

图书管理系统数据库设计实验报告

陈俊驰 PB18000051

一、实验内容及要求：

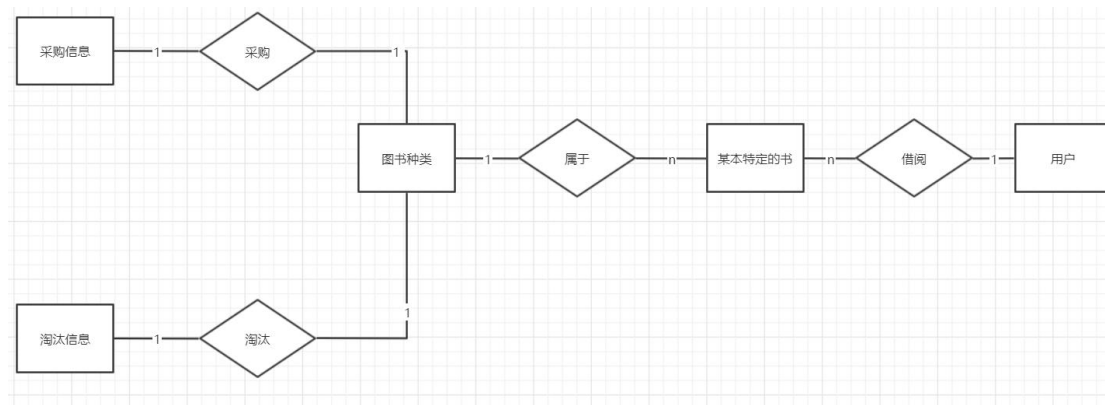
编写一个图书管理系统，要求具有以下内容：

- 1) 存储图书信息、采购和淘汰情况、租借情况
- 2) 实现图书采购、淘汰、租借功能
- 3) 实现图书信息、采购和淘汰、库存、和租借情况查询
- 4) 实现图书的采购、库存、淘汰、租借情况等统计

二、实验设计：

1.数据库设计：

首先，进行需求分析，根据实验内容的要求，进行逻辑设计获得概念模型，抽象并精简出获得五个实体，分别为图书种类、特定某本书、采购信息、淘汰信息以及用户。对于图书种类和特定某本书的关系举例来理解，图书馆有 3 本《算法导论》，《算法导论》即为图书种类，其中的一本为特定某本书。然后根据这五个实体之间的关系，构造他们的 E-R 图，这里省略了实体的属性。



然后，利用各项实体、联系及其属性，进行逻辑结构设计，将已获得的概念模型转换为如下的数据模型，并利用 SQL 语言构建每一个关系模式对应的表。其中，由于各联系均不是 m:n 类型，故将联系的模式都并入了实体

1) 图书种类 (ISBN, 索书号, 书名, 类型, 作者, 出版社, 出版年份, 价格, 库存数量, 剩余数量)，用于存储某种图书的信息，例如《三体》这种书的各类信息，具体的 SQL 代码如下

```
create table PB18000051book(  
    ISBN varchar(20) not null unique,  
    索书号 varchar(20) not null unique,  
    书名 varchar(20) not null,  
    类型 varchar(20) not null,  
    作者 varchar(20) not null,  
    出版社 varchar(20) not null,  
    出版年份 int not null,  
    价格 float not null,  
    库存数量 int not null,  
    剩余数量 int not null constraint book51_check1 check (剩余数量>=0),  
    constraint book51_pk primary key (ISBN),  
    constraint book51_check2 check (库存数量>=剩余数量)  
)
```

其中，码为 **IBSN**，能唯一决定某种图书的信息，索书号根据查询得知为图书馆为某个 **IBSN** 指定的唯一编号，不可重复，故加入 **unique** 限制。库存数量表示该图书馆中总共馆藏的该书的本数，包括在馆可借的以及已被借出的，剩余数量表示该图书馆中可借的数量，显然，有库存数量 \geq 剩余数量 ≥ 0 ，加入该约束条件。此外，类型和图书种类并不是统一概念，例如《三体》是一种书，而其类型为文学类。

2) 用户（学工号，姓名，已借数目，密码，级别），用于存储用户的信息，具体的 SQL 代码如下

```
create table PB18000051student(  
    学工号 varchar(12) not null unique,  
    姓名 varchar(10) not null,  
    已借数目 int not null constraint stu51_chel check(已借数目 $\geq 0$ ),  
    密码 varchar(20) not null,  
    级别 varchar(10) not null constraint stu51_che2 check(级别 in ('管理员', '普通用户')),  
    constraint stu51_pk primary key(学工号)  
)
```

其中，学工号为码，可以唯一确定一个用户（包括管理员）。为对该数据库进行权限限制，故将用户的级别划分为管理员与普通用户，其相应权限将在之后的权限设计详述。为了保障数据库的安全性，加入密码以方便之后的应用程序进行用户标识与鉴定。此外，加入约束限制用户的已借数目不小于 0。

3) 特定某本书（条码号，**IBSN**，学工号，借阅时间，应还时间，是否归还），用于存储某一本书特定的书的信息，例如图书馆有若干本《三体》，其中的一本叫做某一本书特定的书，具体的 SQL 代码如下

```
create table PB18000051specificbook(  
    IBSN varchar(20) not null,  
    条码号 varchar(10) not null unique,  
    学工号 varchar(12),  
    借阅时间 datetime,  
    应还时间 datetime,  
    是否归还 varchar(3) constraint spec51_check1 check(是否归还 in ('是', '否')),  
    constraint spec51_pk primary key(条码号),  
    constraint spec51_fk1 foreign key(IBSN) references PB18000051book(IBSN),  
    constraint spec51_fk2 foreign key(学工号) references PB18000051student(学工号)  
)
```

