



图书借阅系统

详细设计报告

学校：北京交通大学

学院：计算机与信息技术学院

班级：计科1403班

学号：14281066

姓名：苗伟

2016年12月11日

目录

**[1](#_Toc469847650)** [引言 3](#_Toc469847650)

[**1.1** 需求分析 3](#_Toc469847651)

[**1.2** 系统目标 3](#_Toc469847652)

[**1.2.2**、系统中应包含的信息 4](#_Toc469847653)

[**1.2.3**、完整性约束 4](#_Toc469847654)

[**1.2.4**、与系统的交互 5](#_Toc469847655)

[**1.3** 数据流字典和数据字典 5](#_Toc469847656)

[数据结构 12](#_Toc469847657)

[数据流 12](#_Toc469847658)

[数据存储 12](#_Toc469847659)

[处理过程 13](#_Toc469847660)

[**2** 概要设计 14](#_Toc469847661)

[**2.1** 系统 **ER** 图 14](#_Toc469847662)

[**2.2** 系统功能结构图 15](#_Toc469847663)

[**3** 程序系统的结构 15](#_Toc469847664)

[**3.1** 数据库设计 15](#_Toc469847665)

[**3.2** 数据库关系图 19](#_Toc469847666)

[**3.3** 数据库连接 19](#_Toc469847667)

[**4** 系统实现 20](#_Toc469847668)

[**4.1** 程序总体概述 20](#_Toc469847669)

[**4.2** 详细设计和运行界面 20](#_Toc469847670)

[**4.2.1**、认证 20](#_Toc469847671)

[**4.2.2**、读者登录 23](#_Toc469847672)

[**4.2.3**、管理员登录 27](#_Toc469847673)

[**5** 软件使用说明 37](#_Toc469847674)

[**5.1** 软件开发和运行环境**.** 37](#_Toc469847675)

[**6** 实验总结、问题与解决 37](#_Toc469847676)

# **1** 引言

## **1.1** 需求分析

图书借阅系统需要满足来自图书馆管理员、借阅者两方面人员的需求。图书馆管理员可以给借阅者分配编号，注册借阅者信息，图书入库，办理借书证，完成图书的借阅和归还，获取图书和读者的相关信息，维护图书和读者信息，逾期不还处理，图书遗失和借书证挂失管理等操作；图书借阅者可以查询自己的个人信息，搜索查看图书馆的书籍，查看自己的借阅历史和未还图书。

图书借阅者需要根据本人借书证号和密码登录系统，可以进行本人借书情况的查询。本功能实现对馆内图书信息的查询。可以通过图书名称、作者的名字、出版社、类别来查询相应的图书信息，此查询可模糊查询也可精确查询。所以不但满足了图书借阅者的要求，还保护了图书借阅者的个人隐私。

图书馆管理员有对图书借阅者借书和还书记录进行操作的权限，所以需对管理员登陆本模块进行更多的考虑。在此模块中，图书馆工作人员可以为图书借阅者加入借书记录或是还书记录，并生成相应的报表给用户查看和确认。图书馆管理员可以浏览、查询、统计、添加借阅图书的基本信息及删除图书的基本信息，当删除某条图书借阅的基本信息记录时，应实现对该图书借阅者借阅记录的级联删除。本功能实现对图书信息的管理和统计。图书馆管理员功能的信息量大，数据安全性和保密性要求最高。

## **1.2** 系统目标

#### 1.2.1 系统目标是使管理员和读者完成以下操作：

1. 对系统的用户身份进行认证
2. 图书入库
3. 办理借书证
4. 完成图书借阅和归还
5. 获取图书和读者的相关信息
6. 维护图书和读者信息
7. 逾期不还的处理
8. 图书遗失和借书证挂失的处理

### **1.2.2**、系统中应包含的信息

1. 使用该系统的管理员和读者的密码和身份
2. 每本图书的信息如下：

图书编号，书名，作者,出版社，单价,类别，版次，入库数，出库数，丢失数等

其中，类别包括计算机技术,自然科学,医学,农林，建筑，科普和通信

1. 每个读者的信息：

读者的编号，姓名，性别，出生日期，证件名称，证件号码，会员等 级，办证日期，联系电话，手机号码等。其中，不同级别会员的最长出借天数，最多借书次数和会费不同

1. 读者的借阅、归还、过期和挂失信息

### **1.2.3**、完整性约束

1. 图书编号和读者编号唯一；
2. 读者借阅的图书一定是系统中提供的图书；
3. 读者遗失的图书一定是本人借阅的图书；
4. 已遗失的图书不能再被借阅；
5. 遗失图书数量不能超过馆藏图书数量；
6. 已挂失的借书证不能再借书；
7. 出借天数和最多借书册书由会员级别决定。

