目 录

[1 任务内容 1](#_Toc53649843)

[2 基本要求 1](#_Toc53649844)

[3 系统需求分析 2](#_Toc53649845)

[3.1需求分析阶段的目标与任务 2](#_Toc53649846)

[3.1.1处理对象 2](#_Toc53649847)

[3.1.2 处理功能及要求 2](#_Toc53649848)

[3.2需求分析阶段成果 3](#_Toc53649849)

[3.2.1 读者借/还书流程图 3](#_Toc53649850)

[3.3.2 新书籍增加流程图 3](#_Toc53649851)

[3.3 图书管理系统字典 4](#_Toc53649852)

[3.3.1 数据字典 4](#_Toc53649853)

[3.3.2 触发器字典 6](#_Toc53649854)

[3.3.3 存储过程字典 6](#_Toc53649855)

[4 概念结构设计 7](#_Toc53649856)

[4.1 系统功能模块图 7](#_Toc53649857)

[4.2 E-R图 8](#_Toc53649858)

[4.3数据库逻辑结构设计 8](#_Toc53649859)

[5 数据库实施阶段 9](#_Toc53649860)

[5.1 建立数据库 9](#_Toc53649861)

[5.2 建立数据表 9](#_Toc53649862)

[5.3 建立触发器 11](#_Toc53649863)

[5.4 建立存储过程 15](#_Toc53649864)

[6 系统调试和测试 23](#_Toc53649865)

[6.1 数据存储 23](#_Toc53649866)

[6.2借书/还书功能 25](#_Toc53649867)

[6.2.1 借书 25](#_Toc53649868)

[6.2.2 还书 25](#_Toc53649869)

[6.3 查询功能 26](#_Toc53649870)

[7 课程设计体会和总结 27](#_Toc53649871)

# 1 任务内容

根据所学的数据库原理与程序设计的知识，能够针对一个小型的图书管理系统，进行数据库系统的需求分析，概念结构设计、逻辑结构设计、物理结构设计，数据的录入与各功能的查询测试，存储过程与触发器的设计及使用，数据库的维护管理等，完成题目要求的功能。

# 2 基本要求

用户有以下需求：建立读者档案；建立图书档案，建立书目索引；完成日常图书检索、借还工作，对读者档案、图书档案、借还系统的访问，必需进行身份验证。具体要求如下：

1． 读者档案

数据包括：借书证号、姓名、性别、出生日期、身份证号、单位、通讯地址、邮政编码、联系电话、办证日期、借阅范围（书库）、允许最多借书册数、借书期限、照片、职业等。

2． 图书档案

数据包括：书号、书名、作者、出版单位、出版日期、版次、单价、内容提要、分类号、索书号、藏书册数、每册图书馆藏注册号、所在书库、入库日期等。

3． 检索系统

能根据书号、书名、作者、出版单位、内容提要关键字、分类号、索书号、每册图书馆藏注册号等进行查询。

4． 借书系统

输入借书条后，能根据借书证号判断该读者可以借书的书库，借书是否超出最大允许借书册数，书库中是否还有该书可借。

5、还书系统

对过期未还图书进行罚款，对归还的图书能从借书登记表中取消，对丢失的图书进行登记。

# 3 系统需求分析

## 3.1需求分析阶段的目标与任务

### 3.1.1处理对象

读者档案：借书证号， 姓名， 性别， 出生日期， 身份证号， 单位， 通讯地址， 邮政编码， 联系电话， 办证日期，借阅等级， 借书册数， 借书期限照片， 职业

书籍档案：书号，书名，作者，出版单位，出版日期，版次，单价，内容提要，分类号，藏书册数，入库日期

书本档案：书号，索引号，书名，当前状态，借出日期，借书人

书架总揽：分类，编号，总数

书架信息：分类，编号，序号，书名，书号，索引号

借书记录表：借书条编号，借书证号，书名，书号，借出日期，归还日期，是否生效

### 3.1.2 处理功能及要求

1. 能够通过查询插入/删除书籍
2. .能够在加入书籍时根据数量自动创建书本档案对应条目，自动创建书架对应条目。
3. .能够在删除书籍时，自动判断是否有书被借出，自动删除书本档案与书架中对应的条目。

