建立用户表：

create table users(

account varchar(30) primary key,

passwd varchar(30) not null,

id varchar(10) check (id in ('管理员','读者'))

)

建立读者表：

create table readers(

borrowid varchar(30) primary key,

rname varchar(30) not null,

sex varchar(2) check (sex in ('男','女')),

job varchar(10),

rCurNum int,

rBorrowedNum int,

dept varchar(20),

phone varchar(20),

account varchar(30) foreign key references users(account)

on delete cascade

on update cascade

)

建立图书表：

create table books(

isbn varchar(30) primary key,

bname varchar(30) not null,

pub varchar(30) not null,

author varchar(15) not null,

storeNum int,

bCurNum int,

available char(2) check (available in ('是','否'))

)

增加account的unique约束

alter table readers add constraint UQ\_1 unique(account);

建立借阅关系表：

create table rb(

borrowid varchar(30) references readers(borrowid)

on delete cascade

on update cascade,

isbn varchar(30) references books(isbn)

on delete cascade

on update cascade,

startDate date,

duration int,

returnDate date,

fine int,

primary key(borrowid,isbn)

)

给管理员角色赋予权限：

grant all privileges

on readers

to admin

grant all privileges

on books

to admin

给读者角色赋予权限：

grant insert,delete,select

on rb

to reader

grant select

on readers

to reader

grant select on books

to reader;

grant update

on books(bCurNum)

to reader

加入触发器防止读者表的插入更新项的account不是用户表中的读者：

create trigger insertReaderOnly

on readers

for insert

as

begin

if exists(select inserted.account from inserted,users where inserted.account=users.account and users.id!='读者')

begin

delete from readers where account in (select account from inserted)

and account in (select account from users where id!='读者')

RAISERROR ('该读者未以读者身份注册!',15,1)

end

end

create trigger updateReaderOnly

on readers

for update

as

begin

if exists(select inserted.account from inserted,users where inserted.account=users.account and users.id!='读者')

begin

rollback

RAISERROR ('该读者未以读者身份注册!',15,1)

end

end

借书的触发器：

create trigger borrowTrigger

on rb

for insert

as

begin

declare @borrowid varchar(30)

declare @isbn varchar(30)

select @borrowid=borrowid,@isbn=isbn from inserted

if((select available from books where isbn=@isbn)='否')

begin

delete from rb where borrowid=@borrowid and isbn=@isbn

RAISERROR ('该书不可借!',15,1)

end

else if(not exists(select \* from rb where CONVERT(varchar(10),GETDATE(),120)>returnDate and borrowid=@borrowid and isbn!=@isbn))

begin

update books set bCurNum=bCurNum-1

where isbn=@isbn

update readers set rCurNum=rCurNum-1

where borrowid=@borrowid

if((select bCurNum from books where isbn=@isbn)=0)

begin

update books set available='否' where isbn=@isbn

end

end

else

begin

delete from rb where borrowid=@borrowid and isbn=@isbn

RAISERROR ('该读者当前还有未还书籍，不能借书!',15,1)

end

end

还书触发器：

create trigger returnTrigger

on rb

for delete

as

begin

declare @isbn varchar(30)

set @isbn=(select isbn from deleted)

update books set bCurNum=bCurNum+1 where isbn=@isbn

update books set available='是' where isbn=@isbn

end