### **1.2.4**、与系统的交互

1. 认证

验证用户身份，输入ID和密码，随后的交互取决于用户的身份；

1. 如果用户是读者，可进行如下交互　 查询图书的各种信息

按书名、图书类别、作者、出版社查，可提供选择和输入两种方式：查询人的借阅信息列出本人所有借阅历史信息（六个月内，一年内）和当前借阅信息（指未还的书籍）；

1. 如果用户是管理员，可进行如下交互：

1)查询图书的各种信息

查询馆藏图书和借阅图书，以列表显示，点击每本图书显示该图书详细信 息；

2）查询读者的各种信息

提供输入姓名和编号两种方式

3)图书入库

登记图书入库信息

4)办理借书证

给读者分配一个编号，登记读者信息

5)图书借阅和归还

不同级别会员的出借天数和最多借阅册书不同

6)逾期未还处理

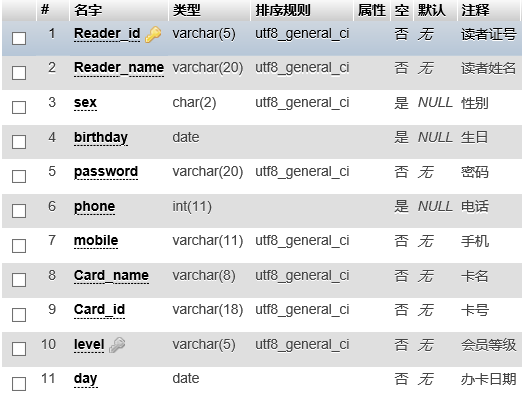
列出未还的图书清单和读者清单

7)挂失处理

包括借书证挂失和图示遗失处理，借书证挂失在挂失数据表中处理，图书遗失在借阅表和图书表中处理。

## **1.3** 数据流字典和数据字典

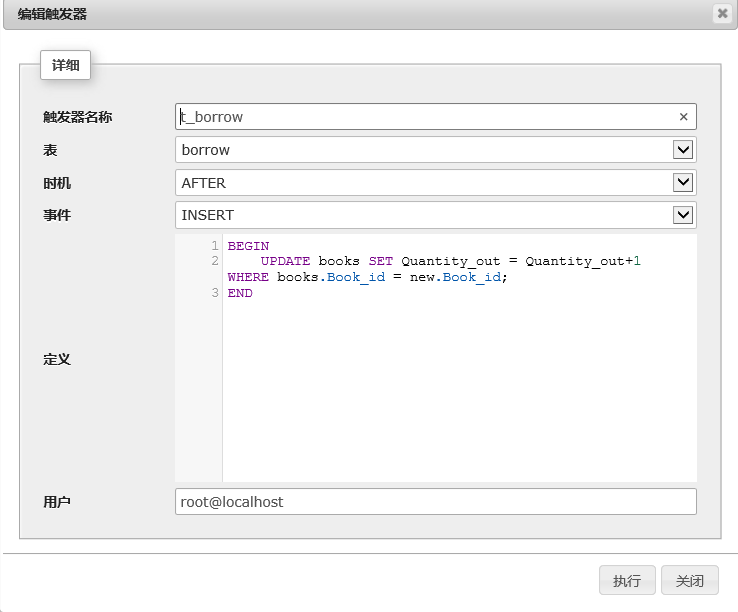
Readers表结构

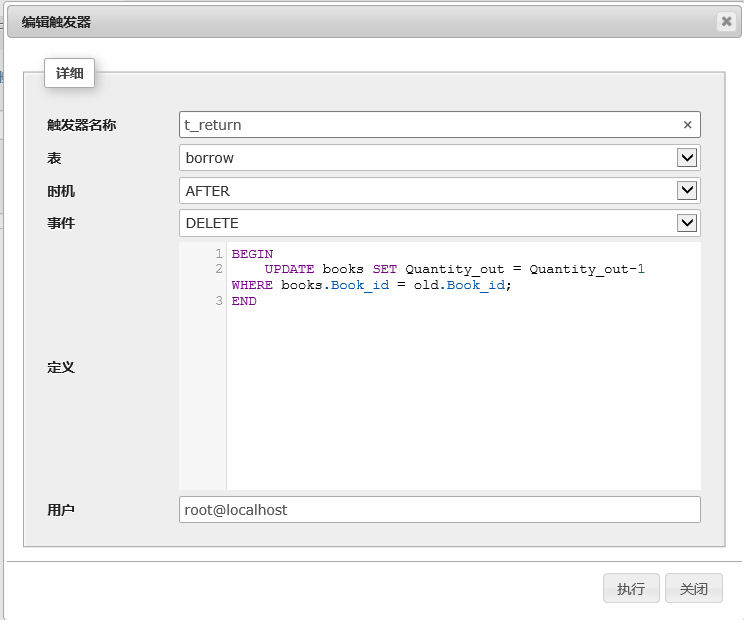


Borrow表结构

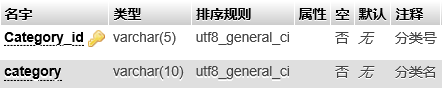


Borrow表的触发器





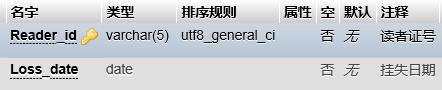
b\_category表



Books表



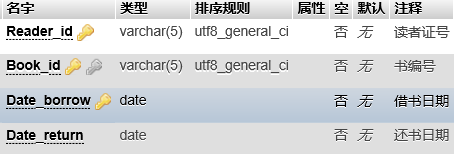
Loss\_reporting表



Member\_level表



History\_borr表



存储过程borr



存储过程ret

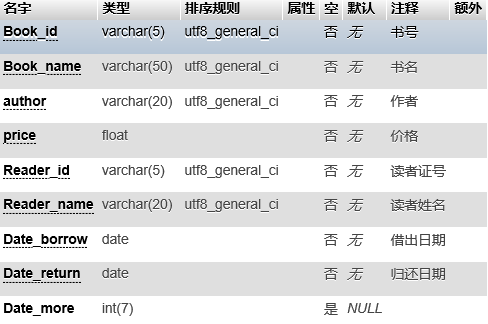


视图

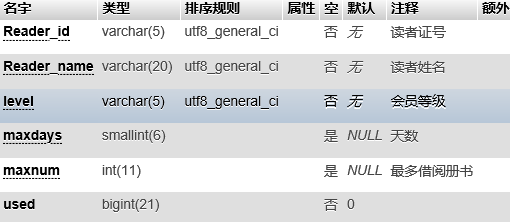
视图Book\_info



视图Borrow\_info



视图Card\_info



视图History



### 数据结构

|  |  |  |
| --- | --- | --- |
| 名称 | 含义 | 组成 |
| 读者信息 | 定义了读者的基本信息 | 读者编号、读者姓名、性别、生日、手机号、固话号、证件名称、证件号码、级别、注册日期、密码 |
| 书本信息 | 定义了书本的基本信息 | 书本编号、书本名字、作者、出版社、书本类型、价格、入库时间、入库数、借出量、丢失量 |
| 借出状态 | 定义了读者借出的全部信息，以及所有的历史记录 | 读者编号、书本编号、借书日期、归还日期、是否丢失、是否归还 |
| 用户级别 | 反应了读者不同的级别 | 级别编号、级别名称、最大借书天数、最大借书量、会员费 |
| 书本类型 | 反应了书本的不同类型 | 类型编号、类型时间 |
| 丢失状态 | 反应了用户借书证的丢失状态 | 读者编号、丢失时间 |

### 数据流

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据流 | 说明 | 来源 | 去向 | 组成 |
| 书籍查询 | 根据条件进行组合查询 | 读者或者管理员 | 查询结果 |  |
| 借阅图书 | 进行图书借阅 | 管理员、书籍查询结果 | 得到图书编号和读者编号 |  |
| 借阅登记 | 将借阅情况进行登记 | 借阅图书信息 | 登记借阅 |  |
| 读者借书情况 | 列出读者的借阅信息 | 读者、管理员 | 借阅详细信息 |  |
| 读者信息查询 | 列出读者的相关信息 | 管理员 | 读者详细信息 |  |
| 读者资料维护 | 更改读者相关信息 | 管理员 | 读者资料变更 |  |
| 借书证挂失 | 读者借书证丢失 | 管理员 | 读者挂失 |  |
| 读者注册 | 按读者信息进行数据录入 | 管理员 | 读者注册 |  |
| 图书入库 | 将书籍录入入库信息 | 管理员 | 书籍入库 |  |
| 图书归还 | 读者还书 | 管理员 | 借书表和书籍表数据变更 |  |
| 图书挂失 | 书籍丢失 | 管理员 | 书籍表数据变更 |  |