2.能够借书/还书

(1)通过书号自动选取没被借出的索引号

(2)通过书条自动完成书本归还

3.能够查询图书馆相关信息

(1)根据某一信息查找对应书籍及同属此书籍的书本的具体信息

(2)查找对应书架及书架内书本的具体信息

4.能够有一定的提示/帮助语句

(1)告诉用户相关的语句名字，参数，用法

## 3.2需求分析阶段成果

### 3.2.1 读者借/还书流程图

读者在进入图书馆后：若要借书，先通过终端查询书籍信息，再向系统发出申请，

若要还书，向系统提交书条号，若无违约则完成还书，

若有违约，则还需缴纳罚金（图3-1）

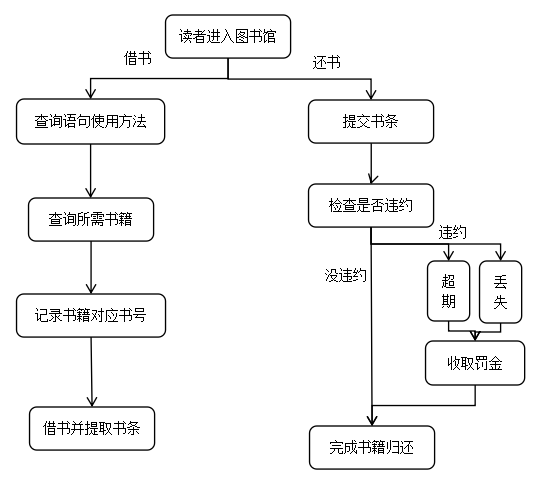


图3-1 读者借/还书流程图

### 3.3.2 新书籍增加流程图

在管理员向图书馆中增加了新的书籍后，信息会保存到书籍档案中，同时通过约束器的效果，会自动添加对应数量的书本到书本档案与书架信息中。（图3-2）

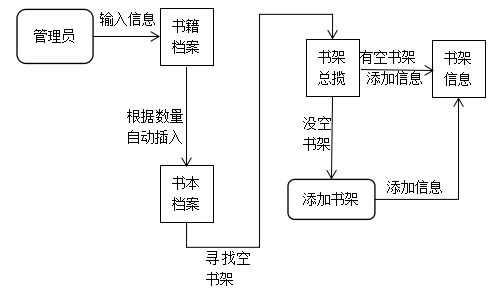


图3-2新书籍增加流程图

## 3.3 图书管理系统字典

### 3.3.1 数据字典

对于下表（表3-1）：

R（Reader）代表读者档案的数据

B（Books）代表书籍档案的数据

B1（Book）代表书本档案的数据

BC(BookCases) 代表书架的总揽信息的数据

BC1(BookCase) 代表书架详细信息的数据

LL（LendLog）代表借书记录的数据

表 3-1 数据字典

|  |  |  |  |
| --- | --- | --- | --- |
| R-1 | 借书证号 | Rno | int |
| R-2 | 姓名 | Rname | nvarchar(MAX) |
| R-3 | 性别 | Rsex | nvarchar(1) |
| R-4 | 出生日期 | Rbitrh | date |
| R-5 | 身份证号 | Rid | nvarchar(18) |
| R-6 | 单位 | Rworkadd | nvarchar(MAX) |
| R-7 | 通讯地址 | Radd | nvarchar(MAX) |
| R-8 | 邮政编码 | Rpos | nvarchar(6) |
| R-9 | 联系电话 | Rtel | nvarchar(11) |
| R-10 | 办证日期 | Rdate | smalldatetime |
| R-11 | 借阅等级 | Rrank | int |
| R-12 | 借书册数 | Rsum | int |
| R-13 | 借书期限 | Rlenddate | int |
| R-14 | 照片 | Rphoto | nvarchar(MAX) |
| R-15 | 职业 | Rwork | nvarchar(20) |
| B-1 | 书号 | Bno | int |
| B-2 | 书名 | Bname | nvarchar(MAX) |
| B-3 | 作者 | Baut | nvarchar(MAX) |
| B-4 | 出版单位 | Bco | nvarchar(50) |
| B-5 | 出版日期 | Bcodate | date |
| B-6 | 版次 | Bturn | int |
| B-7 | 单价 | Bprice | float |
| B-8 | 内容提要 | Bcontent | nvarchar(MAX) |
| B-9 | 分类号 | Bcalss | nvarchar(1) |
| B-10 | 藏书册数 | Bsum | int |
| B-11 | 入库日期 | Bindate | date |
| B1-1 | 书号 | Bno | int |
| B1-2 | 索引号 | Bid | int |
| B1-3 | 书名 | Bname | nvarchar(10) |
| B1-4 | 当前状态 | Bstate | bit |
| B1-5 | 借出日期 | Bdate | date |
| B1-6 | 借书人 | Rno | int |
| BC-1 | 书架分类 | BCclass | nvarchar(1) |
| BC-2 | 书架序号 | BCunder | int |
| BC-3 | 书本总数 | BCsum | int |
| BC1-1 | 书架分类 | BCclass | nvarchar(1) |
| BC1-2 | 书架序号 | BCunder | int |
| BC1-3 | 书序 | BCorder | int |
| BC1-4 | 书名 | Bname | nvarchar(MAX) |
| BC1-5 | 书号 | Bno | int |
| BC1-6 | 索引号 | Bid | int |
| LL-1 | 书条号 | Lno | int |
| LL-2 | 书号 | Bno | int |
| LL-3 | 索引号 | Bid | int |
| LL-4 | 借书证号 | Rno | int |
| LL-5 | 书名 | Bname | nvarchar(MAX) |
| LL-6 | 借书人姓名 | Rname | nvarchar(MAX) |
| LL-7 | 借出日期 | Ldate | date |
| LL-8 | 归还日期 | Rdate | date |
| LL-9 | 有效性 | Lstate | bit |