其中，条码号为码，条码号是每一本书的“身份证”，标识了这是哪一种的哪一本书，条码号将根据图书馆历史上所拥有过的书籍数目顺序生成，不需手动设定，可以理解为图书馆历史上的第 7 本书条码号就是 0000000007。**IBSN** 和学工号为外码，分别参照图书种类和用户两张表。对于该图书是否在馆，通过是否归还来标识，“否”表明借出。应还时间在这里设定为借阅时间加 30 天，将在应用程序的借阅功能体现。

4) 采购信息（采购号，**IBSN**，日期，采购数量，单价，采购原因），用于存储采购信息，其 SQL 代码如下

```
create table PB18000051buy(  
    采购号 varchar(15) not null unique,  
    IBSN varchar(20) not null,  
    日期 datetime not null,  
    采购数量 int not null constraint buy51_chel check(采购数量 $\geq 1$ ),  
    单价 float not null constraint buy51_che2 check(单价 $\geq 0$ ),  
    采购原因 varchar(100),  
    constraint buy51_pk primary key(采购号),  
    constraint buy51_fk foreign key(IBSN) references PB18000051book(IBSN)  
)
```

其中，码为采购号，唯一的决定每个采购信息的录入，采购号由程序自动按序生成。为了保证采购信息的有效性，对采购数量加入了不小于 1 的约束。为了

避免每次采购都录入一遍图书的信息，我将 **IBSN** 设为外码，参考图书种类表。如果是图书馆已有的种类，则直接录入采购信息，如果是图书馆没有的书种，则在应用程序中先录入该图书的信息，然后在录入采购信息。

5) 淘汰信息（淘汰号，**IBSN**，条码号，日期，淘汰原因），用于**存储淘汰信息**，其 SQL 代码如下

```
create table PB18000051elim(  
    淘汰号 varchar(15) not null unique,  
    IBSN varchar(20) not null,  
    条码号 varchar(10) not null unique,  
    日期 datetime not null,  
    淘汰原因 varchar(100),  
    constraint elim51_pk primary key(淘汰号),  
    constraint elim51_fk foreign key(IBSN) references PB18000051book(IBSN)  
)
```

其中，码为淘汰号，自动顺序生成。外码为 **IBSN**，用来保证图书馆存在这个条码号及其对应的书，在应用程序中，将会先从特定某本书表中查询条码号并获得 **IBSN**，用来避免不存在的条码号。

在创建各表之后，输入一些数据进行初始化。

2. 功能设计

根据实验内容的要求，上述的数据库设计完成了存储信息的功能，然后将在应用程序中具体实现操作、查询、统计三类功能。对于每个功能，所包含如下的不同权限：

操作：借阅、归还、采购、淘汰、查询个人未还书籍、录入图书数据；

查询：图书信息、借阅情况、用户/个人信息、采购信息、淘汰信息、图书搜索；

统计：借阅情况统计、近期采购情况统计、近期淘汰情况统计、借出书籍统计。

每一个权限的作用将在之后的应用程序功能介绍部分详述。

3. 权限设计

根据图书管理系统所面向的对象，我在该系统中将用户分为“普通用户”、“管理员”两种权限组，分别拥有不同的权限。此外，还有数据库管理员及系统设计人员，但由于可以直接接触到 SQL 语言本身，就不额外为他们添加权限级别了。作为“管理员”，拥有该系统的最高权限，拥有操作、查询、统计中的所有权限，可以直接看到所设计的全部表格并对数据库录入信息。在这里，我认为管理员仅拥有一个学工号和密码，故其也可以进行借阅归还操作。

作为用户，不应该看到具体的采购和淘汰数据、进行采购淘汰录入图书数据的操作，也不应该获得所有人的用户信息和借阅情况，故需对其权限进行设置，仅保留如下的权限：

操作：借阅、归还、查询；

查询：图书信息、个人信息、图书搜索；

统计：借阅情况统计、近期采购情况统计、近期淘汰情况统计。

对于不同权限的用户的区分，将利用所实现应用程序的登录页面进行区分，同时也实现了用户的鉴定。

如上，完成了实验的设计部分，设计了数据库、操作、权限划分。

三、具体实现：

根据上述的设计，首先在老师所提供的服务器 202.38.88.99,1434 上，通过 SQL 查询器，利用所给出的 SQL 代码建立五个表，并输入一些数据作为对表所进行的初始化。

然后，进行应用程序的实现。这里为了实现所要求的具有 GUI 的应用程序，选择了老师所推荐的 Delphi7 语言，Delphi 本身自带图形化界面的开发与展示，并且直接提供链接数据库支持 SQL 语言的组件，相较于其他的语言，开发更为简单快捷。应用程序开发的核心思想就是利用 ADOConnection 连接到数据库以后，利用对于 ADOQuery 组件的 SQL 语句的修改、运行、获得结果，实现设计中的各项功能。具体代码见附件，在此不过多展示。