### 数据存储

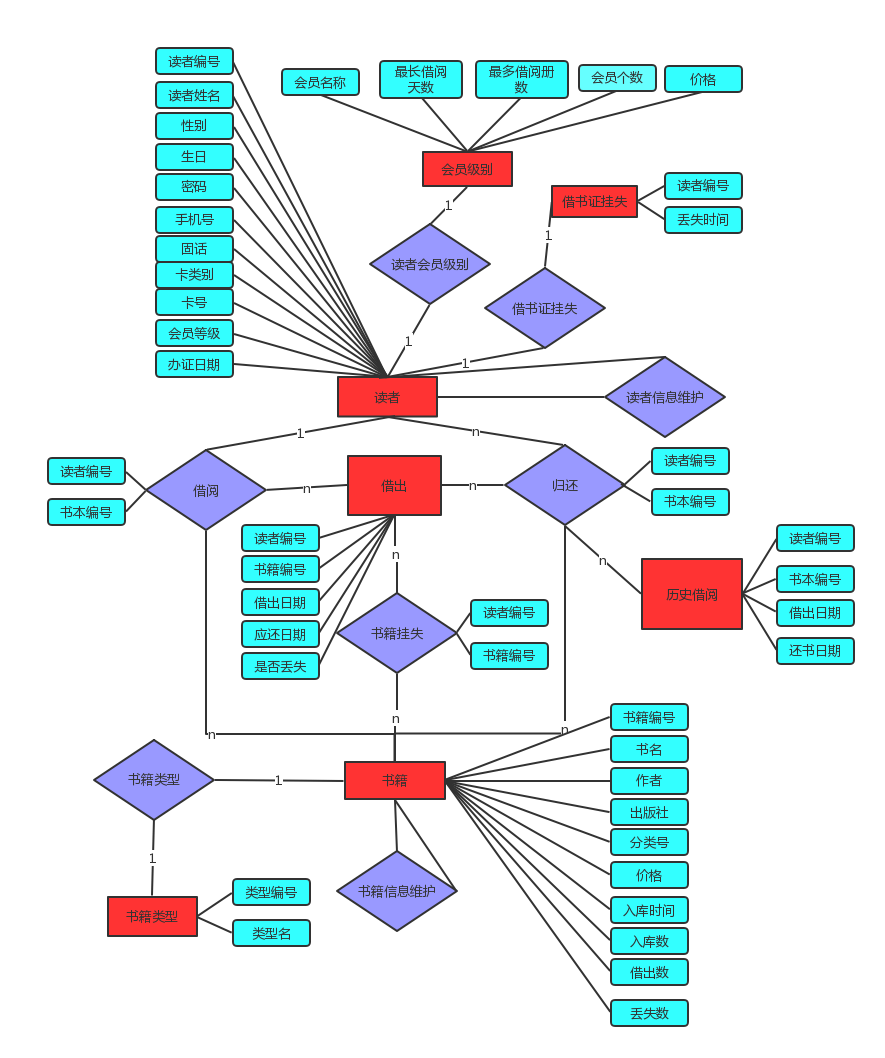
|  |  |  |  |
| --- | --- | --- | --- |
| 数据存储 | 说明 | 输入数据流 | 输出数据流 |
| 读者信息表 | 读者相关信息 | 读者注册 | 读者信息查询 |
| 书本信息表 | 书籍相关信息 | 图书入库 | 书籍查询 |
| 借出表 | 读者借书情况记录 | 借阅登记 | 读者借书情况 |
| 书本类型表 | 存数书籍的类型 | - | - |
| 用户级别表 | 存数读者级别的分类 | - | - |
| 丢失登记表 | 存储读者借书证丢失的相关信息 | 借书证挂失 | - |

### 处理过程

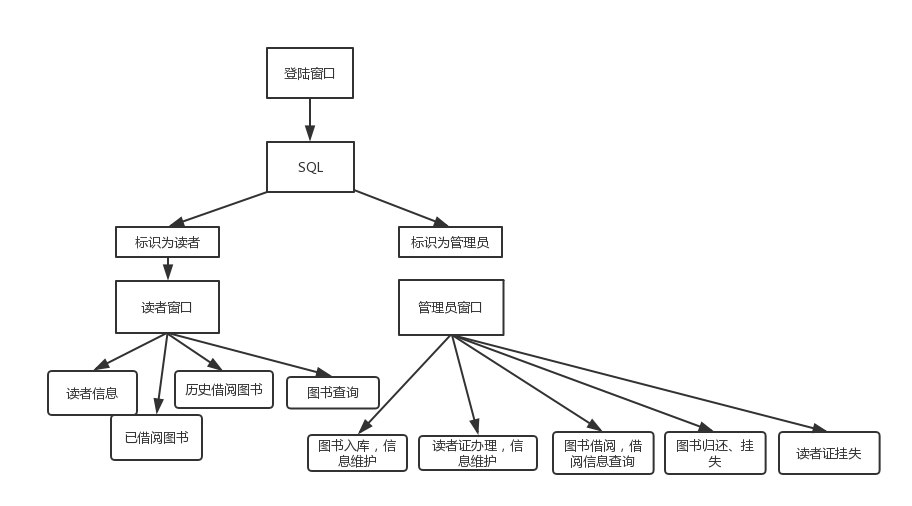
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 处理过程 | 说明 | 输入 | 输出 | 处理 |
| 图书借阅 | 将读者借阅的信息进行登记 | 书籍编号、读者编号 | 书籍借阅信息 | 首先选择读者借阅的书籍，输入读者编号，进行借阅。 |
| 图书归还 | 读者借出的书籍进行归还操作 | 书籍编号、读者编号 | 书籍归还成功 | 输入读者编号、书籍编号点击归还即可 |
| 图书挂失 | 读者借阅图书丢失 | 书籍编号、读者编号 | 挂失成功 | 输入书籍编号、读者编号点击挂失即可 |
| 图书入库 | 录入新书信息 | 书籍信息 | 录入成功 | 输入书籍的详细信息，点击添加即可 |
| 读者注册 | 读者注册 | 读者详细信息 | 读者详细信息 | 输入读者的详细信息，点击注册即可 |
| 借书证挂失 | 读者借书证丢失 | 读者编号、挂失时间 | 挂失成功 | 输入读者编号，点击挂失 |

# **2** 概要设计

## **2.1** 系统 **ER** 图



## **2.2** 系统功能结构图



# **3** 程序系统的结构

## **3.1** 数据库设计

1、读者

readers（Reader\_id,Reader\_name, sex,birthday,password,phone, mobile, card\_name,card\_id,level,day）

读者( 读者编号，姓名，性别，出生日期，密码，电话，手机，证件名称，证件编号,会员级别，办证日期）

readers表

CREATE TABLE `readers` (

`Reader\_id` varchar(5) NOT NULL,

`Reader\_name` varchar(20) DEFAULT NULL,

`sex` char(2) DEFAULT NULL,

`birthday` varchar(255) DEFAULT NULL,

`password` varchar(255) NOT NULL,

`phone` varchar(255) DEFAULT NULL,

`mobile` varchar(11) DEFAULT NULL,

`card\_name` varchar(8) DEFAULT NULL,

`card\_id` varchar(18) DEFAULT NULL,

`level` varchar(6) DEFAULT NULL,

`day` date DEFAULT NULL,

PRIMARY KEY (`readerid`),

KEY `levelfkd` (`level`), CONSTRAINT `readers\_ibfk\_1` FOREIGN KEY (`level`) REFERENCES

`member\_level` (`level`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

2、图书

books（Book\_id,Book\_name, author,publishing, Category\_id ,price, Date\_in, Quantity\_in, Quantity\_out, Quantity\_loss）

图书( 图书编号，书名，作者，出版社，类别，单价，入库日期，入库数量，出借数量，遗失数量）

Book表

CREATE TABLE `books` (

`Book\_id` varchar(5) NOT NULL,

`Book\_name` varchar(50) DEFAULT NULL,

`author` varchar(20) DEFAULT NULL,

`publishing` varchar(20) DEFAULT NULL,

`Category\_id` varchar(5) DEFAULT NULL,

`price` double(10,2) DEFAULT NULL,

`Datein` date DEFAULT NULL,

`Quantityin` int(255) DEFAULT NULL,

`Quantityout` int(255) DEFAULT NULL,

`Quantityloss` int(255) DEFAULT NULL,

PRIMARY KEY (`bookid`),

KEY `categoryidfkd` (`Category\_id`), CONSTRAINT `categoryidfkd` FOREIGN KEY (`Category\_id`) REFERENCES

`b\_category` (`Category\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

1. 图书类别

b\_category（Category\_id, category)

