滚到保存点my\_sav,即删除前的状态。

(2)退到指定的事务保存点。

USE JXGL

GO

BEGIN TRAN

SAVE TRAN spl

INSERT INTO S(sno, sname, sex, age, sdept) VALUES('S21','王忠明', 'M',21, 'MA')

SAVE TRAN sp2

INSERT INTO S(sno, sname, sex, age, sdept) VALUES( 'S22', '田秀荣','F',22, 'IS')

SAVE TRAN sp3

INSERT INTO S(sno, sname, sex, age, sdept) VALUES('S23', '马忠波', 'M',20, 'CS')

SAVE TRAN sp4

ROLLBACK TRAN sp3

COMMIT TRAN

GO

# 四、总结与体会

在数据库中事务是作为单个逻辑单元执行的一系列操作，它是一个不可分割的工作逻辑单元。它包含了一组数据库操作命令，这组命令要么全部执行，要么全部不执行。

举个例子，我们经常用到的 ATM 存取款机，比如转账的时候，是先减去转出账户的金额，然后再在指定转入账户的金额加上转出的金额。如果刚好这个时候转出的操作已经执行完成，但是由于系统的故障，导致转入的操作失败了。那么怎么办？这就需要用到事务了，只要事务里面有一条命令未成功执行，那么数据就会回滚到事务开始之前的状态。

我们在使用时要注意按同一顺序访问数据库对象资源。避免事务中的用户交互，即事务中等待用户输入、提交等操作。

总而言之，操作一定要注意安全，将死锁减至最少。