

# 实验报告

---

小组成员：

叶鑫茹：[redacted]

邓睿君：[redacted]

## 一、实验题目

图书馆图书管理查询系统

### 1. 组织机构



## 2. 机构职能

机构名称	主要任务	主要职能	下设部门
办公室	集图书馆行政、党务、人事、财务、后勤保障、安全保卫于一体的综合性职能部门	1.处理图书馆的各项日常运行文书工。2.财务工作：协助馆领导做好馆的预决算和小额行政经费的管理支付和日常报销等。3.人事工作：协助馆领导做好包括新员工招聘、岗位聘任、考勤、福利、职称等相关工作；做好临时工、学生助管等管理工作。	
采编部	承担着图书馆文献建设和发展任务，负责图书馆各类文献资源的收集、整理和编目工作	1.根据图书馆文献资源建设发展规划和经费年度预算，制订各类型文献资源采访计划，并有效地组织实施。2.根据图书馆文献资源建设的需要，系统地收集各学科文献资源的目录信息，按照图书馆采访原则，对各类型文献的进行有效的评估与严格的选择，及时地补充馆藏文献资源3.按照图书馆文献资源采访管理流程，对入藏图书馆文献资源进行登到验收、财务结算和资产管理等工作。4.对入藏图书馆的中西文图书进行分类编目、主题标引、加工典藏等工作。5.负责捐赠书刊的接收与转赠，负责同国内外学术机构的书刊交换工作。	
数据管理部	为图书馆文献信息服务提供信息化环境支撑和技术保障；负责数据图书馆的前沿研究、项目规划和建设，承担图书馆或科研机构的数据管理相关项目。	1.图书馆服务器设备、有线网络、软件平台、专用设备和系统、中心机房的管理和维护；2.图书馆信息系统安全管理；3.图书馆应用系统开发和维护；4.图书馆IT设备与信息系统采购；5.数据图书馆相关项目的管理和实施。	

机构名称	主要任务	主要职能	下设部门
文科馆	提供文科馆中外文书刊的流通、阅览管理与服务。	1.负责馆内中、外文图书的流通、阅览、清点工作。2.负责馆内中、外文期刊的阅览管理工作。3.负责校园卡的借书功能开通、借书证和临时阅览证的办理，办理离校手续。办理赔书。按照规定接待校外读者利用图书馆。4.负责合理组织馆藏，对本部门各借书处、阅览室及辅助书库进行图书调配工作。5.负责本馆书刊的整理、修补、提存。	下设：文科综合书库、文学艺术书库、文科过刊库、政经法阅览室、报刊阅览室、参考阅览室。负责人文社科类中文图书和中、外文期刊的阅览管理工作。
理科馆	提供理科馆中外文书刊的流通、阅览管理与服务。		下设：理科基础学科阅览室、理科应用学科阅览室、复旦人著作陈列室、理科书库、外国教材中心阅览室、理科报刊阅览室、学位论文阅览室、理科密集书库等。
医科馆	提供医科中外文图书的流通、阅览管理与服务。		下设：咨询台、期刊工作室、医科馆办公室等。管理中文图书区、外文图文区、教学参考书区、上医人文库、参考阅览区、人文艺术阅览区、密集库、视听区、24小时无人值守图书馆等。

### 3. 需求分析

## 二、开发环境

操作系统: windows10(代码编写,web 服务器),ubuntu 20(运行mysql服务器)

数据库管理软件: Navicat Premium 15

编程语言: python,sql(mysql),html,css,javascript

(B/S结构) web框架: django, bootstrap3

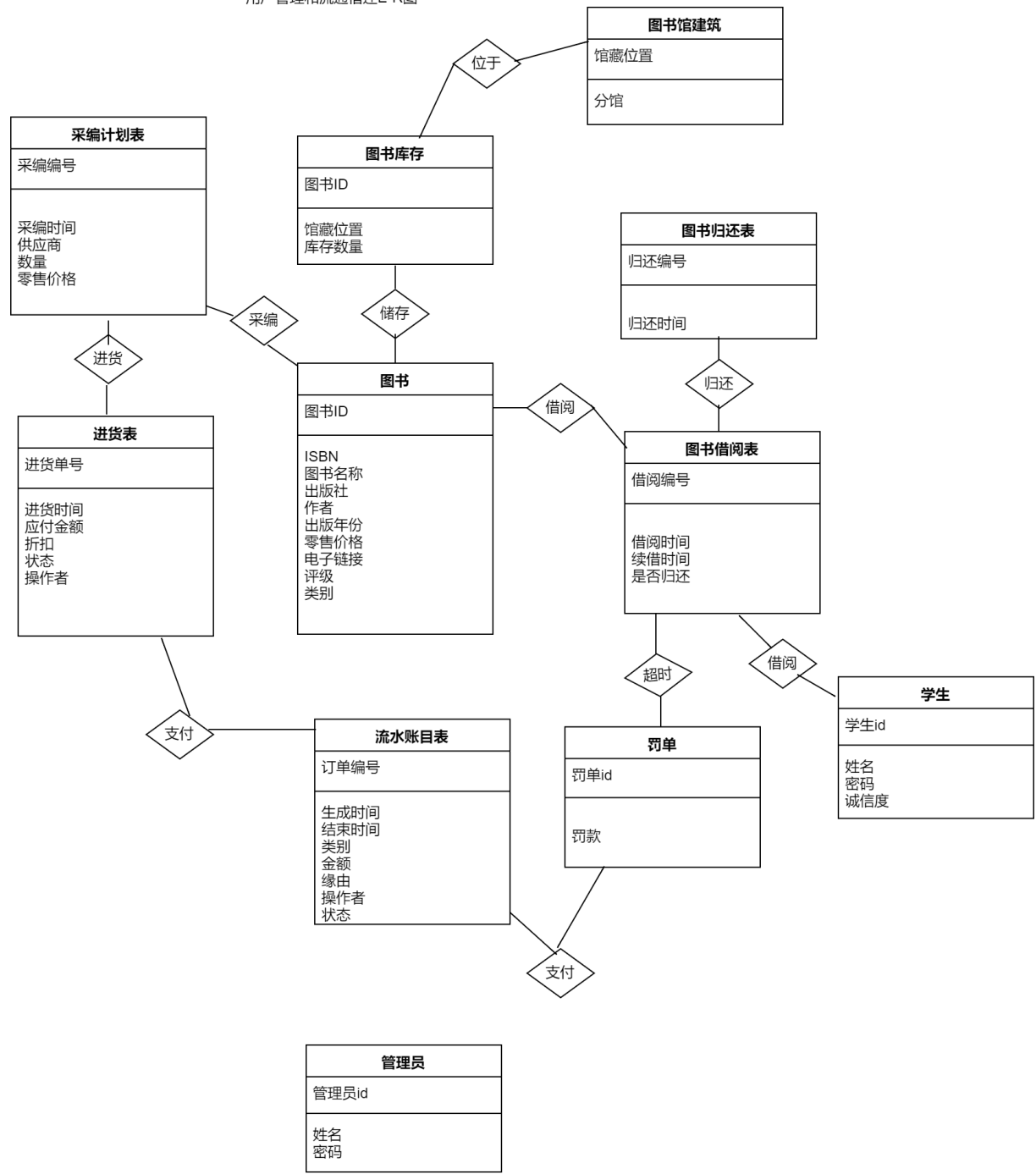
python版本: python3.6以上

## 三、数据库设计

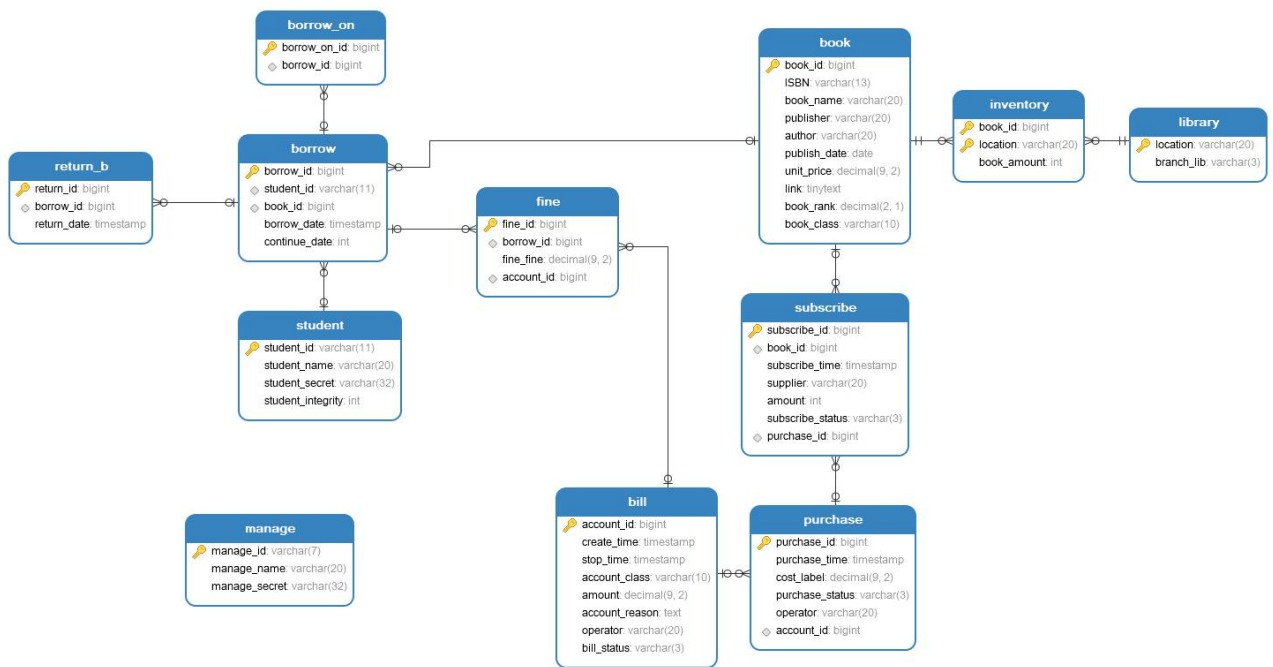
### 1. 设计图

#### 1.1 E-R 图建表前设计

图书管理系统  
用户管理和流通借还E-R图



1.2 E-R图建表后设计



## 2. 主要表格及说明

### 2.1 图书(book)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
图书ID	bigint	自增	book_id
ISBN	varchar(13)		ISBN
图书名称	varchar(20)	not null	book_name
出版社	varchar(20)		publisher
作者	varchar(20)		author
出版年份	date		publish_date
零售价格	decimal(9,2)		unit_price
电子链接	tinytext		link
评级	decimal(2,1) [0,10]		book_rank
类别	varchar(10)	文学艺术 政治经济 军事法律 哲学宗教 数学物理 医学药学	book_class