图书类别( 类别编号，类别名称）

b\_category表

CREATE TABLE `b\_category` (

`Category\_id` varchar(5) NOT NULL,

`category` varchar(10) DEFAULT NULL,

PRIMARY KEY (`Category\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

1. 挂失

loss\_reporting（Reader\_id, loss\_date) 挂失( 读者编号，挂失日期）

loss\_porting表

CREATE TABLE `loss\_reporting` (

`Reader\_id` varchar(5) CHARACTER SET utf8 NOT NULL,

`lossdate` date DEFAULT NULL,

PRIMARY KEY (`Reader\_id `),

KEY `Reader\_id` (`Reader\_id`),

CONSTRAINT `lossreporting\_ibfk\_3` FOREIGN KEY (`Reader\_id`) REFERENCES

`readers` (`Reader\_id`) ON DELETE CASCADE ON UPDATE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=gbk;

5、借阅

borrow（Reader\_id, Book\_id, Date\_borrow, Date\_return, loss)

借阅( 读者编号，图书编号，出借日期，应还日期，遗失）

Borrow表

CREATE TABLE `borrow` (

`readerid` varchar(5) CHARACTER SET utf8 NOT NULL, `bookid` varchar(5) CHARACTER SET utf8 NOT NULL,

`dateborrow` date DEFAULT NULL,

`datereturn` date DEFAULT NULL,

`dateshouldreturn` date DEFAULT NULL,

`loss` int(1) DEFAULT NULL,

`returnstate` int(1) DEFAULT NULL, `leavedays` bigint(1) DEFAULT NULL,

PRIMARY KEY (`Reader\_id`),

KEY `readerid` (`readerid`),

KEY `bookid` (`bookid`), CONSTRAINT `bookid` FOREIGN KEY (`bookid`) REFERENCES `books`

(`bookid`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `readerid` FOREIGN KEY (`readerid`) REFERENCES `readers`

(`readerid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=27 DEFAULT CHARSET=gbk;

6、会员级别

Member\_level（level, days, num,numbers, fee)

会员级别( 会员级别，最长出借天数，最多借书册书，会员个数，会费）

Memberlevel表

CREATE TABLE `memberlevel` (

`level` varchar(6) NOT NULL,

`days` smallint(10) DEFAULT NULL,

`num` int(11) DEFAULT NULL,

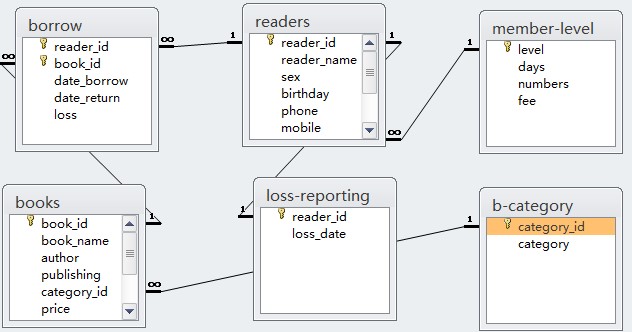
`numbers` int(10) DEFAULT NULL,

`fee` smallint(10) DEFAULT NULL,

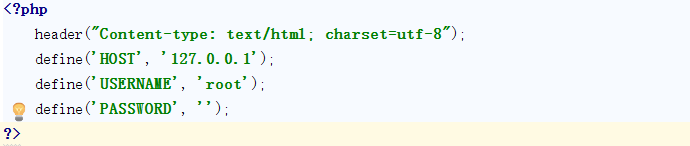
PRIMARY KEY (`level`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## **3.2** 数据库关系图



## **3.3** 数据库连接





# **4** 系统实现

## **4.1** 程序总体概述

图书借阅者需要根据本人借书证号和密码登录系统，可以进行本人借书情况的查询。本功能实现对馆内图书信息的查询。可以通过图书名称、作者的名字、出版社、类别和图书编号来查询相应的图书信息。

图书馆管理员有对图书借阅者借书和还书记录进行操作的权限，所以需对管理员登陆本模块进行更多的考虑。在此模块中，图书馆工作人员可以为图书借阅者加入借书记录或是还书记录，并生成相应的报表给用户查看和确认。图书馆管理员可以浏览、查询、统计、添加借阅图书的基本信息及删除图书的基本信息，当删除某条图书借阅的基本信息记录时，应实现对该图书借阅者借阅记录的级联删除。本功能实现对图书信息的管理和统计。图书馆管理员功能的信息量大，数据安全性和保密性要求最高。

## **4.2** 详细设计和运行界面

### **4.2.1**、认证

用户登录，输入登录名和密码，普通读者和管理员登录名不同，在后台进行判别用户的身份。查询用户的语句是：

**function** collect\_data($con,$name,$password){  
  
 $sql = **"select** *\** **from admin WHERE name='**$name**'and password = '**$password**'"**;  
 $result = mysqli\_query($con,$sql);  
  
 $num= mysqli\_num\_rows($result);  
 **return** $num;  
}  
*//查询是否是读者***function** collect\_data2($con,$name,$password)  
{  
 $sql = **"select Reader\_id,password from readers WHERE Reader\_id='**$name**'and password = '**$password**'"**;  
 $result = mysqli\_query($con,$sql);  
  
