1. **触发器更新（6.20，6.21，6.22，6.26，6.27）**

**insert 1**

USE BookManage

IF EXISTS(SELECT name FROM sysobjects WHERE NAME='tbr\_insert' and type='TR')

DROP TRIGGER tbr\_insert

GO

CREATE TRIGGER tbr\_insert ON BorrowRecord

FOR INSERT AS

IF EXISTS(SELECT \* FROM inserted a WHERE a.Borrower\_ID NOT IN

(SELECT b.Borrower\_ID FROM Borrower b)

OR a.ISBN NOT IN (SELECT c.ISBN FROM BookInfo c)

OR EXISTS(SELECT \* FROM BookInfo WHERE stock<=0

AND a.ISBN= BookInfo.ISBN) )

BEGIN

RAISERROR ('违背数据的一致性.', 16, 1)

ROLLBACK TRANSACTION

END

ELSE

BEGIN

UPDATE BookInfo SET stock=stock-1 WHERE BookInfo.ISBN IN(SELECT inserted.ISBN FROM inserted)

END

GO

**update 1**

USE BookManage

IF EXISTS(SELECT name FROM sysobjects WHERE NAME='tbr\_update' and type='TR')

DROP TRIGGER tbr\_update

GO

CREATE TRIGGER tbr\_update ON BorrowRecord

FOR UPDATE AS

BEGIN

UPDATE BookInfo SET stock=stock+1 WHERE BookInfo.ISBN IN(SELECT inserted.ISBN FROM inserted)

END

GO

**update 2**

USE BookManage

IF EXISTS(SELECT name FROM sysobjects WHERE NAME='tbr\_update' and type='TR')

DROP TRIGGER tbr\_update

GO

CREATE TRIGGER tbr\_update ON BorrowRecord

FOR UPDATE AS

IF EXISTS(select \* from inserted where inserted.Status=1)

BEGIN

UPDATE BookInfo SET stock=stock+1 WHERE BookInfo.ISBN IN(SELECT inserted.ISBN FROM inserted)

END

GO

**insert and update 1**

use BookManage

if exists(select name from sysobjects where name='br\_change' and type='TR')

drop trigger br\_change

go

create trigger br\_change on BorrowRecord for insert,update as

if exists(select \* from inserted a where a.Borrower\_ID not in(select b.Borrower\_ID from Borrower b)

or a.ISBN not in(select c.ISBN from BookInfo c)

or (exists(select \* from BookInfo where stock<=0 and a.ISBN=BookInfo.ISBN) and exists(select Status from inserted where Status='0'))

or (exists(select \* from BookInfo where stock>=copies and a.ISBN=BookInfo.ISBN) and exists(select Status from inserted where Status='1')))

begin

raiserror('违背数据一致性',16,1)

rollback transaction

end

else if exists(select \* from inserted where Status='0')

begin

update BookInfo set stock=stock-1 where BookInfo.ISBN in(select inserted.ISBN from inserted)

end

else

begin

update BookInfo set stock=stock+1 where BookInfo.ISBN in (select deleted.ISBN from deleted)

end

go

**insert and update 2**

use BookManage

if exists(select name from sysobjects where name='br\_change' and type='TR')

drop trigger br\_change

go

create trigger br\_change on BorrowRecord for insert,update as

if exists(select \* from inserted a where a.Borrower\_ID not in(select b.Borrower\_ID from Borrower b)

or a.ISBN not in(select c.ISBN from BookInfo c)

or (exists(select \* from BookInfo where stock<=0 and a.ISBN=BookInfo.ISBN) and exists(select Status from inserted where Status='0'))

or (exists(select \* from BookInfo where stock>=copies and a.ISBN=BookInfo.ISBN) and exists(select Status from inserted where Status='1')))

begin

raiserror('违背数据一致性',16,1)

rollback transaction

end

else if exists(select \* from inserted,deleted where inserted.Status='0' and deleted.Status='0')

begin

update BookInfo set stock=stock-1 where BookInfo.ISBN in(select inserted.ISBN from inserted)

end

else

begin

update BookInfo set stock=stock+1 where BookInfo.ISBN in (select deleted.ISBN from deleted)

end

go

**insert and update 3**

use BookManage

if exists(select name from sysobjects where name='br\_change' and type='TR')