## 2.2 进货表(purchase)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
进货单号	bigint	自增	purchase_id
进货时间	timestamp	默认当前时间	purchase_time
应付金额	decimal(9,2)	not null	cost_label
状态	varchar(3)	待提交 待付款 处理中 已取消 已完成	purchase_status
操作者	varchar(20)	绑定了提交的操作者	operator
订单编号	bigint	外键 bill表	account_id

## 2.3 采编计划表(subscribe)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
采编编号	bigint	自增	subscribe_id
图书ID	bigint	外键 book表	book_id
采编时间	timestamp	默认当前时间	subscribe_time
供应商	varchar(20)		supplier
数量	integer		amount
状态	varchar(3)	已提交 未提交	subscribe_status
进货单号	bigint	外键进货表	purchase_id

## 2.4 图书库存(inventory)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
图书ID	bigint	外键 book表	book_id
馆藏位置	varchar(20)	外键 library表	location
库存数量	integer	总的数量	book_amount

## 2.5 图书馆建筑(library)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
馆藏位置	varchar(20)		location

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
分馆	varchar(3)	文科馆	理科馆

## 2.6 文科馆库存视图(arts\_lib\_inventory)

## 2.7 理科馆库存视图(sci\_lib\_inventory)

## 2.8 医科馆库存视图(med\_lib\_inventory)(三个视图格式一样但筛选内容不同)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
图书ID	bigint		book_id
ISBN	varchar(13)		ISBN
图书名称	varchar(20)		book_name
出版社	varchar(20)		publisher
作者	varchar(20)		author
出版年份	date		publish_date
零售价格	decimal(9,2)		unit_price
电子链接	tinytext		link
评级	decimal(2,1)		book_rank
类别	varchar(10)		book_class
馆藏位置	varchar(20)		location
库存数量	int		book_amount
分馆	varchar(3)		branch_lib

## 2.9 图书利用情况视图 (borrow\_view)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
图书ID	bigint	外键inventory	book_id
时间阶段	year		borrow_duration
借阅次数	integer	降序	borrow_times



## 2.10 流水账目表(bill)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
订单编号	bigint	自增	account_id
生成时间	timestamp	默认当前时间	create_time
结束时间	timestamp	支付或者取消时间	stop_time
类别	varchar(10)	采购图书 违规罚款	account_class
金额	decimal(9,2)		amount
缘由	text		account_reason
操作者	varchar(20)	默认当前操作员	operator
状态	varchar(3)	待付款 已支付 已取消	bill_status

## 2.11 待支付视图

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
账目编号	bigint	自增	account_id
生成时间	timestamp	默认当前时间	create_time
类别	varchar(10)	采购图书	account_class
金额	decimal(9,2)		amount
缘由	text		account_reason

## 2.12 管理员 (manage)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
管理员id	varchar (7)	工号	manage_id
姓名	varchar (20)		manage_name
密码	varchar (32)		manage_secret

## 2.13 学生 (student)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
学生id	varchar(11)	学号	student_id

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
姓名	varchar(20)		student_name
密码	varchar(32)		student_secret
诚信度	int	初始为30	

## 2.14 图书借阅图(borrow)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
借阅编号	bigint	自动增长	borrow_id
学生id	varchar(11)	外键	student_id
图书id	bigint	外键	book_id
借阅时间	timestamp	初始为当前时间	borrow_date
续借时间	int	初始为0,点击续借默认续借30天	continue_date

## 2.15 图书外借表(borrow\_on)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
外借编号	bigint	自动增长	borrow_on_id
借阅编号	bigint	外键borrow	borrow_id

## 2.16 图书归还表(return\_b)

ATTRIBUTE	DOMAIN	EXPLANATION	ENGLISHNAME
归还编号	bigint	自动增长	return_id
借阅编号	bigint	外键borrow	borrow_id
归还时间	timestamp	初始为创建时间	return_date

## 2.17 处罚表(fine)

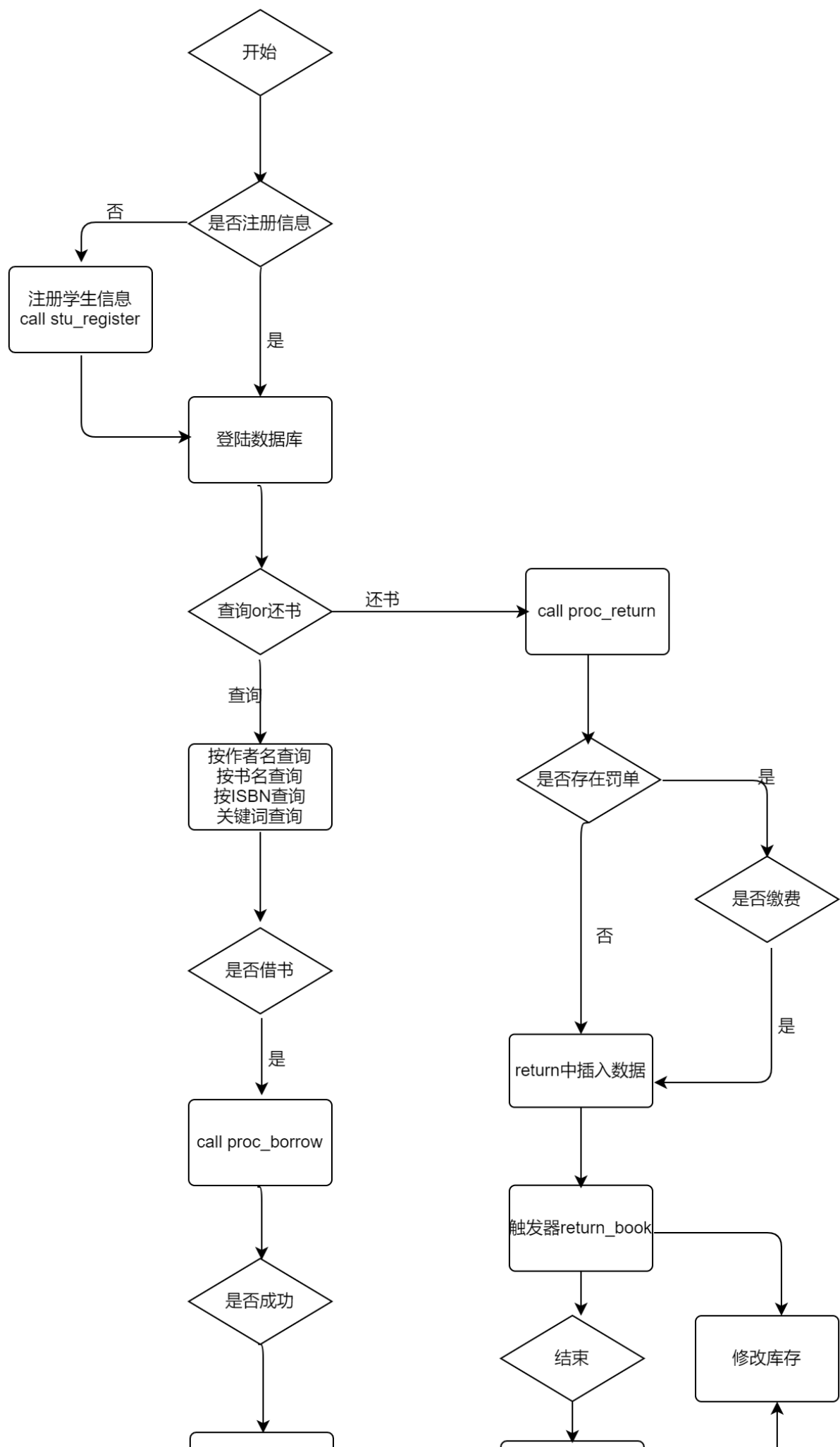
ATTRIBUTE	DOMAIN	EXPLANATION	ENGILSHNAME
处罚id	bigint	自动增长	fine_id
借阅编号	bigint	外键borrow	borrow_id