 $num= mysqli\_num\_rows($result);  
 **return** $num;  
}

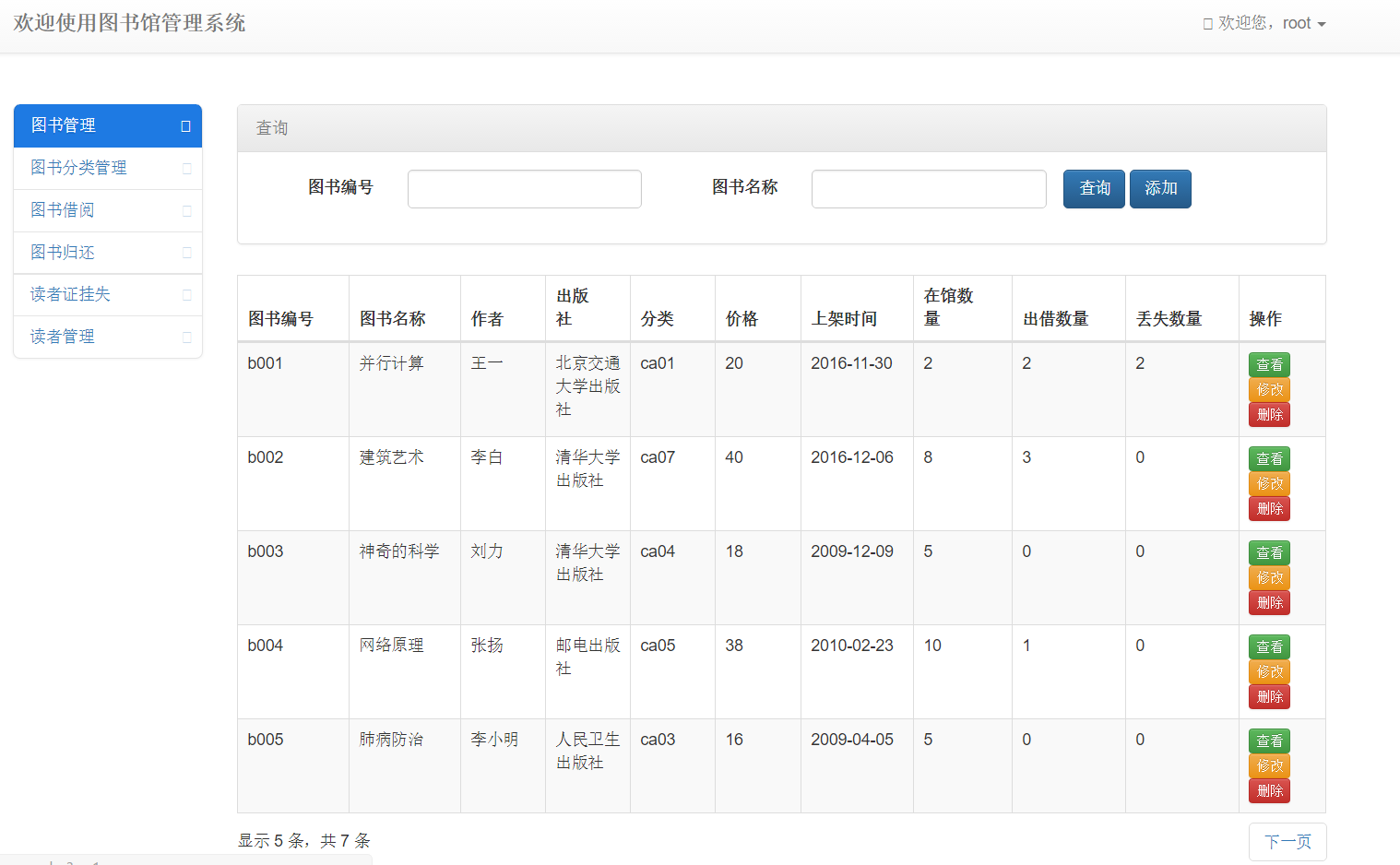


以读者身份登录界面：



 以管理员身份登录主界面





可以看到两个主界面的上菜单栏和左菜单栏是不同的

### **4.2.2**、读者登录

##### **4.2.2.1**、查询图书的各种信息

①根据书名查询

②根据书号查询

③根据作者查询

④根据出版社查询

⑤根据图书类别查询



结果：

 查询代码：

$page = $\_GET[**'p'**];  
$page\_size = 5;  
$Book\_name = $\_GET[**'query\_bname'**];  
$category = $\_GET[**'query\_cname'**];  
$author = $\_GET[**'query\_author'**];  
$publishing = $\_GET[**'query\_publish'**];  
$sql = **"select** *\** **from book\_info WHERE Book\_name LIKE '%**$Book\_name**%'and category LIKE '%**$category**%' AND author LIKE '%**$author**%' AND publishing LIKE '%**$publishing**%'order BY Book\_id LIMIT "**.($page-1)\*$page\_size .**",**$page\_size**"**;  
$query = mysqli\_query($con,$sql);  
$total\_sql = **"select** *\** **from book\_info WHERE Book\_name LIKE '%**$Book\_name**%'and category LIKE '%**$category**%' AND author LIKE '%**$author**%' AND publishing LIKE '%**$publishing**%'order BY Book\_id "**;  
$result=mysqli\_query($con,$total\_sql);  
$total = mysqli\_num\_rows($result);  
$total\_page = ceil($total/$page\_size);

##### **4.2.2.2**、查询本人的借阅信息



查询代码：

$page = $\_GET[**'p'**];  
$page\_size = 5;  
$Reader\_id = $\_SESSION[**'reader\_id'**];  
$Book\_id = $\_GET[**'query\_bno'**];  
$Book\_name = $\_GET[**'query\_bname'**];  
$sql = **"select** *\** **from borrow\_info WHERE Reader\_id = '**$Reader\_id**'AND Book\_id LIKE '%**$Book\_id**%' AND Book\_name LIKE '%**$Book\_name**%' ORDER BY Date\_borrow DESC LIMIT "**.($page-1)\*$page\_size .**",**$page\_size**"**;  
$query = mysqli\_query($con,$sql);  
$total\_sql = **"select** *\** **from borrow\_info WHERE Reader\_id = '**$Reader\_id**'AND Book\_id LIKE '%**$Book\_id**%' AND Book\_name LIKE '%**$Book\_name**%' ORDER BY Date\_borrow DESC "**;  
$result=mysqli\_query($con,$total\_sql);  
$total = mysqli\_num\_rows($result);  
$total\_page = ceil($total/$page\_size);

读者可以根据查自己的借阅历史看到自己的归还状态以及剩余天数等状态

处理图书，用户还可以查看自己的历史借阅信息

##### **4.2.2.3**、所有图书查看



代码部分：

$page = $\_GET[**'p'**];  
$page\_size = 5;  
$sql = **"select** *\** **from book\_info order BY Book\_id LIMIT "**.($page-1)\*$page\_size .**",**$page\_size**"**;  
$query = mysqli\_query($con,$sql);  
$total\_sql = **"select** *\** **from book\_info order BY Book\_id "**;  
$result=mysqli\_query($con,$total\_sql);  
$total = mysqli\_num\_rows($result);  
$total\_page = ceil($total/$page\_size);

##### **4.2.2.4**、图书详细信息和读者信息



查询语句：

**function** *showDetail*(Book\_id,Book\_name,category,author,publishing,price,Quantity\_in,Quantity\_left) {;  
 **$**(**"#detail\_bno"**).**val**(Book\_id);  
 **$**(**"#detail\_bname"**).**val**(Book\_name);  
 **$**(**"#detail\_tid"**).**val**(category);  
 **$**(**"#detail\_author"**).**val**(author);  
 **$**(**"#detail\_publishing"**).**val**(publishing);  
 **$**(**"#detail\_price"**).**val**(price);  
 **$**(**"#detail\_total"**).**val**(Quantity\_in);  
 **$**(**"#detail\_remain"**).**val**(Quantity\_left);  
 **$**(**'#modal\_detail'**).**modal**(**'show'**);  
}