### 3.3.2 触发器字典

下标（表3-2）是图书管理系统中用到触发器的具体信息：

表 3-2 触发器字典

|  |  |  |
| --- | --- | --- |
| 触发器名 | 触发条件 | 主要实现目的 |
| BookCreate | 书名存储时 | 根据B插入项目中Bsum一值，在B1中自动创建 对应个数的子图书，索引号分别为1~Bsum |
| BooksDelete | 书名删除时 | 如果有书被借出，则不允许删除 |
| BookInsert | 书本存储时 | 存放入书本信息，如果书号存在，则依次相 加，并同时加入到书架中 如果书籍不存在，创建书籍信息 |
| BookDelete | 书本删除时 | 如果这是最后一本书，则删除书名 如果这本书被借出，删除卷回 并且在删除同时删除书架对应信息 |

### 3.3.3 存储过程字典

下表（表3-3）是图书管理系统中用到的存储过程的具体信息：

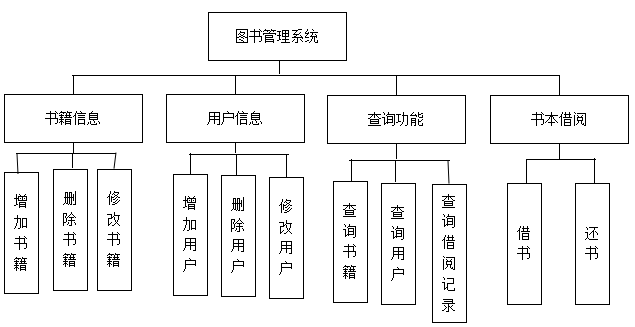
表3-3 存储过程字典

|  |  |  |  |
| --- | --- | --- | --- |
|  | 存储过程名 | 参数 | 主要实现目的 |
| PROC-1 | LendBook | Rno,Bno | 检测信用等级和借书上限 自动查找是否有剩余书籍 |
| PROC-2 | ReturnBook | Lno,state | 判断书籍状态，是否需要交罚金， 并完成对表格的更改/删除 |
| PROC-3 | FindAllBooks |  | 查找所有书名 |
| PROC-4 | FindBooks | @info | 按书号查找 |
| PROC-5 | FindBooks1 | @info | 按内容查找 |
| PROC-6 | FindBooks2 | @info | 按作者查找 |
| PROC-7 | FindBooks3 | @info | 按出版社查找 |
| PROC-8 | FindBooks4 | @info | 按分类查找 |
| PROC-9 | FindAllBook |  | 查找所有书本 |
| PROC-10 | FindBook | Bno | 按书号查找书本信息 |
| PROC-11 | FindBook1 | Bno,Bid | 按书号与序列号查找书本信息 |
| PROC-12 | FindUnreturnBook |  | 查找借出书本 |
| PROC-13 | FingOvertimeBook |  | 查找借出超时的书本 |
| PROC-14 | BookCase |  | 查找所有书架 |
| PROC-15 | BookCase1 |  | 按类别查找书架 |
| PROC-16 | BookCase2 |  | 查看对应书架的详细信息 |
| PROC-17 | LendLog |  | 查看所有借书记录 |
| PROC-18 | LendLog2 |  | 查看未归还借书记录 |
| PROC-19 | Help |  | 帮助索引页 |
| PROC-20 | Help1 |  | 帮助-查找书籍信息 |
| PROC-21 | Help2 |  | 帮助-查找书本信息 |
| PROC-22 | Help3 |  | 帮助-查找书架信息 |
| PROC-23 | Help4 |  | 帮助-查看借阅/归还书本 |
| PROC-24 | Help5 |  | 帮助-查看借书记录 |

# 4 概念结构设计

## 4.1 系统功能模块图

图书管理系统分为书籍信息，用户信息，查询，借阅等模块，具体如下图（图4-1）所示:

 图4-1 功能模块图

4.2 E-R图

图书管理系统E-R图如图4-2所示.