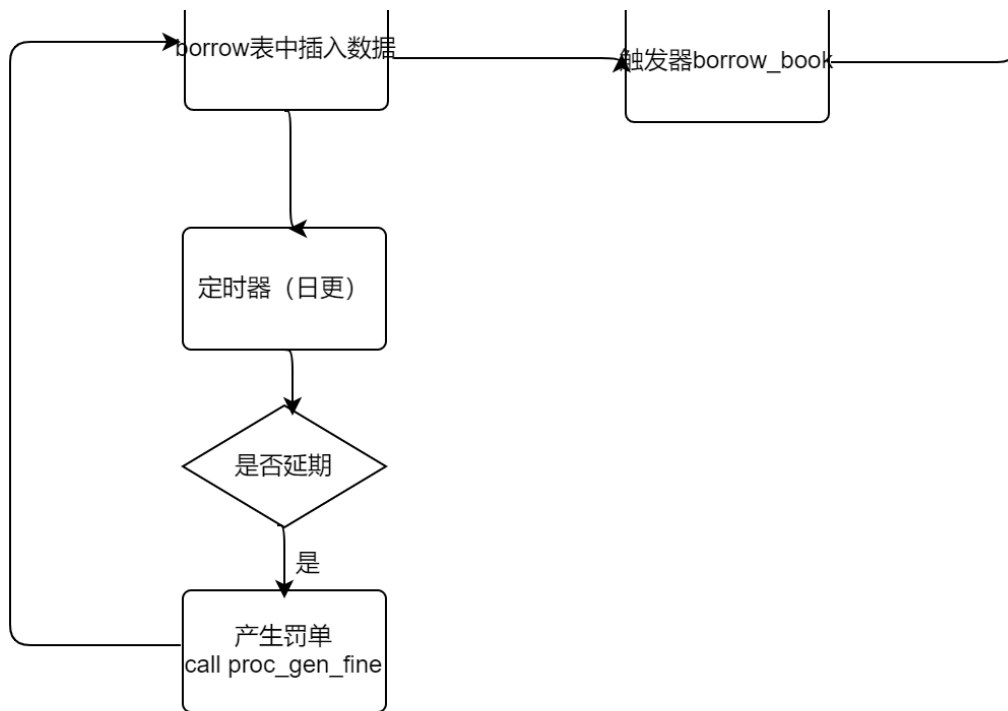
ATTRIBUTE	DOMAIN	EXPLANATION	ENGILSHNAME
罚款	decimal(9,2)		fine_fine
订单编号	bigint	外键bill	account_id

## 四、系统设计

### 1. 用户管理和流通借还子系统

#### 1.1 借阅流程图





## 1.2 用户登陆/管理员登陆

用户（管理员）输入用户名、密码,调用登陆过程`stu_log`（用户名，密码）;`manage_log`(用户名，密码)，若登陆成功，则返回用户或管理者的相关信息（id,姓名,密码,(诚信度) ),若登陆失败则返回相应的失败原因（用户名错误/密码输入错误）

```

1  -- 输入用户名、密码
2  create procedure manage_log(in manage_id varchar(11),in
   manage_secret varchar(20))
3  begin
4  -- 在manage表中查找是否存在该管路员，如果不存在该管理员,报错not exists
   this id,please register first
5  if(not exists (select manage_id
6  from manage
7  where manage_id=manage.manage_id)) then
8  select 'not exists this id,please register first' as info;
9  else
10 -- 在manage表格中验证密码是否正确，若登陆密码错误，提示secret is
    wrong,please check it。其中密码采用MD5加密的方法实现，所以在where条件中
    比较密码，应该采用加密过后的值与原表中储存的密码比较。
11 if (not exists (select manage_id
12 from manage
13 where manage_id=manage.manage_id and
    MD5(manage_secret)=manage.manage_secret)) then
14 select 'secret is wrong,please check it' as info;
  
```

```

15 else
16 -- 返回管理者信息, 包括id,姓名,密码
17 select manage_id,manage_name,manage_secret
18 from manage
19 where manage_id=manage.manage_id;
20 end if;
21 end if;
22 end

```

### 1.3 用户修改密码

用户输入原密码, 在student表中查找对应的用户,update密码, 密码采取MD5函数进行加密。

### 1.4 用户注册/管理员注册

```

1 -- 输入学生id, 学生姓名、登陆密码（未加密）（密码存入表中采用MD5加密），起始信
  用30分）
2 create procedure stu_register(in student_id varchar(11),in
  student_name varchar(20),in student_secret varchar(16))
3 begin
4 -- 在student表中插入一条新的记录,插入之后返回succeed
5 insert into student(student_id,student_name,student_secret)
6 values(student_id,student_name,MD5(student_secret));
7 select 'succeed';
8 end

```

### 1.5 查看用户正在借阅的书

返回图书名称、作者、出版年份、借阅时间、应还时间、续借时间

### 1.6 查看当前用户借阅历史

返回图书名称、作者、出版年份、应还时间、归还时间、罚款、馆藏位置

## 1.7 查看当前用户罚单（可检索罚单状态）

返回图书名称、作者、出版年份、金额

## 1.8 用户查询图书信息（可按照ISBN、作者、图书名）

图书id, 图书名称、作者、ISBN、出版社、出版年份、零售价格、电子链接、评级、类别、库存数量、馆藏位置

## 1.9 用户借书

若学生的诚信度大于0（创建函数func\_integrity返回该学生的诚信度）和借的书库存量大于0（创建函数func\_amount输入图书id和馆藏位置location,返回图书库存量），则向borrow表，borrow\_on表中插入一条记录，否则提示对应的错误。

## 1.10 向borrow表中插入记录

创建函数insert\_borrow，在insert into borrow后调用LAST\_INSERT\_ID()函数返回新插记录的borrow\_id

## 1.11 借书存储过程

```
1 CREATE DEFINER=`root`@`%` PROCEDURE `proc_borrow`(in student_id
  varchar(11), in book_id bigint,in location varchar(20))
2 begin
3 -- 调用函数func_integrity(student_id)判断该学生是否还有信用，若信用不
  够，则提示No credits,please try again 30 days later
4 declare b_i bigint;
5 if(func_integrity(student_id)=0) then
6 select 'No credits,please try again 30 days later' as info;
7 else
8 -- 调用函数func_amount(book_id,location)判断是否还有库存，若没有库存
  了，提示no book on the bookshelves
9 if(func_amount(book_id,location)=0) then
10 select 'no book on the bookshelves' as info;
11 -- 调用函数insert_borrow(student_id,book_id)向borrow中插入记录
12 else
13 set b_i=insert_borrow(student_id,book_id);
14 -- borrow_on中插入记录
15 insert into borrow_on(borrow_id)
16 values(b_i);
```

```

17  -- 同时库存表对应图书数量减1
18  update inventory
19  set book_amount=book_amount-1
20  where book_id=inventory.book_id and location=inventory.location;
21  -- 借书成功消息
22  select 'succeed borrow!' as succ;
23  end if;
24  end if;
25  end

```

## 1.12 续借

创建过程proc\_continue(in b\_o\_id bigint),首先从borrow\_on表中查找需要续借的书籍对应的borrow\_id,同时根据borrow\_id在borrow表中更改续借时间continue\_date

## 1.13 还书过程

若还书时间超过预期还书时间,定时器产生罚单,提示先交完罚单后还书,之后在还书表return中插入一条记录(输入借阅编号),同时修改库存表中的库存,删除borrow\_on表的该条借书记录

```

1  CREATE DEFINER=`root`@`%` PROCEDURE `proc_return`(in b_o_id
    bigint)
2  begin
3  -- 从borrow_on_id 取得 borrow_id
4  declare b_id bigint;
5  select borrow_id into b_id
6  from borrow_on
7  where borrow_on_id = b_o_id;
8  -- 在fine表中查找是否存在罚单且未支付,若存在则提示先交罚单,调用
    proc_payoff过程交款,提示please pay off the fine first
9  if(exists(select borrow_id
10 from fine
11 where b_id=fine.borrow_id)
12 and(func_pay(b_id)='待付款')) then
13 select 'please pay off the fine first' as info;
14 else
15 -- return 表中插入记录
16 insert into return_b(borrow_id)
17 values(b_id);
18 -- 删除borrow_on表

```



```

19 delete from borrow_on
20 where b_id=borrow_on.borrow_id;
21 -- 修改库存
22 update inventory
23 set book_amount=book_amount+1
24 where inventory.book_id=(select book_id
25 from borrow
26 where borrow.borrow_id=b_id);
27 select "succeeded return!" as succ;
28 end if;
29 end

```

### 1.14 支付罚单

创建支付过程proc\_payoff, 通过fine表中的外键关联找到bill中的对应该借书罚单的账单, 修改账单状态为已支付。

### 1.15 定时器产生罚单

创建事件timeclock,每天自动触发一次, 发现当前的预期归还时间小于当前时间, 调用产生罚单函数proc\_gen\_fine

### 1.16 触发器

```

1 -- 若存在事件timeclock 则drop
2 drop event if exists timeclock;
3 create event timeclock
4 -- 每天触发
5 on schedule every 1 day
6 on completion preserve disable
7 do
8 call proc_gen_fine();
9 set global event_scheduler=1;
10 alter event timeclock on completion preserve enable;