### **4.2.3**、管理员登录

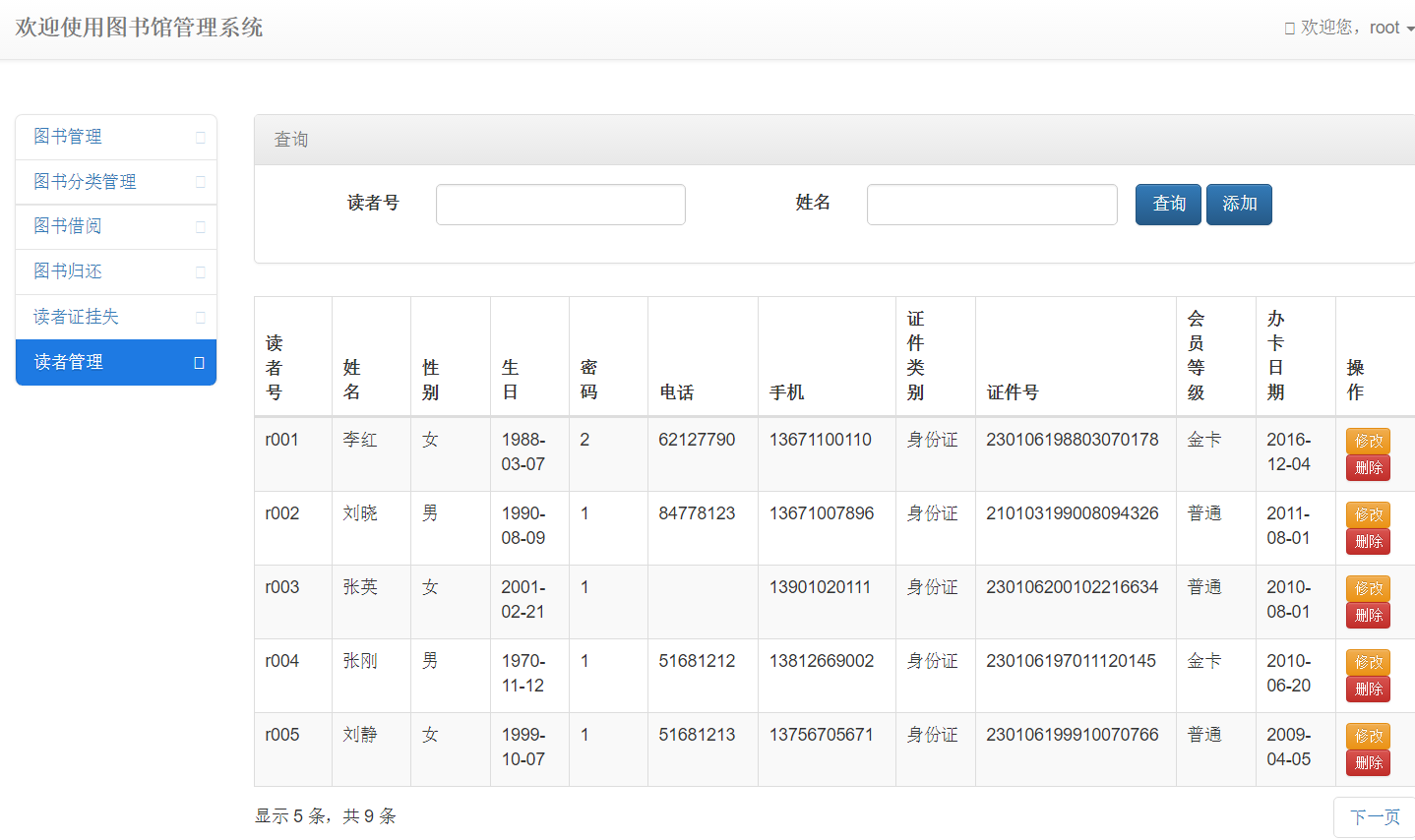
##### **4.2.3.1**、查询及维护图书的各种信息



查询代码：

$page = $\_GET[**'p'**];  
$page\_size = 5;  
$Book\_id = $\_GET[**'Book\_id'**];  
$Book\_name = $\_GET[**'Book\_name'**];  
$sql = **"select** *\** **from books WHERE Book\_id LIKE '%**$Book\_id**%' and Book\_name like'%**$Book\_name**%'order by Book\_id LIMIT "**.($page-1)\*$page\_size .**",**$page\_size**"**;  
$query = mysqli\_query($con,$sql);  
$total\_sql = **"select** *\** **from books WHERE Book\_id LIKE '%**$Book\_id**%' and Book\_name like'%**$Book\_name**%'order by Book\_id"**;  
$result=mysqli\_query($con,$total\_sql);  
$total = mysqli\_num\_rows($result);  
$total\_page = ceil($total/$page\_size);

##### **4.2.3.2**、查询及维护读者的各种信息

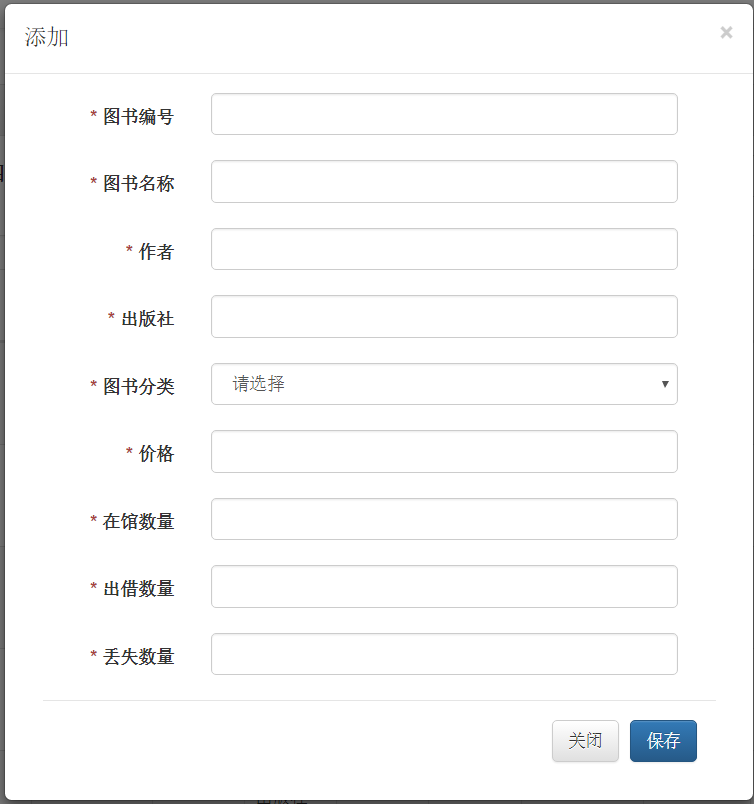


查询代码：

$page = $\_GET[**'p'**];  
$page\_size = 5;  
$sql = **"select** *\** **from readers order by Reader\_id LIMIT "**.($page-1)\*$page\_size .**",**$page\_size**"**;  
$query = mysqli\_query($con,$sql);  
$total\_sql = **"select** *\** **from readers order by Reader\_id"**;  
$result=mysqli\_query($con,$total\_sql);  
$total = mysqli\_num\_rows($result);  
$total\_page = ceil($total/$page\_size);

