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

grant select

on rb

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 //登录时能够判断身份

on users

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' or (select rCurNum from readers where borrowid=@borrowid)<=0)

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

update readers set rBorrowedNum=rBorrowedNum+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)

declare @borrowid varchar(30)

select distinct @isbn=isbn from deleted

select distinct @borrowid=borrowid from deleted

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

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

update readers set rCurNum=rCurNum+1 where borrowid=@borrowid

update readers set rBorrowedNum=rBorrowedNum-1 where borrowid=@borrowid

end

设计一个存储过程，以图书编号为输入参数，返回借阅该图书但未归还的读者姓名和借书证号（存储过程没有返回值，这里写成存储函数）：

create function getUnReturnReaders

(

@isbn varchar(30)

)

returns table

as

return

(

select readers.rname,readers.borrowid from readers,rb

where readers.borrowid=rb.borrowid and

CONVERT(varchar(10),GETDATE(),120)>returnDate and

isbn=@isbn

)

读者资料查询：设计一个有多个输入参数的存储过程，返回读者的详细信息。设计另一存储过程并以读者借书证号为输入参数，返回该读者未归还的图书名称和图书编号（存储过程没有返回值，这里写成存储函数）：

读者资料查询的存储函数：

create function getReaderInfo

(

@borrowid varchar(30),

@rname varchar(30),

@sex varchar(5),

@job varchar(30),

@rCurNum int,

@rBorrowedNum int,

@dept varchar(30),

@phone varchar(15),

@account varchar(30)

)

returns table

as

return

(

select \* from readers where

borrowid=@borrowid and rname=@rname

and sex=@sex and job=@job and rCurNum>=@rCurNum

and rBorrowedNum>=@rBorrowedNum and dept=@dept

and phone=@phone and account=@account

)

查看读者未还的图书名称和图书号：

create function getUnReturnBooks

(

@borrowid varchar(30)

)

returns table

as

return

(

select bname,books.isbn from books,rb

where books.isbn=rb.isbn and rb.borrowid=@borrowid

and CONVERT(varchar(10),GETDATE(),120)>returnDate

)

到期图书查询：设计一个视图，返回所有逾期未归还的图书的编号、书名、读者姓名等信息：

create view delayedInfos

as

select rb.isbn,rname,bname from rb,books,readers

where rb.isbn=books.isbn and rb.borrowid=readers.borrowid

and CONVERT(varchar(10),GETDATE(),120)>returnDate

加快数据检索速度，用图书编号为图书信息表建立索引：

create clustered index IsbnIndex on books(isbn)

借阅处理：为借阅信息表设计INSERT触发器，在读者借阅时更改ISBN类别信息表，且可借数量减1，图书信息表是否可借列的值变为“不可借”，读者信息表中该类读者已借阅数加1：

参考上面的借书触发器

还书处理：为借阅信息表设计UPDATE触发器，在该表的归还日期列被更改后，将图书信息表的是否可借列的值变为“可借”，读者信息表中已借数量减1及ISBN类别信息表中可借数量加1：

参考以上的还书触发器