```

### 1.17 产生罚单和账单

```

1 -- 创建过程proc_gen_fine,产生罚单和账单
2 CREATE DEFINER=`root`@`%` PROCEDURE `proc_gen_fine`()
3 begin

```

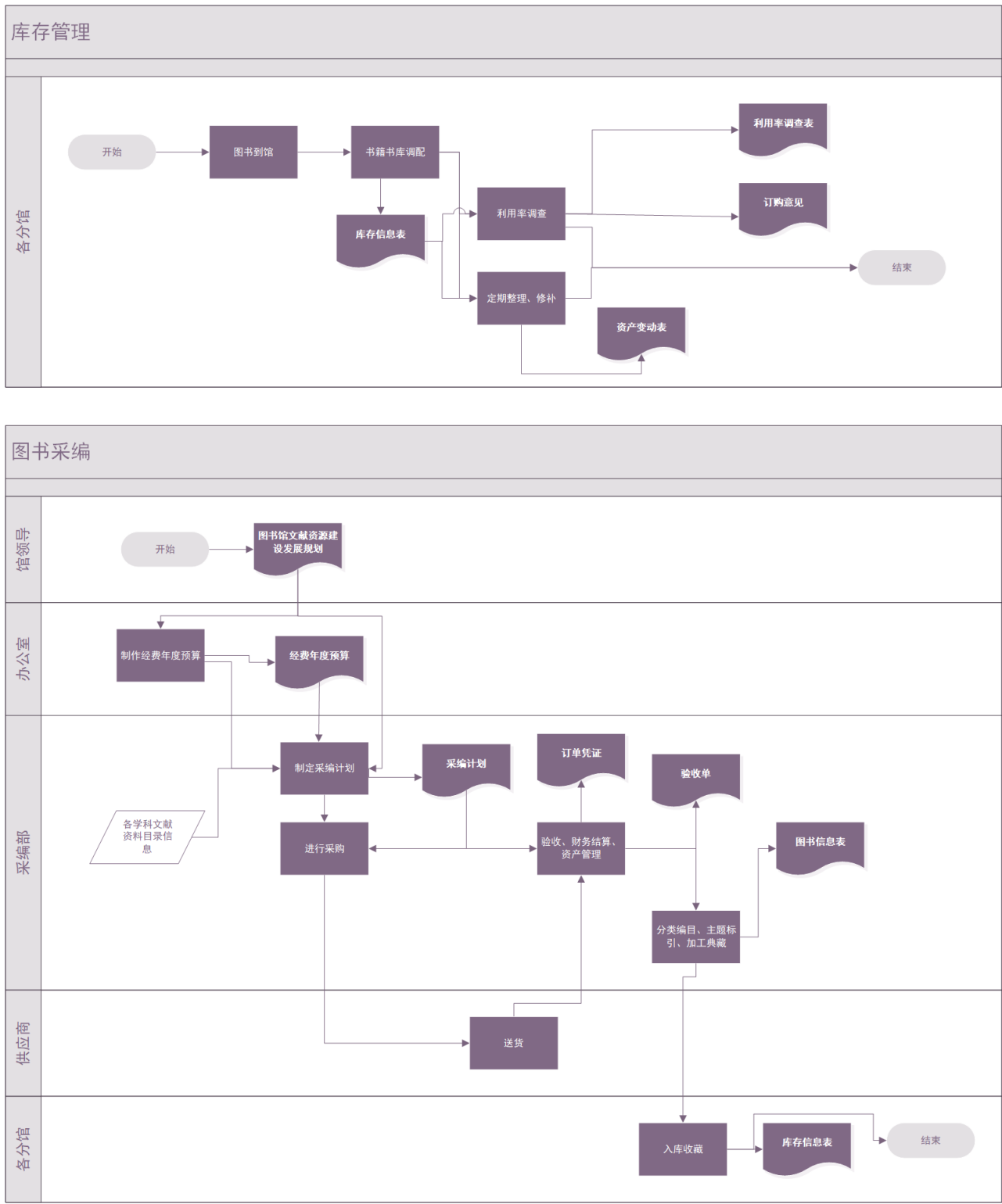
```
4  -- 定义游标接受值
5  declare student_id varchar(11);
6  declare a_i,b_i bigint;
7  declare fine_fee_1 decimal(9,2);
8  -- 游标结束标志
9  declare done int default 0;
10 -- 定义游标cur，每超出一天产生0.1元罚金
11 declare cur cursor for select student_id,borrow_id,0.1*
    (datediff(CURRENT_TIMESTAMP,borrow_date)-continue_date)
12 from borrow_on natural join borrow natural join student
13 where (datediff(CURRENT_TIMESTAMP,borrow_date)-continue_date)>0;
14 -- 指定游标循环结束时的返回值
15 declare continue handler for not found set done = 1;
16 open cur;
17 -- while 循环
18 while done!=1 do
19     fetch cur into s_i,b_i,fine_fee_1;
20     -- 如果在fine表中已经产生该罚单，则update更新fine表和bill对应借阅记录的罚金
21     if(exists (select borrow_id
22     from fine
23     where borrow_id=b_i)) then
24         update bill
25         set amount=fine_fee_1
26         where account_id=(select account_id
27         from fine
28         where borrow_id=b_i);
29         update fine
30         set fine_fine=fine_fee_1
31         where borrow_id=b_i;
32     else
33         -- 如果fine表中未存在对应的罚单，则调用过程insert_bill_fine在fine和bill
            表中产生罚单和账单。
34         call insert_bill_fine(fine_fee_1,b_i,s_i);
35     end if;
36 end while;
37 -- 关闭游标
38 close cur;
39 end
```

1.18 插入bill和fine表insert\_bill\_fine

insert into 向bill,同时LAST\_INSERT\_ID()函数获取bill的主键account\_id的值，插入fine表

2. 采编管理和财务管理子系统

2.1 流程图



## 2.2 功能概述

类别	功能
图书库存管理	图书信息增删改查
	库存信息增删改查
图书采编管理	图书采编计划增删改查
	图书进货状态跟踪
财务管理	流水账目管理

## 2.3 图书库存管理

### 2.3.1 图书信息增删改查

支持图书（book表）的增删改查

增添一行时管理员可以输入除了图书ID(book\_id)以外的所有属性,图书ID为自增属性，图书名称为必填属性。

查找一行时，管理员可以根据图书名称、作者、ISBN、馆藏位置进行模糊查询。

### 2.3.2 库存信息增删改查

库存信息对应(inventory)表。

增添一行时需要输入对应的图书ID和馆藏位置以及库存数量。

查找一库存时，管理员可以根据图书名称、作者、ISBN、馆藏位置进行查找。前三项是模糊查询，馆藏位置是下拉框进行选择。

## 2.4 图书采编管理

### 2.4.1 图书采编计划增删改查

采编表(subscribe)相当于加入购物车的一件商品

进货表(purchase)相当于一个订单

财务流水(bill)表相当于支付宝那边的一个流水账单

增添一项时,输入图书ID、供应商、采编数量。这时候生成的一栏采编项默认的状态时“未提

交”。

查看采编表的时候，可以根据状态（“未提交”、“已提交”来查看）。

提交采编计划的时候，系统会将所有的“未提交”的采编项进行提交。

调用subscribe\_to\_purchase 过程

输入参数为当前操作者（即管理员用户的名称）

- 详细代码：

```
1  -- 提交采编项
2  CREATE DEFINER=`root`@`%` PROCEDURE `subscribe_to_purchase`(in
   op varchar(20))
3  BEGIN
4  -- 计算采编的总金额
5  declare cost decimal(9,2);
6  declare pur_id bigint;
7  select sum(total_price) into cost
8          from (select unit_price*amount as
   total_price
9                  from subscribe natural join
   book
10                 where subscribe_status='未提
   交') as T;
11 -- 插入进货表一行
12 insert into purchase(cost_label,operator)
13 values(cost,op);
14 -- 取得刚插入的id
15 set pur_id = LAST_INSERT_ID();
16 -- 更新采编表的进货编号
17 update subscribe
18 set subscribe.purchase_id = pur_id
19 where subscribe.subscribe_status = '未提交';
20 -- 更新采编表的状态
21 update subscribe
22 set subscribe.subscribe_status='已提交'
23 where subscribe.subscribe_status = '未提交';
24 END
```

- 解释：

该过程计算所有的状态为“未提交”的采编项的总金额，并且将此信息、结合当前时间、操作者等信息插入进货表(purchase)中，并且取得插入进货表中的新

行的（自增的）id，更新到对应的采编项中的属性，表示该采编项和该进货项之间的关联。

#### 2.4.2 图书进货状态跟踪

进货状态包括  
（待提交  
待付款  
处理中  
已取消  
已完成）

管理员可以对采编表项按照状态进行查询。

采编管理处的提交会在进货表中插入一行状态为“待提交”的进货项，此时管理员对每一个进货项可以进行取消、提交操作。此时取消，进货项状态变为“已取消”，并且不能再进行其他的操作。

提交操作是将该进货信息提交到财务系统处，表示请求财务管理处的人进行支付，提交后状态变为“待付款”。

财务管理员支付了该订单以后，进货表中该项的状态由“待付款”变为“处理中”，进货管理员如果收到了图书，就可以对该进货项进行确认收货，该进货项的状态变为“已完成”。

- 代码（提交进货表到财务系统）：

```
1  -- 提交进货表
2  CREATE DEFINER=`root`@`%` PROCEDURE `submit_purchase`(
3  in p_id BIGINT,
4  in a_r text
5  )
6  BEGIN
7  declare a_id BIGINT;
8  declare c_l decimal(9,2);
9  -- 取得进货编号对应的进货记录/元组的应付金额,储在变量中
10 select cost_label into c_l from purchase where
    purchase.purchase_id = p_id;
11 -- 流水账目表插入一个元组,取得返回值账目编号
12 set a_id = insert_bill("采购图书", c_l, a_r, "待付款");
13 -- 进货表修改账目编号属性
14 -- 把所有的待提交的purchase元组的状态都改为待付款
15 update purchase
16 set purchase_status="待付款", account_id=a_id
17 where purchase_id = p_id;
18 END
```