**4.2.3.3**、图书入库

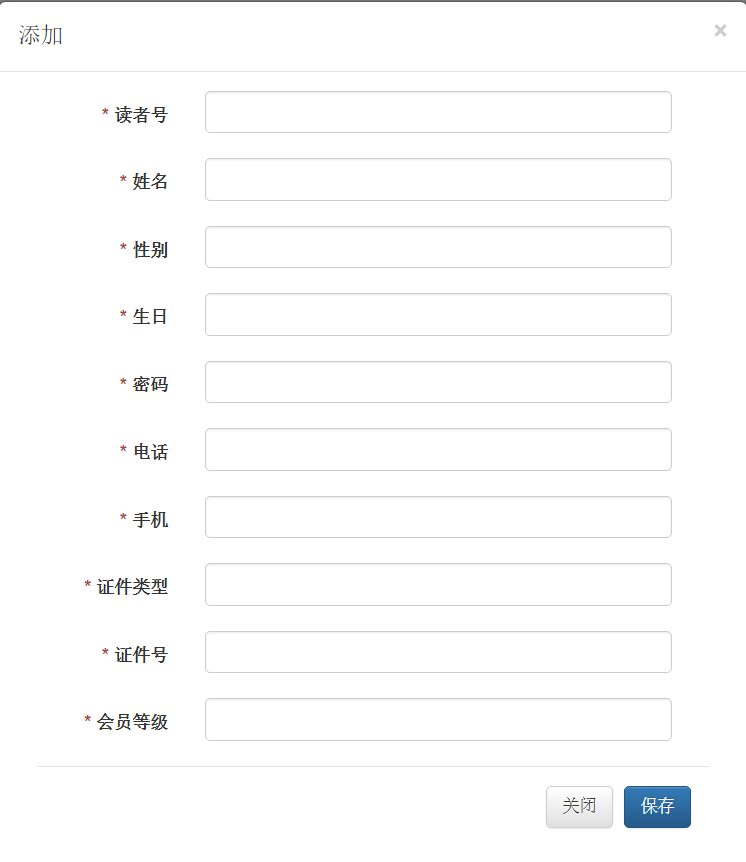
图书入库时要注意书号不能重复，在程序中有判断，若重复的话那么入库会

失败，如下图： 

代码部分：

**<?php  
 require\_once**(**'../connect.php'**);  
 *//把传递过来的信息入库,在入库之前对所有的信息进行校验。* **if**(!(**isset**($\_POST[**'Book\_name'**])&&(!**empty**($\_POST[**'Book\_name'**])))){  
 **echo "<script>alert('标题不能为空');window.location.href='article.add.php';</script>"**;  
 }  
 $Book\_id=$\_POST[**'Book\_id'**];  
 $Book\_name=$\_POST[**'Book\_name'**];  
 $author=$\_POST[**'author'**];  
 $publishing=$\_POST[**'publishing'**];  
 $Category\_id=$\_POST[**'Category\_id'**];  
 $price=$\_POST[**'price'**];  
 $date=date(**"Y-m-d"**,time());  
 $Quantity\_in=$\_POST[**'Quantity\_in'**];  
 $Quantity\_out=$\_POST[**'Quantity\_out'**];  
 $Quantity\_loss=$\_POST[**'Quantity\_loss'**];  
 $insertsql = **"insert into books(Book\_id,Book\_name,author,publishing,Category\_id,price,Date\_in,Quantity\_in,Quantity\_out,Quantity\_loss) values('**$Book\_id**', '**$Book\_name**','**$author**', '**$publishing**', '**$Category\_id**', '**$price**','**$date**','**$Quantity\_in**','**$Quantity\_out**','**$Quantity\_loss**')"**;  
**if**(mysqli\_query($con,$insertsql)){  
**echo "<script>alert('添加图书成功');window.location.href='book.manage.php?p=1';</script>"**;  
  
}**else**{  
 **echo "<script>alert('添加图书失败');window.location.href='book.manage.php?p=1';</script>"**;  
  
}  
**?>**

##### **4.2.3.4**、办理借书证



部分代码如下：

**<?php  
require\_once**(**'../connect.php'**);  
*//把传递过来的信息入库,在入库之前对所有的信息进行校验。***if**(!(**isset**($\_POST[**'add\_rname'**])&&(!**empty**($\_POST[**'add\_rname'**])))){  
 **echo "<script>alert('读者名不能为空');window.location.href='reader.manage.php?p=1';</script>"**;  
}  
$Reader\_id = $\_POST[**'add\_rno'**];  
$Reader\_name = $\_POST[**'add\_rname'**];  
$sex = $\_POST[**'add\_sex'**];  
$birthday = $\_POST[**'add\_birthday'**];  
$password = $\_POST[**'add\_password'**];  
$phone = $\_POST[**'add\_phone'**];  
$mobile = $\_POST[**'add\_mobile'**];  
$Card\_name = $\_POST[**'add\_card\_name'**];  
$Card\_id = $\_POST[**'add\_card\_id'**];  
$level = $\_POST[**'add\_level'**];  
$day = date(**"Y-m-d"**,time());  
$insertsql = **"insert into readers (Reader\_id,Reader\_name,sex,birthday,password,phone,mobile,Card\_name,Card\_id,level,day) values('**$Reader\_id**', '**$Reader\_name**','**$sex**', '**$birthday**', '**$password**', '**$phone**','**$mobile**','**$Card\_name**','**$Card\_id**','**$level**','**$day**')"**;  
**if**(mysqli\_query($con,$insertsql)){  
 **echo "<script>alert('添加读者成功');window.location.href='reader.manage.php?p=1';</script>"**;  
  
}**else**{  
 **echo "<script>alert('添加读者失败');window.location.href='reader.manage.php?p=1';</script>"**;  
  
}  
**?>**

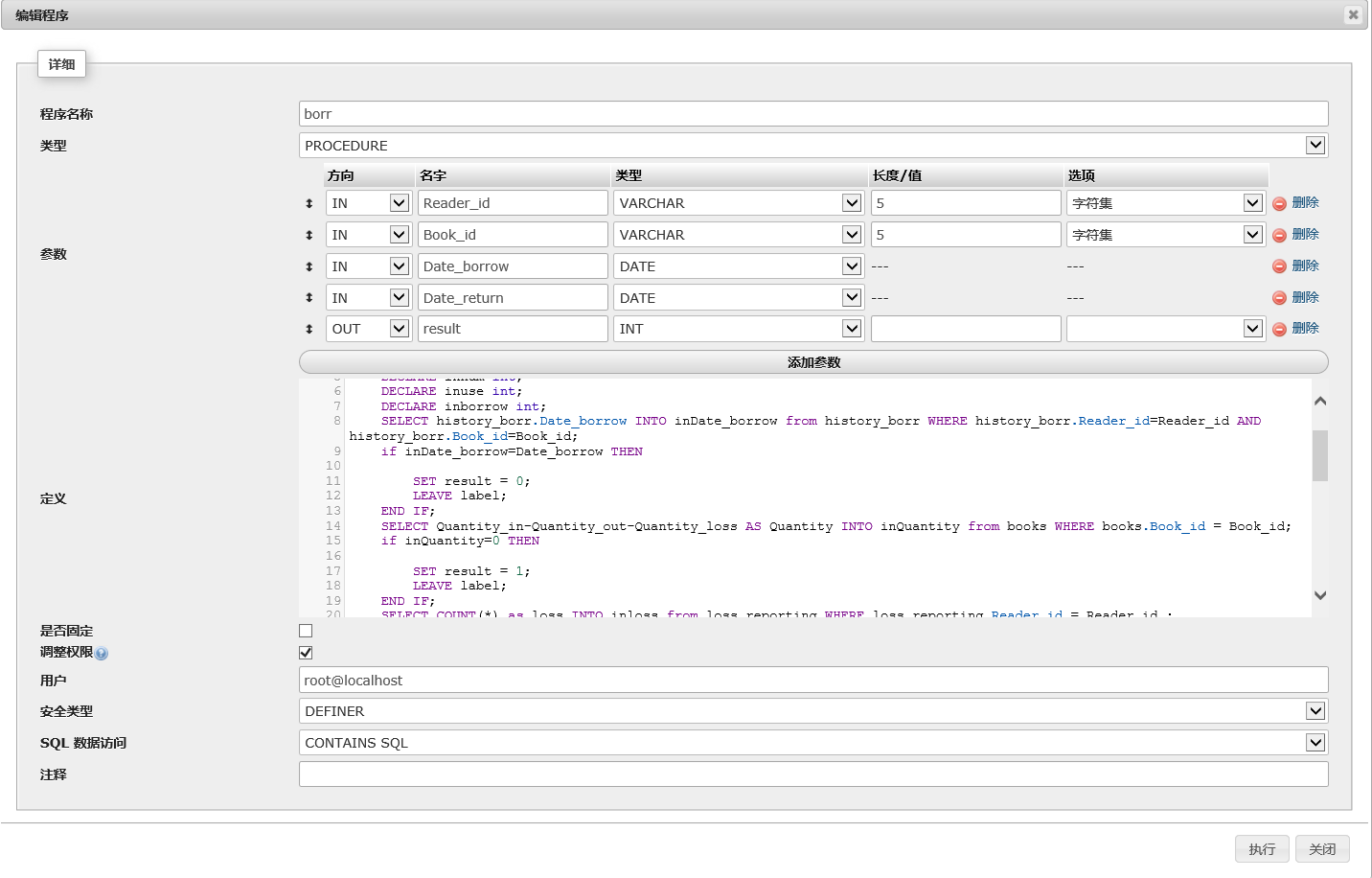
###### 4.2.3.5、图书借阅



点击借阅：

借阅后，borrow表中对应借阅信息，图书表中借出的书籍加1。

借阅过程使用了存储过程



部分代码如下：

**<?php***/\*\*  
 \* Created by PhpStorm.  
 \* User: will  
 \* Date: 2016/12/1  
 \* Time: 20:38  
 \*/***require\_once "../connect.php"**;  
$Reader\_id = $\_POST[**'borrow\_rno'**];  
$Book\_id = $\_POST[**'borrow\_bno'**];  
**if**(!strcmp($Book\_id,**""**)||!strcmp($Reader\_id,**""**))  
{  
 **echo "<script>alert('请输入图书编号和读者证号');window.location.href='book.borrow.php?p=1';</script>"**;  
  