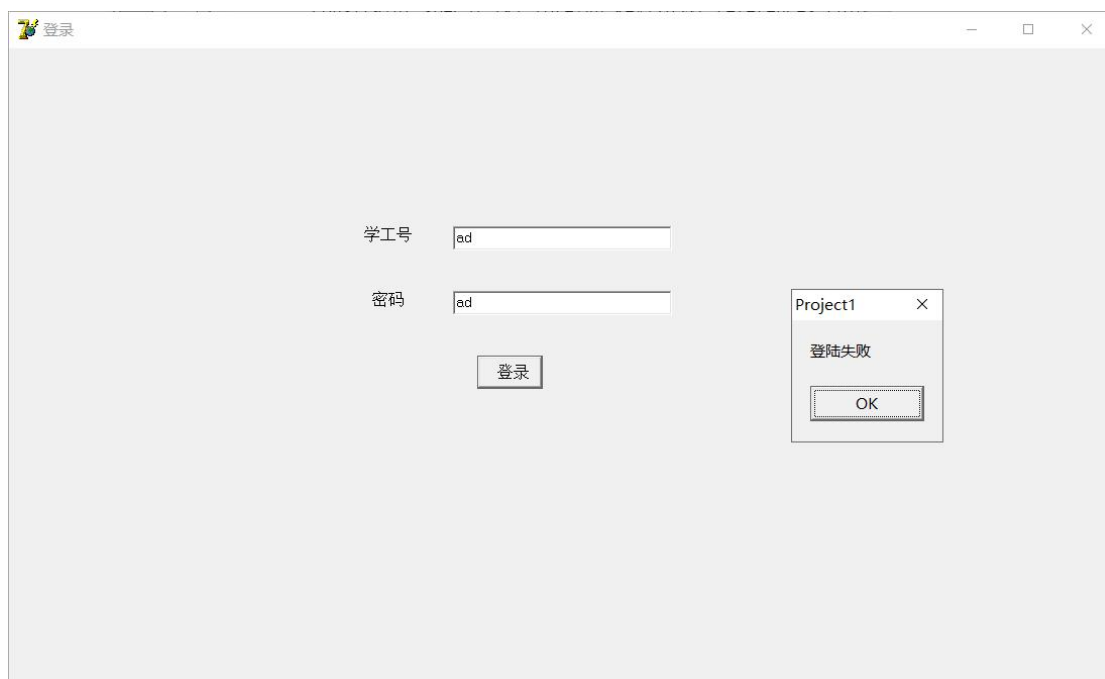
此外，为了区分不同权限组，设计了 3 个窗体，分别为 LoginForm，作为登录界面，然后将用户转到管理员界面或普通用户界面；ManagerForm，作为管理员界面；UserForm，作为普通用户界面。

四、功能展示：

运行该应用程序后，将会首先进入到下图所示的登录界面。

可以在学工号和密码所标识的文本栏中输入学工号和密码，并按下登录按钮。此时，会触发登录按钮的过程，从数据库用户表中查询是否有用户的学工号和密码符合输入。

如果无，则弹出登录失败

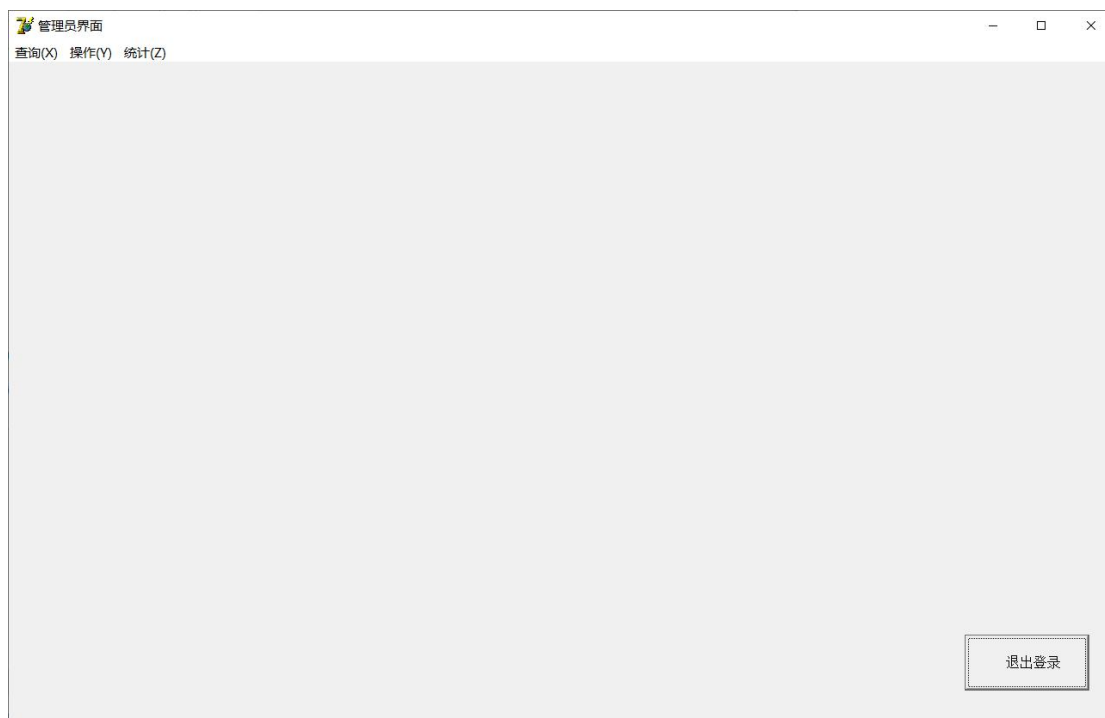


如果有，则根据查询到结果用户级别不同，登录界面会消失，然后显示管理员界面和普通用户界面中的一个。这里让登录界面消失，并不会清除其学生号文本框中的数据，一遍为之后借还等操作提供用户学工号。

然后分别介绍管理员界面和用户界面

1. 管理员界面

利用管理员账号 **AB12345678** 及密码 **admin** 登录，进入管理员页面如下图



选择主菜单**查询**按钮后，会显示出作为管理员所允许进行的查询，包括图书信息、借阅情况、用户信息、采购信息、淘汰信息、图书搜索。其中除了图书搜索外，每一个查询都有对于查询结果根据一定顺序进行排序的功能，在你选中