- 说明

该过程输入为进货表项的id，以及提交时管理员输入的一些说明信息。这些信息都从前端传入。

该过程向财务流水表（bill）中插入一条记录，类型(account\_class)为“采购图书”，并且把新生成的bill表中的记录的id作为属性写回进货表(purchase)中的对应项中，表示二者的关联。

## 2.5 财务管理

### 2.5.1 流水账目管理

财务表记录的状态包括（  
待付款  
已支付  
已取消  
）

管理员可以对财务流水表项按照状态进行查询。

目前系统涉及的流水表类型主要是两项：一个是“采购图书”类别的支出，一个是“违规罚款”的收入，后者表示借书过程中读者缴纳的罚款。

默认新插入的这两项的状态都为“待支付”，出于该状态的记录，财务管理员可以进行支付、取消、收款三种操作，支付、收款都会将该记录的状态改为“已支付”，而取消则将该记录的状态变为已取消。

当财务管理人员支付了“采购图书”类别的流水项时，该流水项对应的进货表（purchase）项的状态会自动变为“处理中”，该功能通过一个触发器来实现。

- 代码：

```
1  -- 触发器：当对应的订单被支付，则修改进货单中对应元组的状态从"待支付"为"处理中"
2  create trigger purchase_in_process after update
3  on bill
4  for each row
5  BEGIN
6  -- 如果是支付了该采购图书订单
7  if (new.account_class='采购图书'and new.bill_status='已支付')
8  -- 修改对应进货单的状态为"处理中"
9  then
10 update purchase
11 set purchase_status = "处理中"
```

```
12 where purchase.account_id = new.account_id;  
13 end if;  
14 -- 还要把“确认收货展示出来”  
15 END;
```

## 3. 前端页面设计

### 3.1 用户界面

#### 3.1.1 登录界面

登陆界面样式：

读者系统 登录界面

127.0.0.1:8000/userApp/

应用 Gmail Google YouTube 地图 百度翻译 Google 翻译 快乐 学习 fdu 阅读清单

读者系统 登录界面

### 登录

18300240024

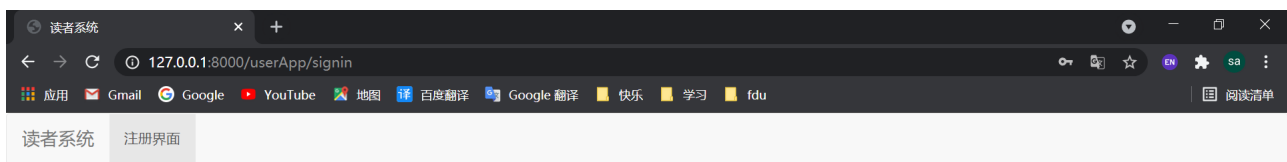
...

登录

注册

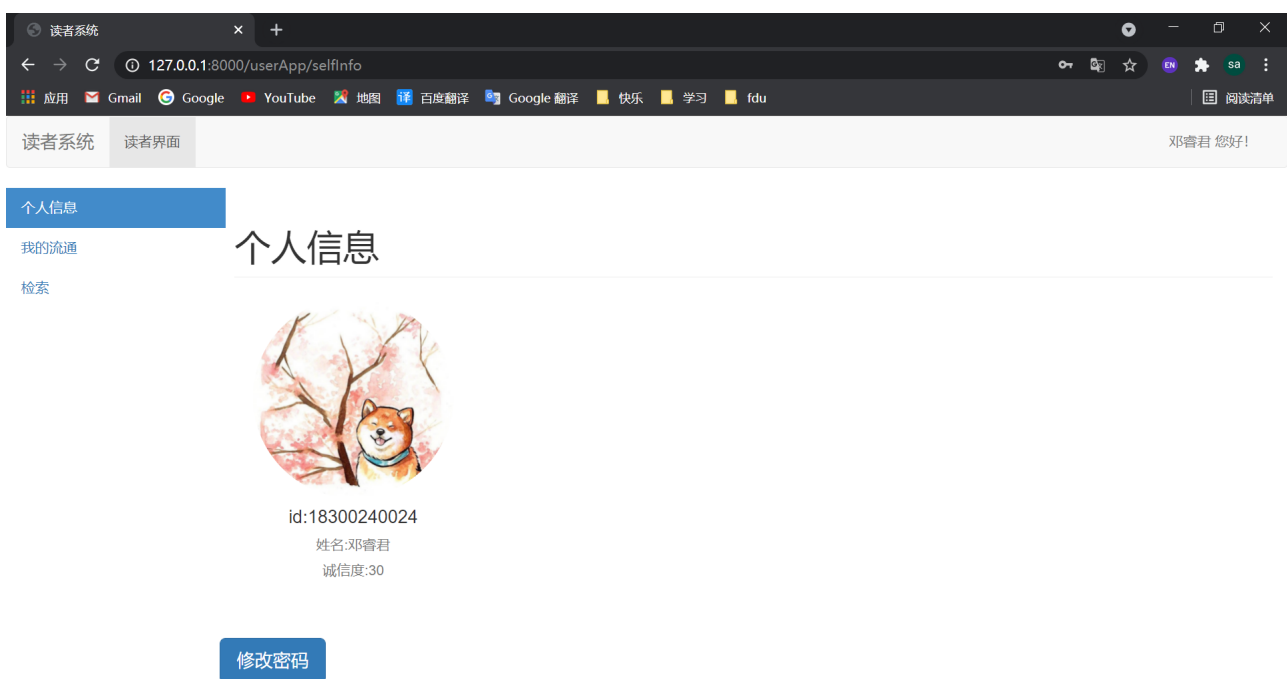
#### 3.1.2 注册界面





### 3.1.3 用户操作界面

#### 3.1.3.1 个人信息



#### 3.1.3.2 图书检索

个人信息

我的流通

检索

# 图书检索

书名 新兵

作者 输入作者...

ISBN 输入ISBN...

查询

图书ID	ISBN	图书名称	出版社	作者	出版日期	电子链接	图书评级	图书类别	馆藏位置	库存量	操作
1	9787500159919	微表情心理学	北京中译出版社	王雄	2019-01-03	None	None	文学艺术	文科综合书库	0	借阅
2	9787520523608	新兵	北京中国文史出版社	北桥	2021-01-03	None	None	哲学宗教	文学艺术书库	1	借阅
3	9787502486969	生物化学	北京冶金工业出版社	常雁红	2021-01-03	None	None	化学生物	文科过刊库	1	借阅
4	9787302573746	操作系统本质	北京清华大学出版社	陈鹏	2021-01-03	None	None	信息电脑	期刊工作室	2	借阅
5	9787500159915	表情心理学	北京中译出版社	王雄	2021-01-03	None	None	文学艺术	理科应用学科阅览室	4	借阅
6	6	一本书	None	drj	None	None	7.0	None	理科应用学科阅览室	2	借阅

## 3.1.3.3 流通管理

个人信息

我的流通

检索

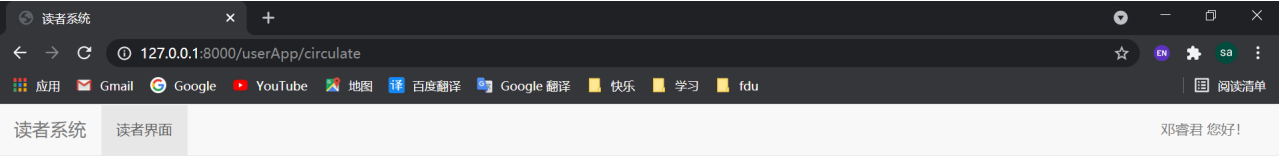
# 流通信息

在借阅 借阅历史 罚单

全部罚单

查询

罚单编号	图书名称	作者	出版时间	罚款	状态	操作
172	表情心理学	王雄	2021-01-03	0.80	已支付	
173	微表情心理学	王雄	2019-01-03	3.80	待付款	支付罚款



个人信息

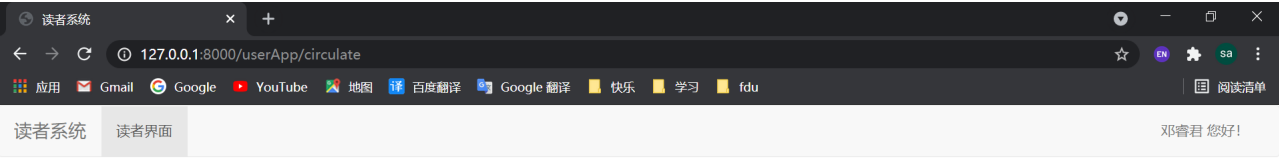
我的流通

# 流通信息

检索

在借阅 借阅历史 罚单

图书名称	ISBN	作者	出版社	借阅日期	续借时间	归还日期	馆藏位置
微表情心理学	王雄	9787500159919	北京中译出版社	2021-06-10	0	2021-06-10	文科综合书库
表情心理学	王雄	9787500159915	北京中译出版社	2021-06-10	30	2021-06-10	理科应用学科阅览室
新兵	北桥	9787520523608	北京中国文史出版社	2021-06-10	30	2021-06-10	文学艺术书库