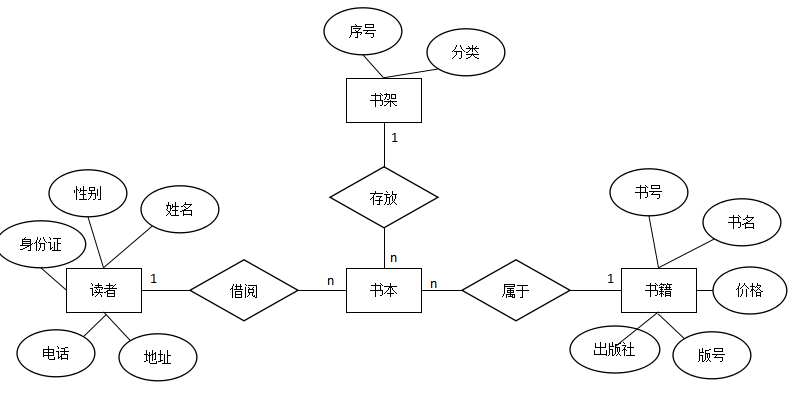


图 4-2 图书管理系统E-R图

## 4.3数据库逻辑结构设计

图书管理系统逻辑结构设计如下所示：

读者档案：Reader(Rno,Rname,Rsex,Rbirth,Rid,Rworkadd,Radd,

Rpos,Rtel,Rdate,Rrank,Rsum,Rlenddate,Rphoto)

书籍档案：Books(Bno,Bname,Baut,Bco,Bcodate,Bturn,Bprice,Bcontent,Bclass,Bsum,Bindate)

书本档案：Book(Bno,Bid,Bname,Bstate,Bdate,Rno)

书架总档案：BookCases(BCclass,BCunder,BCsum)

书架档案：BookCase(BCclass,BCsunder,BCorder,Bname,Bno,Bid)

借书记录：LendLog(Lno,Bno,Bid,Rno,Bname,Rname,Ldate,Rdate,Lstate)

# 数据库实施阶段

## 5.1 建立数据库

create database library

## 5.2 建立数据表

1.建立读者档案表

create table R

(

Rno int identity(1,1) primary key,--借书证号

Rname nvarchar(MAX) not null,--姓名

Rsex nvarchar(1) check(Rsex = '男' or Rsex = '女'),--性别

Rbirth date,--出生日期

Rid nvarchar(18) not null unique check(Rid like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0- 9|X]'),--身份证号

Rworkadd nvarchar(MAX),--单位

Radd nvarchar(MAX),--通讯地址

Rtel nvarchar(11) check(Rtel like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

Rpos nvarchar(6) check(Rpos like '[0-9][0-9][0-9][0-9][0-9][0-9]'),--邮政编码

Rdate date not null default(getdate()),--办证日期

Rrank int not null default(1),--借阅等级

Rsum int not null default(0),--允许最多借书册数

Rlenddate int not null default(20),--借书期限

Rphoto nvarchar(MAX),--照片

Rwork nvarchar(MAX),--职业

)

2.建立书籍档案表

create table B

(

Bno int identity(1,1) primary key,--书号

Bname nvarchar(MAX) not null,--书籍

Baut nvarchar(MAX),--作者

Bco nvarchar(50),--出版单位

Bcodate date,--出版日期

Bturn tinyint,--版次

Bprice float not null,--单价

Bcontent nvarchar(MAX),--内容提要

Bclass nvarchar(1) not null,--分类号

Bsum int not null,--藏书册数

Bindate date not null default(getdate())--入库日期

)

1. 建立书本档案表

create table B1

(

Bno int,--书号

Bid int,--索引号

Bname nvarchar(MAX) not null,--书籍

Bstate bit not null default(1),--当前状态

Bdate date,--借书人

Rno int,--借出日期

constraint pk\_B1 primary key (Bid,Bno),

constraint fk\_B1\_B foreign key (Rno) references R(Rno),

constraint fk2\_B1\_B foreign key (Bno) references B(Bno)