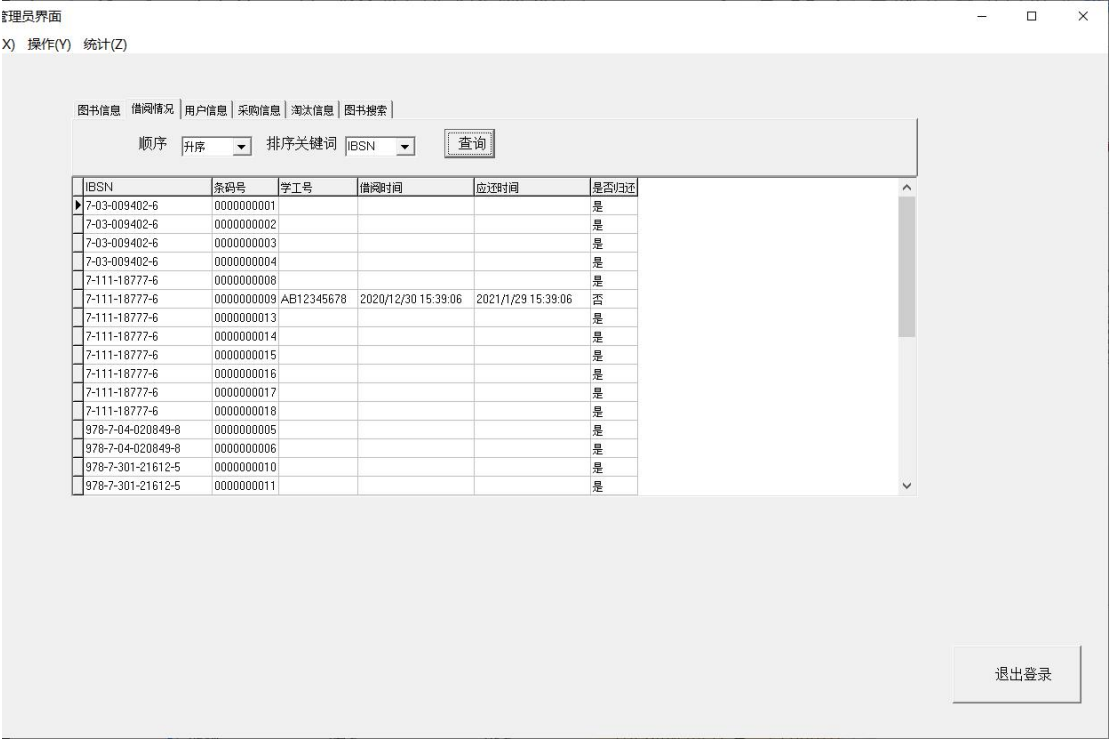
需要进行的查询、顺序以及排序关键词后，点击查询按钮，就会在表格中显示结果。这些查询的实现思想主要在于利用按钮触发，向 ADOQuery 中插入一个根据所选择的查询项目及下拉栏得到文本所构成的 select 语句，获得结果后，将查询结果输出到表格上。