个人信息

我的流通

# 流通信息

检索

在借阅 借阅历史 罚单

外借编号	图书名称	作者	出版年份	借阅时间	预计归还时间	续借时间	操作
14	表情心理学	王雄	2021-01-03	2021-06-10	2021-10-08	30	<a href="#">续借</a> <a href="#">还书</a>
15	微表情心理学	王雄	2019-01-03	2021-06-10	2021-09-08	0	<a href="#">续借</a> <a href="#">还书</a>
16	生物化学	常雁红	2021-01-03	2021-06-10	2021-09-08	0	<a href="#">续借</a> <a href="#">还书</a>
17	新兵	北桥	2021-01-03	2021-06-10	2021-09-08	0	<a href="#">续借</a> <a href="#">还书</a>
18	新兵	北桥	2021-01-03	2021-06-11	2021-09-09	0	<a href="#">续借</a> <a href="#">还书</a>

文件占用中，关闭后将自动备份  
“登陆界面(1)....”等2个文件尚未完成备份  
[立即查看>](#)

## 3.2 管理员界面

### 3.2.1 图书信息页面



书名  作者  ISBN

	book_id	ISBN	book_name	publisher	author	publish_date	unit_price	link	book_rank	book_class	操作
1		9787500159919	微表情心理学	北京中译出版社	王雄	2019-01-03	149.00	None	None	文学艺术	<a href="#">修改</a> <a href="#">删除</a>
2		9787520523608	新兵	北京中国文史出版社	北桥	2021-01-03	49.80	None	None	哲学宗教	<a href="#">修改</a> <a href="#">删除</a>
3		9787502486969	生物化学	北京冶金工业出版社	常雁红	2021-01-03	42.00	None	None	化学生物	<a href="#">修改</a> <a href="#">删除</a>
4		9787302573746	操作系统本质	北京清华大学出版社	陈鹏	2021-01-03	59.00	None	None	信息电脑	<a href="#">修改</a> <a href="#">删除</a>
5		9787500159915	表情心理学	北京中译出版社	王雄	2021-01-03	129.00	None	None	文学艺术	<a href="#">修改</a> <a href="#">删除</a>
6		6	一本书	None	drj	None	None	None	7.0	None	<a href="#">修改</a> <a href="#">删除</a>

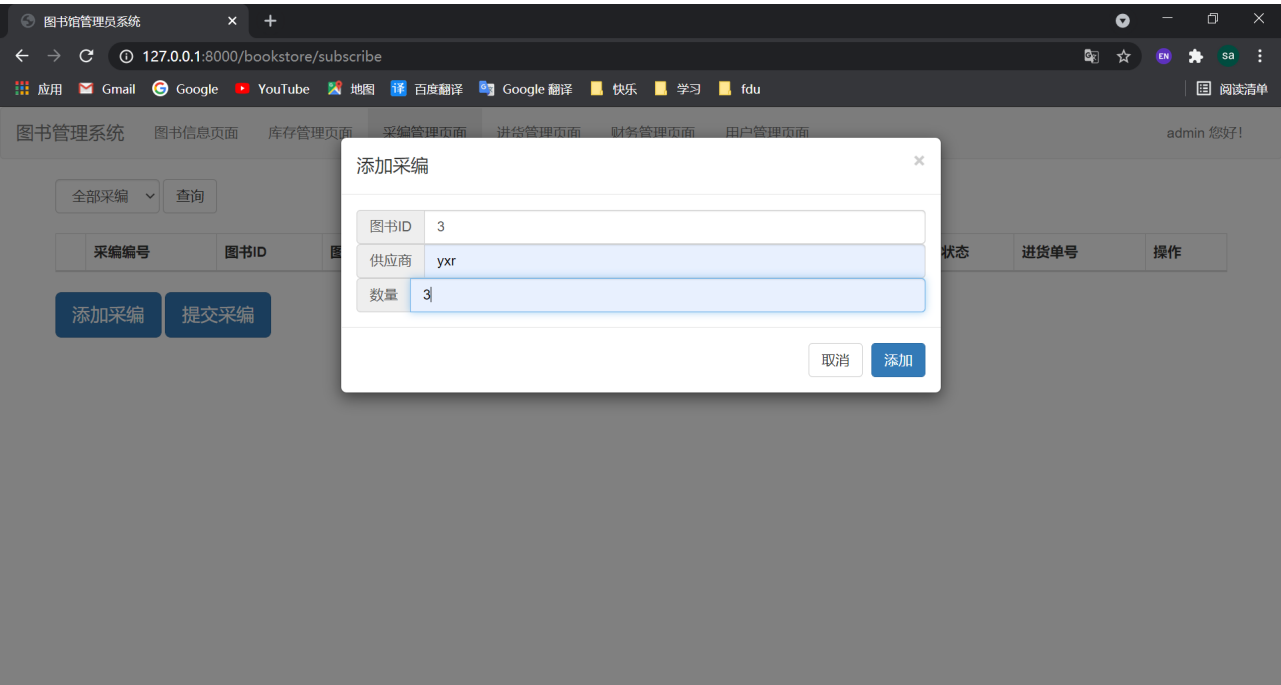
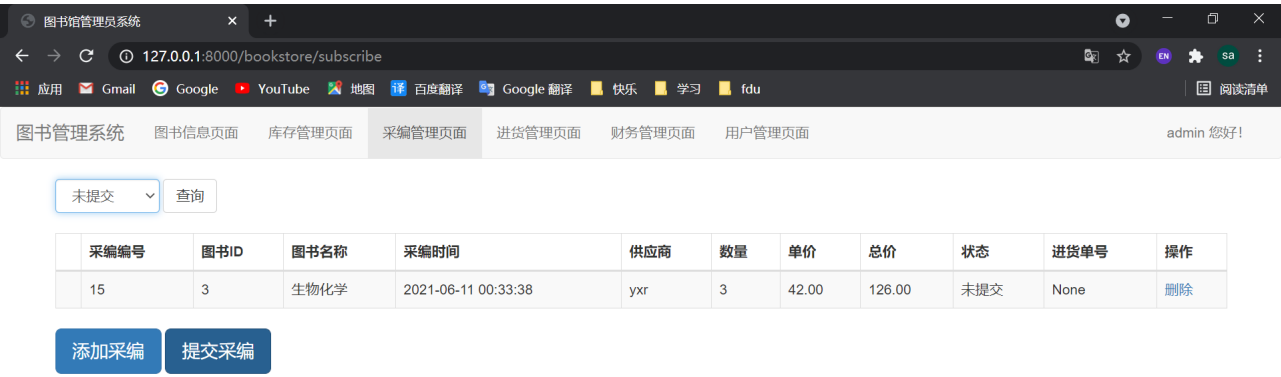
### 3.2.2 库存管理页面



书名  作者

	book_id	book_name	author	ISBN	publisher	publish_date	location	book_amount	操作
1		微表情心理学	王雄	9787500159919	北京中译出版社	2019-01-03	文科综合书库	0	<a href="#">修改</a> <a href="#">删除</a>
2		新兵	北桥	9787520523608	北京中国文史出版社	2021-01-03	文学艺术书库	1	<a href="#">修改</a> <a href="#">删除</a>
3		生物化学	常雁红	9787502486969	北京冶金工业出版社	2021-01-03	文科过刊库	1	<a href="#">修改</a> <a href="#">删除</a>
4		操作系统本质	陈鹏	9787302573746	北京清华大学出版社	2021-01-03	期刊工作室	2	<a href="#">修改</a> <a href="#">删除</a>
5		表情心理学	王雄	9787500159915	北京中译出版社	2021-01-03	理科应用学科阅览室	4	<a href="#">修改</a> <a href="#">删除</a>
6		一本书	drj	6	None	None	理科应用学科阅览室	2	<a href="#">修改</a> <a href="#">删除</a>

### 3.2.3 采编管理页面



3.2.4 进货管理页面

图书管理系统							
127.0.0.1:8000/bookstore/purchase							
应用 Gmail Google YouTube 地图 百度翻译 Google 翻译 快乐 学习 fdu							
图书管理系统 图书信息页面 库存管理页面 采编管理页面 进货管理页面 财务管理页面 用户管理页面 admin 您好!							
全部进货 查询							
purchase_id	purchase_time	cost_label	purchase_status	operator	account_id	操作	
1	2021-06-09 02:47:33	60.00	处理中	None	1	确认收货	
2	2021-06-09 03:04:18	100.00	处理中	None	2	确认收货	
3	2021-06-09 03:10:23	100.00	处理中	None	3	确认收货	
4	2021-06-09 20:02:21	210.00	处理中	admin	4	确认收货	
5	2021-06-10 19:56:36	763.00	处理中	admin	176	确认收货	
6	2021-06-10 19:57:19	387.00	待提交	admin	None	提交 取消	
7	2021-06-10 20:00:42	745.00	待提交	None	None	提交 取消	
8	2021-06-10 20:03:08	298.00	待提交	admin	None	提交 取消	
9	2021-06-10 20:21:24	126.00	待提交	admin	None	提交 取消	
10	2021-06-10 20:58:54	645.00	处理中	admin	177	确认收货	
11	2021-06-11 00:34:50	126.00	待提交	admin	None	提交 取消	