drop trigger br\_change

go

create trigger br\_change on BorrowRecord for insert,update as

if exists(select \* from inserted a where a.Borrower\_ID not in(select b.Borrower\_ID from Borrower b)

or a.ISBN not in(select c.ISBN from BookInfo c)

or (exists(select \* from BookInfo,deleted where stock<=0 and a.ISBN=BookInfo.ISBN and a.Deadline != deleted.Deadline))

or (exists(select \* from BookInfo where stock>=copies and a.ISBN=BookInfo.ISBN and a.Status='1') ))

begin

raiserror('违背数据一致性',16,1)

rollback transaction

end

else if exists(select \* from inserted,deleted where inserted.Status='0' and deleted.Status='0')

begin

update BookInfo set stock=stock-1 where BookInfo.ISBN in(select inserted.ISBN from inserted)

end

else

begin

update BookInfo set stock=stock+1 where BookInfo.ISBN in (select deleted.ISBN from deleted)

end

go

**触发器测试**

insert into BorrowRecord values('123','234','2018-6-26','2018-6-26','0')

update BorrowRecord set Deadline=DATEADD(mm,1,Deadline) where Borrower\_ID='123' and ISBN='234' and Status='0'

update BorrowRecord set Status=Status+1 where Borrower\_ID='123' and ISBN='234'

1. **存储过程更新（6.27）**

**with output**

use BookManage

go

create procedure bkinfo

@isbn varchar(20),

@name varchar(50) output,

@writer varchar(20) output,

@publish varchar(30) output,

@cop int output,

@sto int output

as

select @name=name,@writer=author,@publish=publisher,@cop=copies,@sto=stock

from BookInfo where ISBN=@isbn

go

**without output**

use BookManage

go

create procedure bkinfo

@isbn varchar(20)

as

select \* from BookInfo where ISBN=@isbn

go

**测试存储过程with output**

declare @name varchar(50),

@writer varchar(20),

@publish varchar(30),

@cop int,

@sto int

exec bkinfo '123',

@name output,

@writer output,

@publish output,

@cop output,

@sto output

select '123',@name, @writer, @publish, @cop, @sto

go

**java调用方式更新**

rs=ps.executeQuery("exec bkinfo '"+isbn+"'");

/\*CallableStatement cst=con.prepareCall("call bkinfo(?,?,?,?,?,?)");

System.out.println(isbn);

cst.setString(1, isbn);

cst.registerOutParameter(2, Types.VARCHAR);

cst.registerOutParameter(3, Types.VARCHAR);

cst.registerOutParameter(4, Types.VARCHAR);

cst.registerOutParameter(5, Types.INTEGER);

cst.registerOutParameter(6, Types.INTEGER);

cst.execute();\*/

if(rs.next())

{

String bname=rs.getString("name");

String writer=rs.getString("author");

String pub=rs.getString("publisher");

int cop=rs.getInt("copies");

int sto=rs.getInt("stock");

}

1. **JTable构造及事件响应（6.19-6.20）[未保存旧代码]**

**public** JPanel InitInfoTable(String ID){

JPanel jp=**new** JPanel();

JTable table=**new** JTable(**new** InfoTable(ID));

table.addMouseListener(**new** MouseAdapter() {

**public** **void** mouseClicked(MouseEvent e){

**if**(e.getClickCount()==2)

{

**int** row=((JTable)e.getSource()).rowAtPoint(e.getPoint());

**new** BorrowerBook(ID,getInfo(ID),row);

dispose();

}

}

});

JScrollPane scroll=**new** JScrollPane(table);

jp.add(scroll);

**return** jp;

}

**private** **class** InfoTable **extends** AbstractTableModel{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

String[] columnNames={"书号","书名","借阅时间","归还期限"};

String[][] obj;

**public** InfoTable(String ID){

obj=getInfo(ID);

}

@Override

**public** **int** getRowCount() {

// **TODO** Auto-generated method stub

**return** obj.length;

}

@Override

**public** **int** getColumnCount() {

// **TODO** Auto-generated method stub

**return** columnNames.length;

}

@Override

**public** String getColumnName(**int** column) {

**return** columnNames[column];

}

@Override

**public** Object getValueAt(**int** rowIndex, **int** columnIndex) {

// **TODO** Auto-generated method stub

**return** obj[rowIndex][columnIndex];

}

}