)

4. 建立书架总表

create table BC

(

BCclass nvarchar(1),

BCunder int,

BCsum int,

constraint pk\_BC primary key(BCclass,BCunder)

)

5.建立书架档案表

create table BC1

(

BCclass nvarchar(1),

BCunder int default(0),

BCorder int,

Bname nvarchar(MAX),

Bno int,

Bid int,

constraint pk\_BC1 primary key (BCorder,BCclass,BCunder)

)

6. 建立借阅记录表

create table LL

(

Lno int identity(1,1) primary key,

Bno int,

Bid int,

Rno int,

Bname nvarchar(MAX),

Rname nvarchar(MAX),

Ldate date default(getdate()),

Rdate date,

Lstate bit default(1),

)

## 5.3 建立触发器

1. 建立BoolCreat触发器,根据B插入项目中Bsum一值，在B1中自动创建

对应个数的子图书，索引号分别为1~Bsum

create trigger BookCreate

on B

after insert

as

begin

declare @no int,@sum int ,@now int

declare @name nvarchar(MAX)

select @no = Bno,@sum = Bsum from inserted;

select @now = count(Bid) from B1 where Bno=@no;

select @name = (select Bname from B where Bno=@no);

while @now < @sum

begin

insert into B1 (Bno,Bid,Bname,Bstate)values(@no,@now+1,@name,1)

select @now=@now+1

end

End

1. 建立BooksDelete触发器,如果有书被借出，则不允许删除

create trigger BooksDelete

on B

instead of delete

as

begin

declare @no1 int,@sum1 int

select @no1 = Bno from deleted

select @sum1 = COUNT(Bid) from B1 where Bno=@no1 and Bstate=0

if(@sum1>1)

print('有书正在被借出，不能删除！')

else

begin

delete from BC1 where Bno=@no1

delete from B1 where Bno=@no1

delete from B where Bno=@no1

update BC set BCsum = (select COUNT(BCorder) from BC1 where BC.BCclass = BC1.BCclass and BC.BCunder = BC1.BCunder)

end

End

1. 建立BookInsert 触发器,存放入书本信息，如果书号存在，则依次相加，并同时加入到书架中如果书籍不存在，创建书籍信息

create trigger BookInsert

on B1

instead of insert

as

begin

declare @no int,@id int,@check int

declare @sum int,@name nvarchar(MAX)

declare @class nvarchar(1),@order int

select @id = Bid from inserted

select @no = Bno from inserted

select @name =Bname from inserted

select @class = Bclass from B where Bno = @no

select @check = COUNT(Bno) from B1 where Bno=@no and Bid=@id

select @sum = COUNT(Bno) from B1 where Bno=@no

declare @flag int,@temp int,@BCsum int,@check1 int,@under int

begin

while(@check =1)

begin

select @id = @id+1

print('索引号已存在 自动后移')

select @check = COUNT(Bno) from B1 where Bno=@no and Bid=@id

end

update B set Bsum=Bsum+1 where Bno=@no

insert B1 (Bno,Bid,Bname,Bstate) values (@no,@id,@name,1)

select @flag = 0

select @BCsum = MAX(BCunder) from BC where BCclass = @class

if(@BCsum is NULL)

select @BCsum = 0

select @temp =1

while(@temp <= @BCsum and @flag = 0)

begin

select @check1 = BCsum from BC where BCclass =@class and BCunder =@temp

if(@check1 <20)

begin

select @flag =1

select @order = MAX(BCorder) from BC1 where BCclass = @class and BCunder = @temp

insert BC1 (BCclass,BCunder,BCorder,Bname,Bno,Bid)values (@class,@temp,@order+1,@name,@no,@id)

update BC1 set Bname = B.Bname from B where B.Bno = @no and Bid =@id

update BC set BCsum += 1 where BCclass = @class and BCunder = @temp

end

select @temp += 1

end

if(@flag = 0)

begin

insert BC (BCclass,BCunder,BCsum)values (@class,@BCsum+1,1)

select @BCsum = MAX(BCunder) from BC where BCclass = @class

insert BC1 (BCclass,BCunder,BCorder,Bname,Bno,Bid)values (@class,@BCsum,1,@name,@no,@id)

update BC1 set Bname = B.Bname from B where B.Bno = @no and Bid =@id

end

end

end

1. 建立BookDelete触发器,如果这是最后一本书，则删除书名,如果这本书被借出，删除卷回,并且在删除同时删除书架对应信息

create trigger BookDelete

on B1

instead of delete

as

begin

declare @no int,@id int,@check int

declare @sum int,@name nvarchar(MAX)