对于**图书信息**的查询，可以看到馆藏图书的相关信息，查看图书种类表。图中可以看到，算法导论库存数量比剩余数量多 1 本，说明有一本书被借走了。

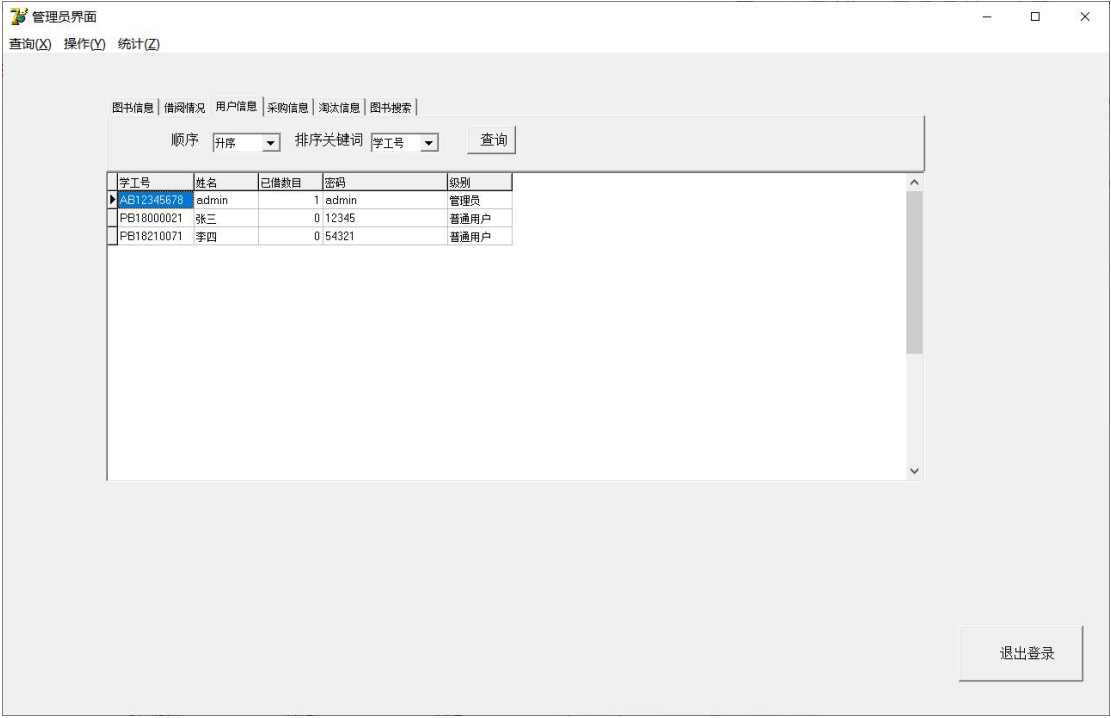


对于**借阅情况**的查询，可以获得每一本未被淘汰的书当前状态、条码号与 ISBN 的对应、借阅情况，查看特定某本书表。可以看到 0000000009 号书被

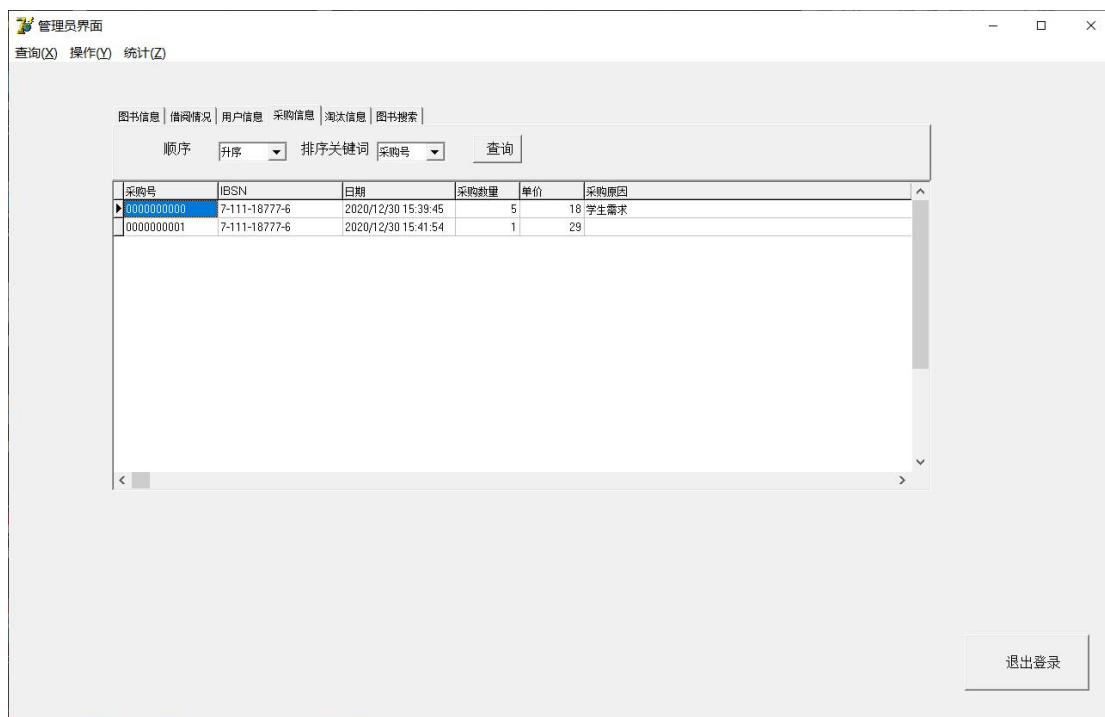
AB12345678 用户在 12 月 30 日节奏，尚未归还。查看 IBSN 知，正式上面图书信息中得知被节奏的算法导论



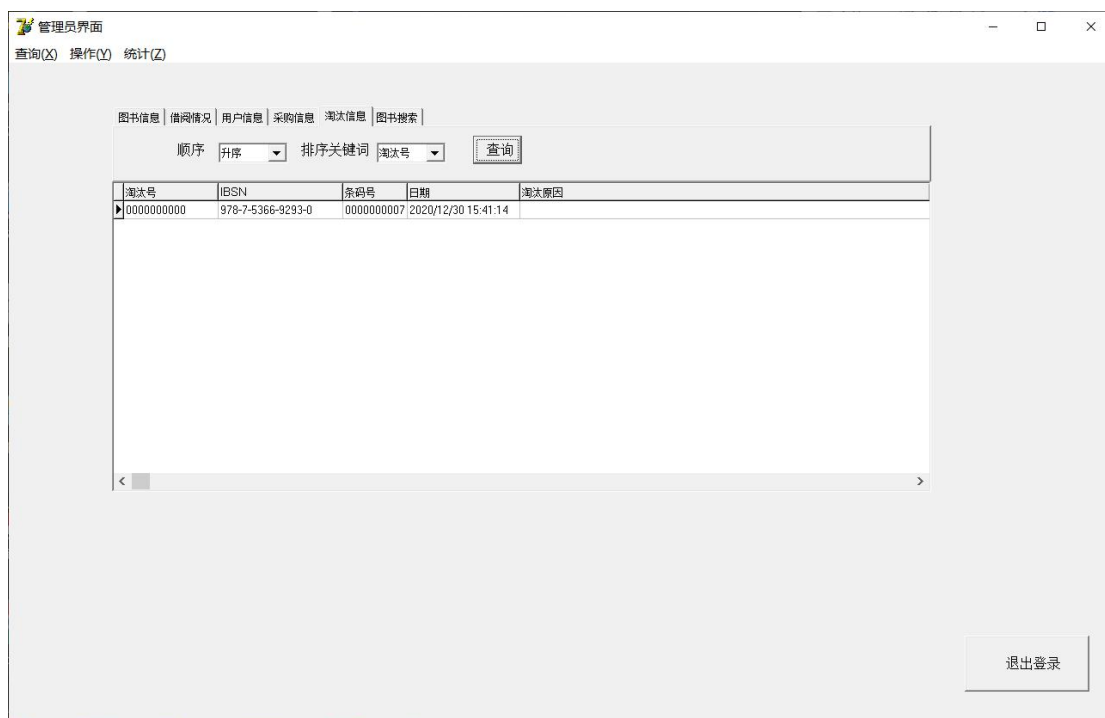
对于用户信息的查询，可以获得所有用户的信息，查看用户表。可以看出，作为管理员 admin 借走了一本书，在这里显示密码确实有不安全因素，但我的想法是在用户忘记密码后，可以从管理员这里查得。