## 3.2.5 财务管理页面

图书管理系统								
127.0.0.1:8000/bookstore/finance								
应用 Gmail Google YouTube 地图 百度翻译 Google 翻译 快乐 学习 fdu								
图书管理系统 图书信息页面 库存管理页面 采编管理页面 进货管理页面 财务管理页面 用户管理页面 admin 您好!								
全部流水 查询								
账目编号	生成时间	结束时间	类别	金额	缘由	操作者	状态	操作
1	2021-06-09 02:59:09	2021-06-09 18:28:08	采购图书	60.00	None	None	已支付	
2	2021-06-09 03:04:37	2021-06-09 18:19:35	采购图书	100.00	提交进货单	None	已取消	
3	2021-06-09 03:10:39	2021-06-09 18:19:42	采购图书	100.00	提交进货单	None	已收款	
4	2021-06-09 20:02:34	2021-06-09 20:03:18	采购图书	210.00	提交进货单	admin	已收款	
176	2021-06-10 20:05:43	2021-06-10 20:05:57	采购图书	763.00	提交进货单	admin	已支付	
177	2021-06-10 20:59:10	2021-06-10 20:59:23	采购图书	645.00	提交进货单d	admin	已支付	
178	2021-06-10 21:04:49	2021-06-10 21:05:01	违规罚款	0.80	None	None	已支付	
179	2021-06-10 21:04:50	None	违规罚款	3.80	None	None	待付款	支付 取消 收款

## 3.2.6 用户管理页面

用户id	用户名	用户诚信度	操作
18300240024	邓睿君	30	<a href="#">删除</a>
18300240025	路小凡	30	<a href="#">删除</a>
19307130360	叶鑫茹	30	<a href="#">删除</a>
19307130361	甲	30	<a href="#">删除</a>
19307130363	丙	30	<a href="#">删除</a>
19307130364	丁	30	<a href="#">删除</a>

### 3.3 模板继承

用户界面和管理界面分别从管理员应用"bookstore"和用户应用"userApp"中的base.html模板中继承html标签元素，该base.html包括导航条等固定内容，并且预留出供子模板使用的块，使得前端的设计的代码可以被复用，提高程序的灵活性、可复用性。

- 父模板(bookstore\base.html)

```

1  ...
2  <!--预留出这样的模块供子模版添加-->
3  {% block tableInfo %}
4  {% endblock %}
5  ...

```

- 子模板: (bookstore\inventory.html)

```

1  <!--继承父模板-->
2  {% extends "bookstore/base.html" %}
3
4  <!--扩展父模板预留出来的块-->
5  {% block tableInfo %}
6  <div class="table-responsive">
7      <table class="table table-bordered table-striped table-
8      hover">
9      <thead>
10     <th></th>

```

```

10     {% for name in inventoryDisplayList %}
11     <th>{{ name }}</th>
12     {% endfor %}
13     <th>操作</th>
14     </thead>
15     <tbody>
16     {% if queryList %}
17         {% for row in queryList %}
18             <tr>
19                 <td> </td>
20                 <td class="book_id">{{ row.book_id }}</td>
21                 <td class="book_name">{{ row.book_name }}</td>
22                 <td class="author">{{ row.author }}</td>
23                 <td class="ISBN">{{ row.ISBN }}</td>
24                 <td class="publisher">{{ row.publisher }}</td>
25                 <td class="publish_date">{{ row.publish_date }}
26
27                 <td class="location">{{ row.location }}</td>
28                 <td class="book_amount">{{ row.book_amount }}
29
30                 <td class="operate">
31                     <a href="#" data-toggle="modal" data-
32                     target="#modifyInventoryModal" onclick="modify_inventory(this)">
33                     修改</a>
34                     <a href="#"
35                     onclick="delete_inventory(this)">删除</a>
36                 </td>
37             </tr>
38         {% endfor %}
39     {% endif %}
40     </tbody>
41     </table>
42 </div>
43
44 {% endblock %}
45

```



## 3.4 提示消息

### 3.4.1 删除库存提示

- 代码:

```
1  //删除库存信息
2  function delete_inventory(obj){
3      ret = confirm("确定删除库存吗? ");
4      if(ret){ //确定
5          row = $(obj).parent().parent(); // 操作的行
6          $.ajax({
7              url: "{% url 'deleteInventory' %}",
8              type: "POST",
9              data: {
10                  "book_id":$(row).children('.book_id').text(),
11                  "location":$(row).children('.location').text(),
12              },
13              success: function (backData) {
14                  if(backData.status==="0")//减少成功
15                      alert("删除成功!");
16                  else if(backData.status==="1")// 减少失败
17                      alert("删除失败!");
18              }
19          })
20      }
21  }
```

- 解释: 当点击删除库存操作的时候, 触发该函数, 提示确认是否删除库存, 如果答案是确认, 那么利用ajax向后端发起http请求, 并且接受后端返回的json数据, 使用alert提示操作结果(成功或失败)

## 3.5 url导航

django框架下配置url路径导航:

下面是用户应用的所有url配置

```
1  urlpatterns = [
2      path('', views.login, name='login'), # 登录页面
3      path('loginSubmit', views.login_submit, name='loginSubmit'),
4      # 提交登录
```

```

4     path('signin', views.signin, name='signin'), # 注册页面
5     path('signinSubmit', views.signin_submit,
name='signinSubmit'), # 提交注册
6     path('selfInfo', views.self_info, name='selfInfo'), # 个人信
息
7     path('circulate', views.circulate, name='circulate'), # 流通
借还
8     path('search', views.search, name='readerSearch'), # 借阅页
面
9     path('borrow', views.borrow, name='borrow'), # 借一本书
10    path('continueBorrow', views.continue_borrow,
name='continueBorrow'), # 续借一本书
11    path('returnBook', views.return_book, name='returnBook'), #
还一本书
12    path('payFine', views.pay_fine, name='payFine'), # 支付罚金
13    path('modifyPassword', views.modify_password,
name='modifyPassword'), # 修改密码
14 ]

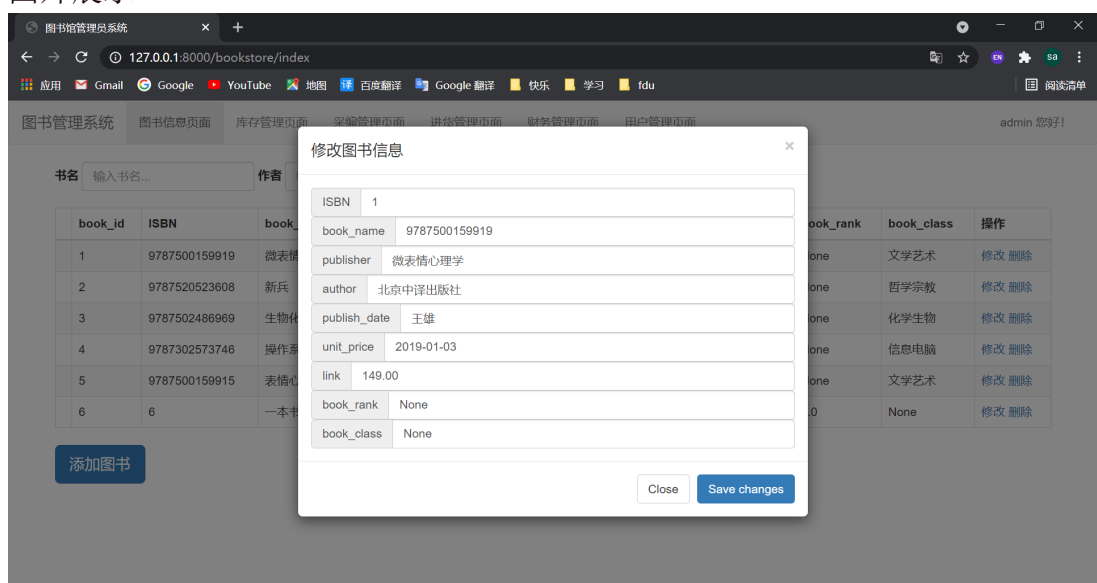
```

## 3.6 bootstrap3 框架

使用了bootstrap3的导航条，按钮，响应式表格、模态框等组件

### 3.6.1 模态框：

- 图片展示



- 说明：  
点击修改图书按钮，弹出模态框，利用javascript函数将该图书对应的信息先填入框中，等待前端用户修改。

## 4. 前后端交互设计

web服务器与web浏览器使用http协议进行交互

web服务器与数据库服务器使用django提供的数据库连接配置进行交互

### 4.1 数据库服务器配置

#### 4.1.1 数据库服务器授权远程用户

在开发环境下，这里为了方便，允许所有的ip访问数据库，但在生产环境下这样存在安全问题，需要设定特定的ip范围

允许所有的ip以root用户身份访问数据库

```
1 update user set host = '%' where user = 'root';
2 flush privileges;
```

授权root 用户所有的权限，密码为123456

```
1 mysql -uroot -p
2 ## mysql>命令 begin
3 grant all privileges on *.* to 'root'@'%' identified by '123456';
4 flush privileges;
5 ## end
6 exit
```

#### 4.1.2 web服务器连接数据库服务器

数据库的连接的配置如下：(djangoBookstore\settins.py)

```
1 DATABASES = {
2     'default': {
3         'ENGINE': 'django.db.backends.mysql',
4         'NAME': 'bookwarehouse', # 数据库名称
5         'USER': 'root', # 虚拟机mysql授权的用户
6         'PASSWORD': '123456', # 用户密码
7         'HOST': '192.168.221.133', # 虚拟机ip
8         'PORT': 3306 # 端口号
9     }
10 }
```