select @id = Bid from deleted

select @no = Bno from deleted

select @name = Bname from deleted

select @check = Bsum from B where Bno=@no

if(@check = 1)

begin

delete from BC1 where Bno=@no and Bid = @id

delete from B1 where Bno=@no and Bid = @id

delete from B where Bno=@no

end

else

begin

delete from BC1 where Bno=@no and Bid = @id

delete from B1 where Bno=@no and Bid = @id

update B set Bsum=Bsum-1 where Bno=@no

update BC set BCsum = (select COUNT(BCorder) from BC1 where BC.BCclass = BC1.BCclass and BC.BCunder = BC1.BCunder)

end

End

## 5.4 建立存储过程

1. LendBook,检测信用等级和借书上限,自动查找是否有剩余书籍

create proc LendBook

@Rno int,

@Bno int

as

begin

declare @sum int,@check int,@temp int,@Rrank int,@sum2 int,@num int

select @Rrank = Rrank from R where Rno = Rno

select @sum = COUNT(Bid) from B1 where Bno=@Bno

select @sum2 = Rsum from R where Rno = @Rno

select @temp = 1

select @check = Bstate from B1 where Bno=@Bno and Bid=@temp

if(@Rrank >0)

begin

if(@sum2 < 2)

begin

if(@sum>0)

begin

while(@check = 0)

begin

select @temp = @temp +1

select @check = Bstate from B1 where Bno=@Bno and Bid=@temp

end

update B1 set Bstate = 0 ,Rno = @Rno,Bdate = GETDATE()

where Bid = @temp and Bno = @Bno

insert LL (Bno,Bid,Rno) values (@Bno,@temp,@Rno)

update LL set Bname = (select Bname from B where Bno=@Bno)

update LL set Rname = (select Rname from R where Rno=@Rno)

update R set Rsum = Rsum+1 where Rno = @Rno

print('借书成功,你的借书条编号为：')

select @num = MAX(Lno) from LL

print cast(cast(@num as decimal(10)) as nvarchar(MAX))

end

else

print('此书没有剩余')

end

else

print('你借书过多，请归还后再借')

end

else

print('信用等级不足，不能借书')

End

1. ReturnBook,判断书籍状态，是否需要交罚金，并完成对表格的更改/删除

create proc ReturnBook

@Lno int,

@state bit

as

begin

declare @flag1 bit,@flag2 bit,@Bno int ,@Rno int,@nowdate date

declare @cost float,@date date,@Bid int,@day int,@price float

declare @LLstate bit

select @LLstate = Lstate from LL where Lno = @Lno

select @flag1 = 0,@flag2 = 0,@cost = 0.0

select @Bno = Bno , @Rno = Rno , @Bid =Bid ,@date = Ldate from LL where Lno = @Lno

select @nowdate = GETDATE()

select @day = DATEDIFF(DAY,@date,@nowdate)

select @price = Bprice from B where Bno = @Bno

if(@LLstate = 1)

BEGIN

update LL set Lstate = 0

update R set Rsum -= 1

update B1 set Bstate = 1,Bdate = null

update LL set Rdate = @nowdate

if(@day>30)

begin

select @cost = @cost+@day\*0.2

select @flag1 = 1

end

if(@state = 0)

begin

select @cost = @cost+@price\*1.5

select @flag2 = 1

end

if(@flag1 = 1)

update R set Rrank = 0 where Rno = @Rno

if(@flag2 = 1)

begin

update R set Rrank = 0 where Rno = @Rno

delete from B1 where Bno=@Bno and Bid =@Bid

end

if(@flag1 = 1 or @flag2 = 1)

begin

print '你需要交罚金(元):'

print cast(cast(@cost as decimal(10,2)) as nvarchar(MAX))

end

else

print '还书成功'

END

else

print '此书条已被归还过'

End

1. FindAllBooks,查找所有书籍

create proc FindAllBooks

as

select \* from B

1. FindBooks，按书号查找

create proc FindBooks

@Bno int

as

select \* from B where Bno=@Bno

1. FindBooks1，按内容查找

create proc FindBooks1

@info nvarchar(MAX)

as

select \* from B where Bcontent like ('%'+@info+'%')

1. FindBooks2，按作者查找

create proc FindBooks2

@info nvarchar(MAX)

as

select \* from B where Baut like ('%'+@info+'%')