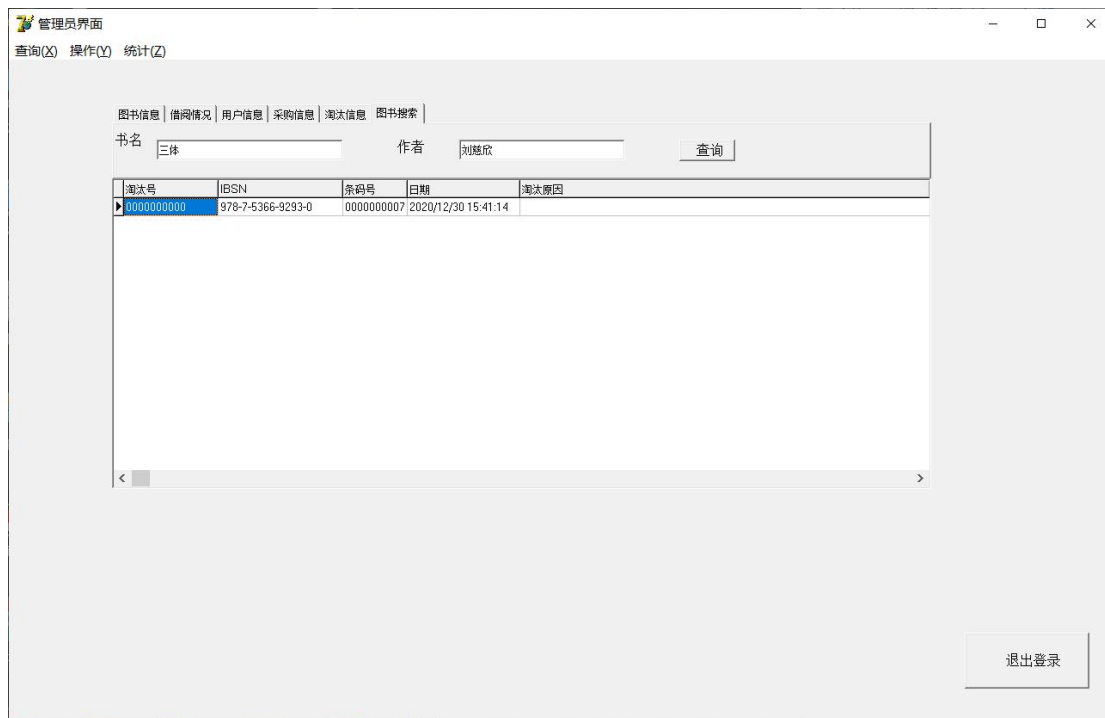
对于采购信息的查询，可以获得所有采购的具体信息，查看采购信息表。显示在 12 月 30 号对 7-111-18777-6IBSN 的书以每本 18 元的价格采购了 5 本，原因是学生需求。



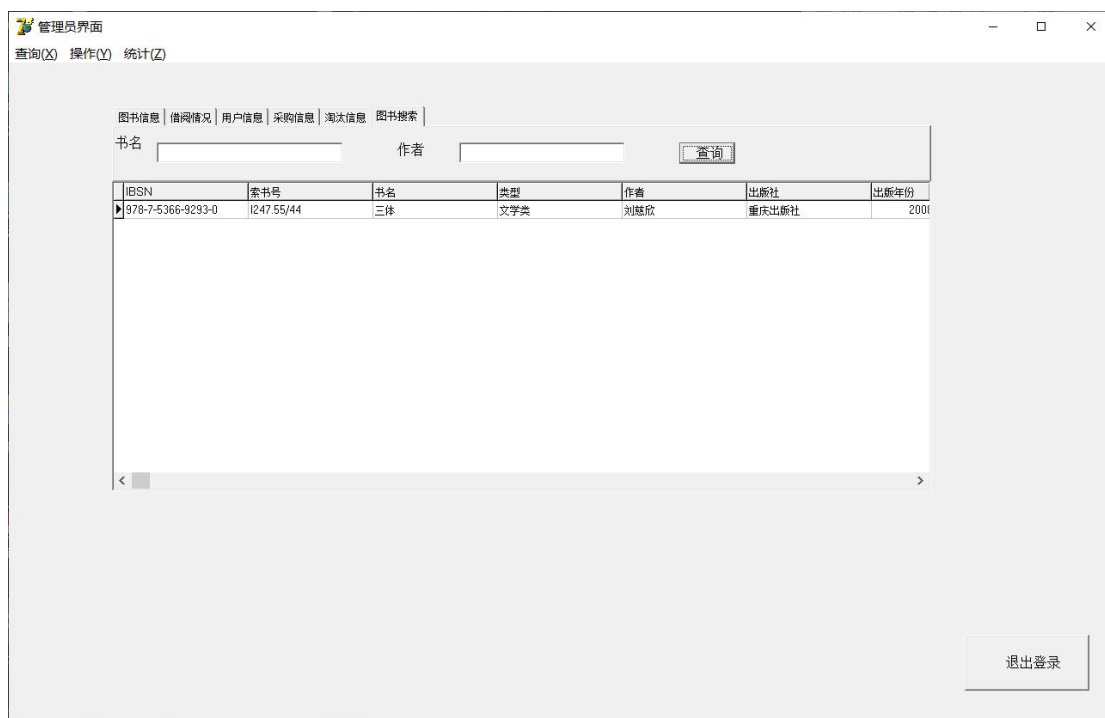
对于**淘汰信息**的查询，可以获得所有淘汰的具体信息，查看淘汰信息表。可以看出有个条码号为 0000000007，ISBN 为 978-7-5366-9293-0 的书被淘汰，未备注淘汰原因，恰好与借阅情况中条码号 0000000007 不存在相符。