### 4.1.3 增删改查功能的前后端交互以及数据库访问

#### 4.1.3.1 查询图书

```
1  # 图书搜索查询
2  def search_book(request):
3      if request.method == "POST":
4          input_raw = [] # 储存图书信息 ,若没有输入某项查询,则值为空串""
5          input_raw.append(request.POST.get("book_name")) # 增添上
           图书信息的一个字段
6          input_raw.append(request.POST.get("author")) # 增添上图书
           信息的一个字段
7          input_raw.append(request.POST.get("ISBN")) # 增添上图书信
           息的一个字段
8
9          # 执行sql语句
10         with connection.cursor() as cursor:
11             sqlquery = """SELECT *
12             FROM book
13             WHERE book_name like '%{}%' and author like '%{}%' and ISBN like '%{}%';""".format(input_raw[0],
14                                                         input_raw[1],
15                                                         input_raw[2])
16             cursor.execute(sqlquery) # 执行sql语句
17             queryBookList = dictfetchall(cursor) # 取得执行结果
18
19         for dic in queryBookList: # 把时间日期类型转成字符串
20             dic = dic2str(dic)
21         template = loader.get_template('bookstore/index.html')
           # 前端模板文件
22         context = { # 需要渲染到前端的数据
23             'queryBookList': queryBookList,
24             'bookTableList': bookTableList,
25         }
26         # 返回httpresponse 响应, 渲染前端
27         return render(request, 'bookstore/index.html', context)
```

#### 4.1.3.2 修改图书信息

```
1  # 改图书信息
2  def modify_book(request):
3      if request.method == "POST":
4          try:
5              inputbook_raw = [] # 储存图书信息
6              for name in bookTableList:
7                  inputbook_raw.append(request.POST.get(name)) #
增添上图书信息的一个字段
8                  # 未输入的值用None代替
9                  modifybook = [None if i == '' else i for i in
inputbook_raw] # 输入的字段值
10                 sqlquery = """UPDATE book
11                               SET
12                               ISBN=%s,
13                               book_name=%s,
14                               publisher=%s,
15                               author=%s,
16                               publish_date=%s,
17                               unit_price=%s,
18                               link=%s,
19                               book_rank=%s,
20                               book_class=%s
21                               WHERE book_id=%s;"""
22                 modifybook.append(modifybook[0]) # 加上最后的book_id
匹配参数
23                 # 执行sql语句
24                 with connection.cursor() as cursor:
25                     cursor.execute(sqlquery, modifybook)
26                 return JsonResponse({"status": "0"}) # 插入成功
27             except BaseException:
28                 return JsonResponse({"status": "1"}) # 插入失败
```

#### 4.1.3.3 插入图书

```
1  ## 插入图书
2  def insert_book(request):
3      if request.method == "POST":
4          try:
5              inputbook_raw = [] # 储存图书信息
6              for name in bookTableList:
```

```

7         inputbook_raw.append(request.POST.get(name)) #
    增添上图书信息的一个字段
8         # 未输入的值用None代替
9         inputbook = [None if i == '' else i for i in
    inputbook_raw]
10        sqlquery = "INSERT INTO book VALUES
    (%s,%s,%s,%s,%s,%s,%s,%s,%s);"
11        # 执行sql语句
12        with connection.cursor() as cursor:
13            cursor.execute(sqlquery, inputbook)
14            return JsonResponse({"status": "0"}) # 插入成功
15        except BaseException:
16            return JsonResponse({"status": "1"}) # 插入失败

```

#### 4.1.3.4 删除图书信息

```

1  # 删除图书信息
2  @csrf_exempt # 屏蔽csrf（不安全）
3  def delete_book(request):
4      if request.method == "POST":
5          try:
6              book_id = request.POST.get("book_id")
7
8              sqlquery = """
9              DELETE from book
10             where book_id = %s
11             """
12             with connection.cursor() as cursor:
13                 cursor.execute(sqlquery, [book_id])
14                 return JsonResponse({"status": "0"}) # 删除成功
15             except BaseException:
16                 return JsonResponse({"status": "1"}) # 删除失败

```

## 4.2 调用数据库中的存储过程

### 4.2.1 调用"submit\_purchase()"过程 提交进货表

```

1  # 提交进货表
2  @csrf_exempt # 屏蔽csrf (不安全)
3  def submit_purchase(request):
4      purchase_id= request.POST.get("purchase_id")
5      account_reason = request.POST.get("account_reason") # 提交说明
6      try:
7          with connection.cursor() as cursor:
8              cursor.callproc('submit_purchase',
9                             [purchase_id,account_reason])
10             return JsonResponse({"status": "0"}) # 提交成功
11         except BaseException:
12             return JsonResponse({"status": "1"}) # 提交失败

```

### 4.3 在会话中存储身份信息

```

1  # 提交登录信息
2  def login_submit(request):
3      if request.method == "POST":
4          student_id = request.POST.get("student_id") # 用户id
5          student_secret = request.POST.get("student_secret") # 用户密码
6          input = [student_id, student_secret]
7          # call 登录
8          try:
9              with connection.cursor() as cursor:
10                 cursor.callproc('stu_log', input)
11                 # 成功
12                 result = dictfetchall(cursor)
13                 if "info" in result[0]: # 登录失败:
14                     message = result[0]["info"] # 失败消息
15                     return render(request, "userApp/login.html",
16                                   locals())
17                 else: # 储存登录者信息在会话中
18                     request.session['student_id'] = result[0]
19                     request.session['student_name'] = result[0]
20                     request.session['student_integrity'] =
21                     result[0]["student_integrity"]

```

```

20         request.session['student_secret'] =
result[0]["student_secret"]
21         return redirect('selfInfo') # 重定向到个人信息
界面
22     except BaseException: # 失败
23         message = "系统错误！"
24         return render(request, "userApp/login.html",
locals())

```

## 五、创新点和特色

本项目采用了B/S结构，实现了面向用户和管理员的两个子应用。主要借鉴了复旦大学图书馆信息系统的设计。基本满足学校图书管理系统的核心需求。

我们的选题是中规中矩的，但是该选题与数据库的设计关系密切，让我们把注意力集中在数据库相关知识本身上。

## 六、实验分工

叶鑫茹：

- 用户管理和流通借还数据库设计和sql语句
- 用户管理和流通借还ui界面设计
- 整合er图、表
- 数据收集填写

邓睿君：

- 库存、采编管理和财务管理数据库设计和sql语句
- 流通借还和财务管理ui界面设计
- ui界面实现
- 前后端交互实现

## 七、文件提交说明

- bookstore.sql: 本项目所有的sql脚本
- djangoBookstore: 项目代码 包括userApp 和 bookstore两个应用 以及项目的配置文件



- 两应用中的view.py实现了所有的视图函数(主要的python代码)
- 两应用中的urls.py实现了路径导航
- 两应用中的templates中储存了所有html文件

## 八、实验总结

邓睿君：

通过本次实验，加深了我对数据库相关知识的理解。学会了mysql服务器的搭建和mysql使用的语言，Navicat的使用，django框架和bootstrap框架的使用，锻炼了web前端的编写技术。

### 1. 数据库的效率与实际业务需求交织在一起

在这个过程中我第一次从实际应用的角度来设计数据库，编写sql语句，一方面要考虑数据库的设计，从技术角度思考“好的”数据库的设计应该是怎样的，另一方面还要结合实际的功能需求，由需求来驱动设计。这是比较考验设计能力的地方。

比如，从功能出发，按照馆藏位置来查询库存，开始我从功能出发，只使用了一个select...from...where 语句，之后又想到可以为这三个馆藏建立视图，利于效率的替身，进而又想到可能在功能上授权三个馆藏的管理员只看到本馆藏的库存，但由于时间限制，不同角色看不同视图的功能并没有实现。再比如，我还遇到了先设计了一些对于数据库来说比较高效的内容，但是后来发现功能需求实际上并不需要这些东西。

虽然按照助教给的图书馆项目pg做了初步的需求分析，但是在具体实施过程中，还会出现很多的变化，我考虑到原因是没有事先把业务需求转为系统的需求，没有明确地把需求表达出来，结果系统地功能随着代码的编写而游走（这在工业中可能导致很严重的软件开发的失败）。

### 2. 数据库的设计

对于这个系统而言，具体业务功能的需求多半和某些过程、函数、触发器等相关，是比较明确、自由度小的工作。但数据库的设计实际上在整个pg完成过程中我花费了近一半的时间。

我先初步画了er图，然后将er图映射为表，其中我的部分（库存、采编和财务管理）我把所有的关系都用属性的形式映射到表中，并没有关系表。如采编表和进货表，多个采编项对应一个进货项，此时把进货项的id放到采编项的属性当中。

这些设计并不是一蹴而就的，过程中我删除了、新增了很多表，修改了很多属性，最后一版的表已经把最初的表改得面目全非了。

### 3. sql语言

不同的系统都对sql标准有一些细小的改动，不像python，c++这些语言是非常固定的，所以sql标准、postgresql、mysql的这些语言在我脑子里交织在一起，常常出现“syntax error”

### 4. 程序的脆弱性

这个问题不仅仅是良好的代码风格就能解决的问题，我认为这个问题要从系统的设计来解决，系统的架构、结构、设计风格都为一个“好的”系统做出重要的支撑。我的代码还存在很多冗余的问题，尤其是html，即使使用了模板的继承也还有很多不够“dry”的代码。前后端的交互也存在很多问题，我曾想过所有的查询功能，全部封装成一个python函数，但这似乎并不是一个“高内聚”的方式，按功能来实现模块的封装似乎更加优良，但这又涉及更多的设计问题，我的代码还需要再优化。



[REDACTED]

[REDACTED]

[REDACTED]