1. FindBooks3，按出版社查找

create proc FindBooks3

@info nvarchar(MAX)

as

select \* from B where Bco like ('%'+@info+'%')

1. FindBooks4，按分类查找

create proc FindBooks4

@info nvarchar(MAX)

as

select \* from B where Bclass like ('%'+@info+'%')

1. FindAllBook，查找所有书本

create proc FindAllBook

as

select \* from B1

1. FindBook，按书号查找书本信息

create proc FindBook

@Bno int

as

select \* from B1 where Bno=@Bno

1. FingBook1，按书号与序列号查找书本信息

create proc FindBook1

@Bno int,

@Bid int

as

select \* from B1 where Bno=@Bno and Bid=@Bid

1. FindUnretrunBook，查找借出书本

create proc FindUnreturnBook

as

select \* from B1 where Bstate = 0

1. FindOvertimeBook，查找借出超时的书本

create proc FingOvertimeBook

as

select \* from B1 where Bstate = 0 and datediff(day,Bdate,GETDATE()) > 20

1. BookCase，查找所有书架

create proc BookCase

as

select \* from BC

1. BookCase1，按类别查找书架

create proc BookCase1

@class nvarchar(1)

as

select \* from BC where BCclass =@class

1. BookCase2，查看对应书架的详细信息

create proc BookCase2

@class nvarchar(1),

@under int

as

select \* from BC1 where BCclass =@class and BCunder = @under

1. LendLog，查看所有借书记录

create proc LendLog

as

select \* from LL

1. LendLog2，查看未归还借书记录

create proc LendLog2

as

select Lno,Bno,Bid,Rno,Bname,Rname,Ldate from LL where Lstate = 1

1. Help，帮助索引页

create proc Help

as

begin

print'欢迎来到图书馆：

请问想查询：

1.查找书籍信息 Help1

2.查找书本信息 Help2

3.查找书架信息 Help3

4.查看借阅/归还书本

5.查看借书记录

'

end

1. Help1，帮助-查找书籍信息

create proc Help1

as

begin

print '1.查找书籍信息'

print ' 查找所有 FindAllBooks

按书号查找 FindBools 书号

按内容查找 FindBools2 内容

按作者查找 FindBools3 作者

按出版社查找 FindBools4 出版社

按分类查找 FindBools5 分类'

print '以下是分类列表'

print ' D 政治、法律

E 军事

F 经济

G 文化、科学、教育、体育

H 语言、文字

I 文学

'

end

1. Help2，帮助-查找书本信息

create Proc Help2

as

begin

print '2.查找书本信息

查找所有 FindAllBook

按书号查找 FindBook 书号

按书号与序列号查找 FindBook2 书号，序列号

查找已借出书本 FindUnreturnBook

查找借出超时书本 FindOvertimeBook

'

end

1. Help3，帮助-查找书架信息

create proc Help3

as

begin

print '3.查找书架信息

查找所有书架 BookCase

按书架类别查找 BookCase1 类别

查找对应书架详细信息 BookCase2 类别 书架号

'

print '以下是分类列表'

print ' D 政治、法律

E 军事

F 经济

G 文化、科学、教育、体育

H 语言、文字

I 文学

'

End

1. Help4，帮助-查看借阅/归还书本

create proc Help4

as

begin

print '借书 LendBook 你的读者号 书号

还书 Return Book 书条编号，书本状态（0为书本丢失或损坏，1为正常）'

end

1. Help5，帮助-查看借书记录

create proc Help5

as

begin

print '查看所有借书记录 LendBook

查看还为归还的借书记录 LendBook2'

End

# 系统调试和测试

## 6.1 数据存储

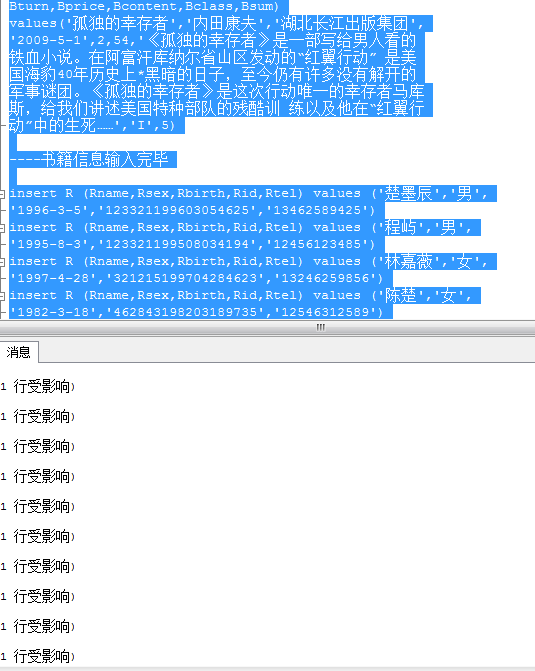


图6-1 存储结果

由图6-1可知，对于一次输入的多个信息，是可以成功加入的，另外通过触发器也成功的做到了书籍信息的输入（图6-2），书本的自动插入（图6-3），书架的自动管理（图6-4）