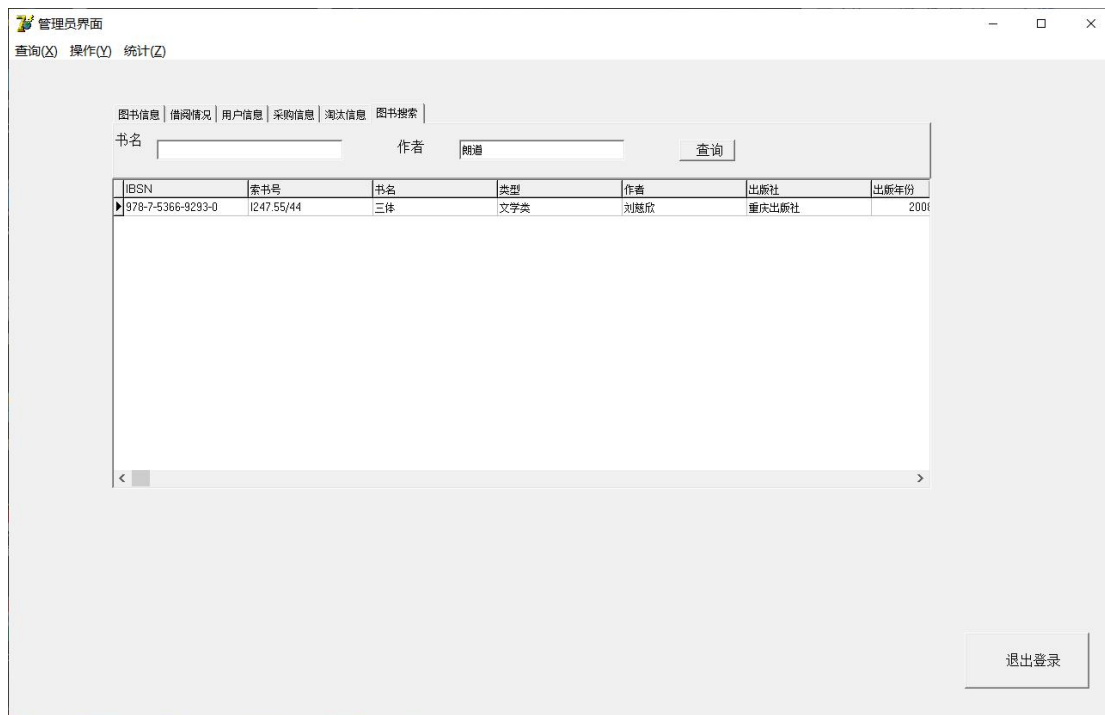
对于**图书搜索**，可以根据你在作者与书名文本框中输入的文本，获得具体该本书的信息。其支持对于书名和作者兼有的搜索，输入搜索条件如下图