}  
$sql = **"select** *\** **from readers WHERE Reader\_id = '**$Reader\_id**'"**;  
$result = mysqli\_query($con,$sql);  
$num = mysqli\_num\_rows($result);  
**if**($num == 0)  
{  
 **echo "<script>alert('该读者证号不存在');window.location.href='book.borrow.php?p=1';</script>"**;  
}  
$sql = **"select** *\** **from books WHERE Book\_id = '**$Book\_id**'"**;  
$result = mysqli\_query($con,$sql);  
$num = mysqli\_num\_rows($result);  
**if**($num == 0)  
{  
 **echo "<script>alert('该图书不存在');window.location.href='book.borrow.php?p=1';</script>"**;  
}  
  
$sql = **"select level from readers WHERE Reader\_id='**$Reader\_id**'"**;  
$query = mysqli\_query($con,$sql);  
$row =mysqli\_fetch\_assoc($query);  
$date\_more = date(**"Y-m-d"**,time());  
  
**if**(!strcmp($row[**'level'**],**"金卡"**))  
{  
 $date\_more = date(**'Y-m-d'**,strtotime(**'+90 day'**,strtotime(date(**"Y-m-d"**,time()))));  
}**else if**(!strcmp($row[**'level'**],**"银卡"**))  
{  
 $date\_more = date(**'Y-m-d'**,strtotime(**'+60 day'**,strtotime($date\_more)));  
}**else if**(!strcmp($row[**'level'**],**"普通"**)){  
 $date\_more = date(**'Y-m-d'**,strtotime(**'+30 day'**,strtotime($date\_more)));  
}  
  
  
$date=date(**"Y-m-d"**,time());  
$result = **"result"**;  
$sql = **"set @**$result **= -1"**;  
mysqli\_query($con,$sql);  
$sql = **"CALL borr('**$Reader\_id**','**$Book\_id**','**$date**','**$date\_more**',@**$result**)"**;  
mysqli\_query($con,$sql);  
$sql = **"Select @**$result**"**;  
$query = mysqli\_query($con,$sql);  
$row = mysqli\_fetch\_assoc($query);  
**if**($row[**'@result'**] == 0)  
{  
 **echo "<script>alert('借书失败,此读者今天已经借过此书');window.location.href='book.borrow.php?p=1';</script>"**;  
}  
**else if**($row[**'@result'**] == 1)  
{  
 **echo "<script>alert('借书失败,此书没有余量');window.location.href='book.borrow.php?p=1';</script>"**;  
}**else if**($row[**'@result'**] == 2)  
{  
 **echo "<script>alert('借书失败,该读者读者证丢失');window.location.href='book.borrow.php?p=1';</script>"**;  
}**else if**($row[**'@result'**]==3)  
{  
 **echo "<script>alert('借书失败,该读者借书数达到上限');window.location.href='book.borrow.php?p=1';</script>"**;  
}**else if**($row[**'@result'**]==4)  
{  
 **echo "<script>alert('借书失败,该读者借过此书还未还');window.location.href='book.borrow.php?p=1';</script>"**;  
} **else**{  
 **echo "<script>alert('借书成功');window.location.href='book.borrow.php?p=1';</script>"**;  
  
}  
**?>**

**4.2.3.6**、逾期未还处理和挂失



可以看到超期天数，说明此书逾期多少天没还，如果用户要挂失的话，输入读者编号，书籍编号，图书丢失数加1。

##### **4.2.3.7**、图书下架

 点击删除，假设我点击b005书的下架，那么在图书表以及包含的借阅表里这本书的信息将被级联删除.

##### **4.2.3.8**、用户信息查询



①通过读者编号进行模糊查询

②通过读者姓名进行模糊查询

可以看到我在读者姓名里输入了”刘”，那么将姓刘的读者都被查出来了。





##### **4.2.3.9**、挂失和补办借书卡



我点击挂失按钮，那么r001的读者借书卡将被挂失，然后r001将不会借阅。

在r001点击解除挂失，那么r001将从挂失表删除。

挂失部分代码：

**require\_once "../connect.php"**;  
$Reader\_id = $\_POST[**'add\_id'**];  
$loss\_date = $\_POST[**'add\_tdate'**];  
$sql = **"insert into loss\_reporting(Reader\_id,Loss\_date)VALUE('**$Reader\_id**','**$loss\_date**')"**;  
**if**(mysqli\_query($con,$sql)){  
 **echo "<script>alert('挂失成功');window.location.href='loss\_reporting.php?p=1';</script>"**;  
  
}**else**{  
 **echo "<script>alert('挂失失败');window.location.href='loss\_reporting.php?p=1';</script>"**;  
}

解除挂失部分代码：

**require\_once**(**'../connect.php'**);  
$Reader\_id = $\_POST[**'delete\_id'**];  
$delsql = **"delete from loss\_reporting WHERE Reader\_id='**$Reader\_id**'"**;  
**if**(mysqli\_query($con,$delsql)){  
 **echo "<script>alert('解除挂失成功');window.location.href='loss\_reporting.php?p=1';</script>"**;  
}**else**{  
 **echo "<script>alert('解除挂失失败');window.location.href='loss\_reporting.php?p=1';</script>"**;  
}

# **5** 软件使用说明

## **5.1** 软件开发和运行环境**.**

编译器：PhpStorm

操作系统：windows、linux或其他（数据库服务器需要windows系统）

服务程序：mysql服务，Apache服务

浏览器：支持所有主流浏览器，如IE7+、chrome、firefox等。

# **6** 实验总结、问题与解决

本次数据库课程设计对我的帮助很大，之前有自学过Mysql数据库，也动手写过文章管理系统，对数据库操作也有了一些初步的认识，主要使用增、删、改、查，对其他数据库本身的功能使用的很少。我又在暑假期间自学了一段时间PHP web开发，就借着数据库课程设计这个机会来从初步学习数据库与web开发，自己开发界面，从前端到后端的所有内容，其中加入了一些JavaScript来美化界面。本次课程设计从数据库连接到框架学习使用，再到页面布局都是本人边学习边做出来的，界面上做的还算简洁美观，底层也有很多问题有待改进。开发期间问题遇到了很多，比如sql语句的拼接，少加了一个空格符，反反复复改了一整天才查到这个低级错误，再者增删改查功能的封装操作，这对以后无论是学习还是工作都有非常大的帮助。