图 6-2 书籍信息表



图6-3 书本信息表

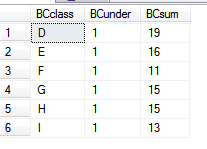


图6-4 书架信息表

## 6.2借书/还书功能

### 6.2.1 借书

如图6-5所示，借书能成功进行

如果借书过多结果如图6-6所示

如果信用等级不足，结果如图6-7所示

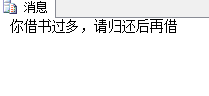
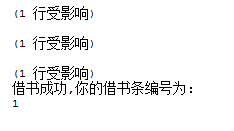


图6-5 借书 图6-6 借书过多

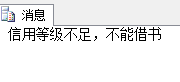


图6-7 信用等级不足

### 6.2.2 还书

如果没有违约，结果如图6-8所示

如果违约，结果如图6-9所示

如果书条已经还过，结果如图6-10所示

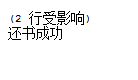


图6-8

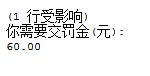


图6-9

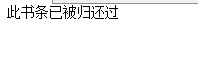


图6-10

## 6.3 查询功能

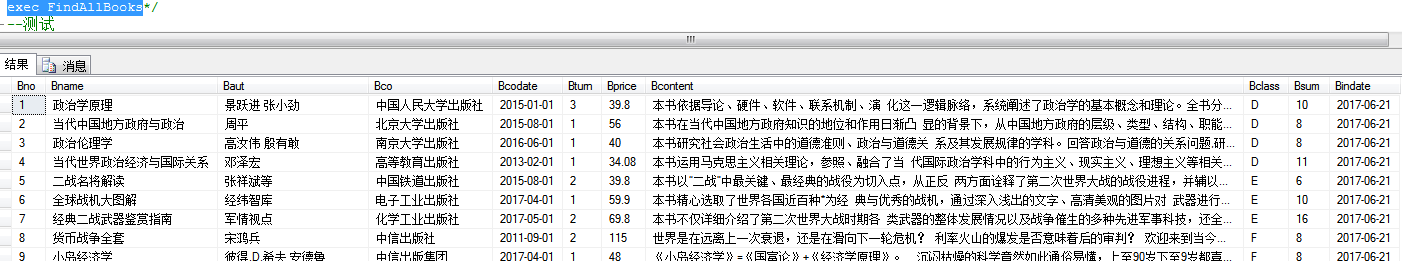
1. 书籍查询（图6-11）

图6-11 查询数据

2.书本查询（图6-12）

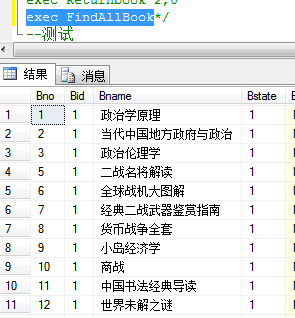


图6-12 查询书本

1. 书架查询（图6-13）

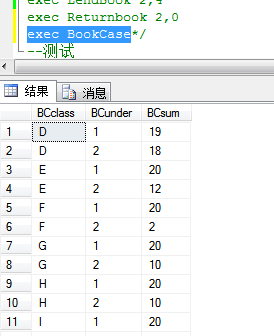


图6-13 查询书架

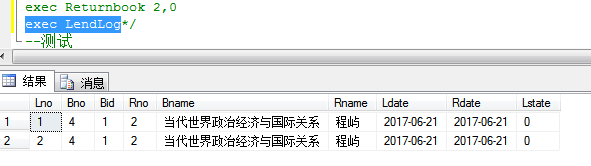
1. 借阅记录查询（图6-14）

图6-14 查询记录

5.帮助查询（图6-15）

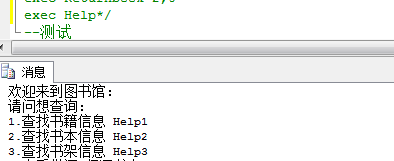


图6-15 查询帮助

# 课程设计体会和总结