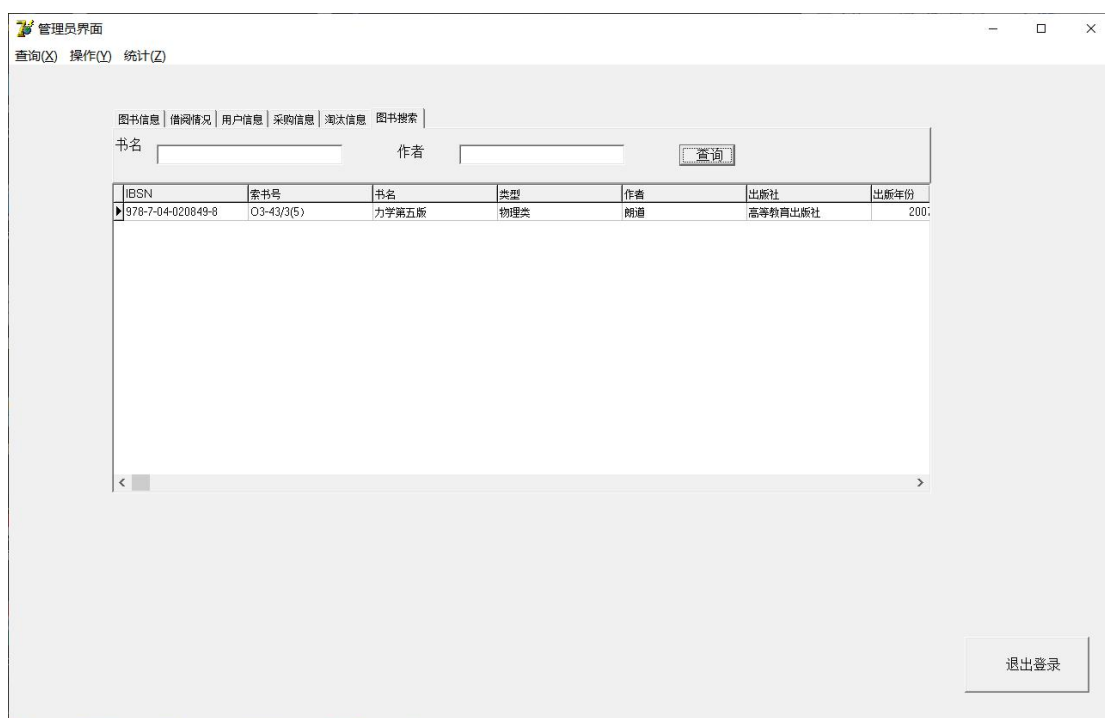
搜索结果为，正确搜索出书名为三体，作者为刘慈欣的图书信息



同时也可以仅对书名或作者名进行搜索，例如输入朗道



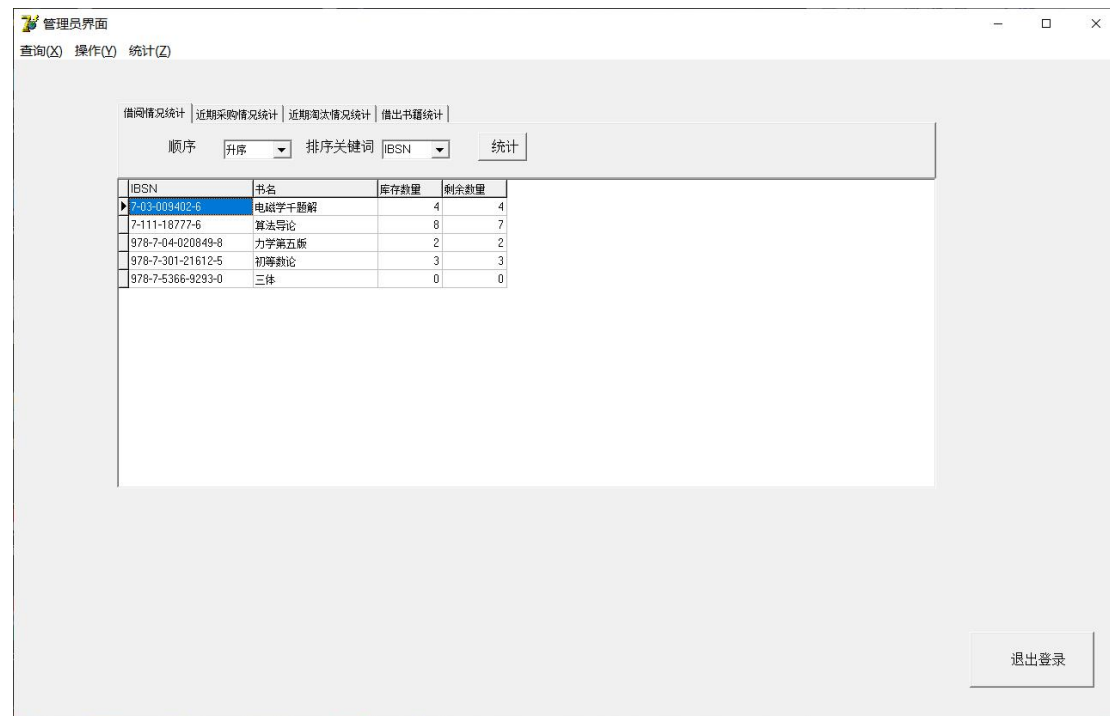
搜索结果为，结果正确



上述的图书搜索功能与其他的信息查询功能实现的思路类似，对于图书种类表进行查找，不过额外增加 **where** 条件判断部分，在 **where** 部分分别判断图书信息表中的属性“书名”、“作者”是否等于文本框输入。此外，还增加了对于书名和作者文本栏是否为空的判断，以决定 **select** 语句的 **where** 条件中是否会出现 **and** 连接两个条件。

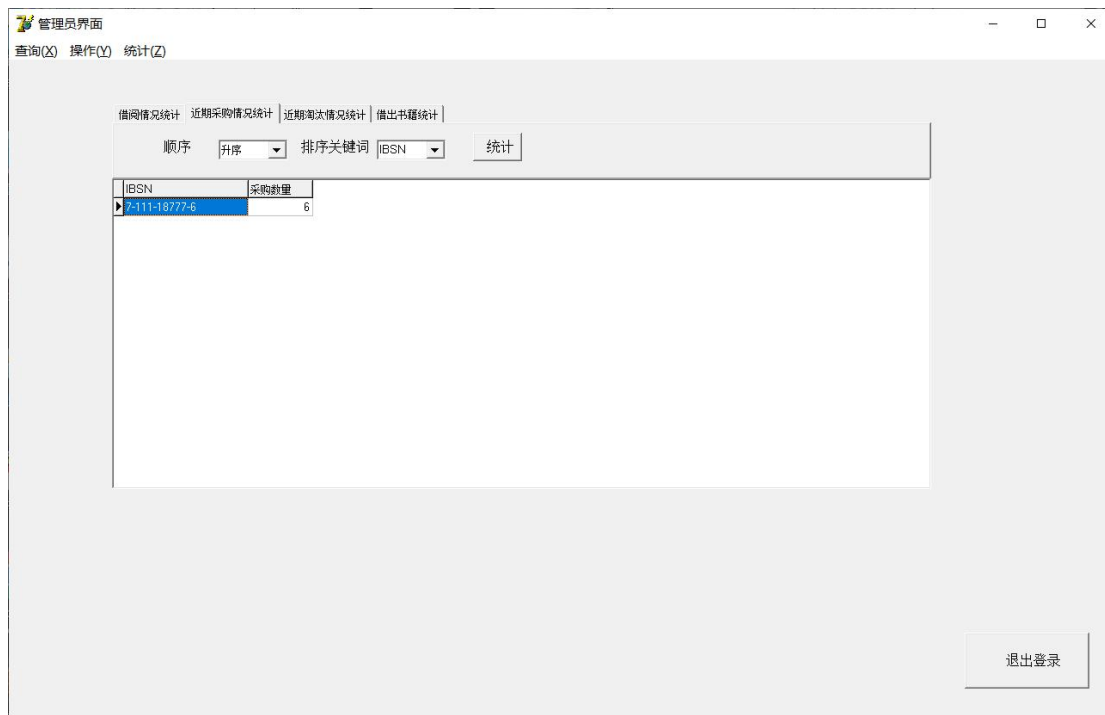
选择主菜单**统计**按钮后，会显示出作为管理员所允许进行的统计，包括借阅情况统计、近期采购情况统计、近期淘汰情况统计、借出书籍统计。这些功能实现的主要方法是根据你所选的统计以及排序，构造一个含有 **where**、**group**、**order** 等的部分选择语句，然后利用 **ADOQuery** 将结果反馈。

对于**借阅情况统计**，结果如下，可以看出每种书的库存数量（总共数量）和剩余数量（未借出的数量）。这些信息均在图书信息表格中维护，故实现的核心在于利用 **select** 语句，仅对 **IBSN**、书名、库存数量、剩余数量进行查找。

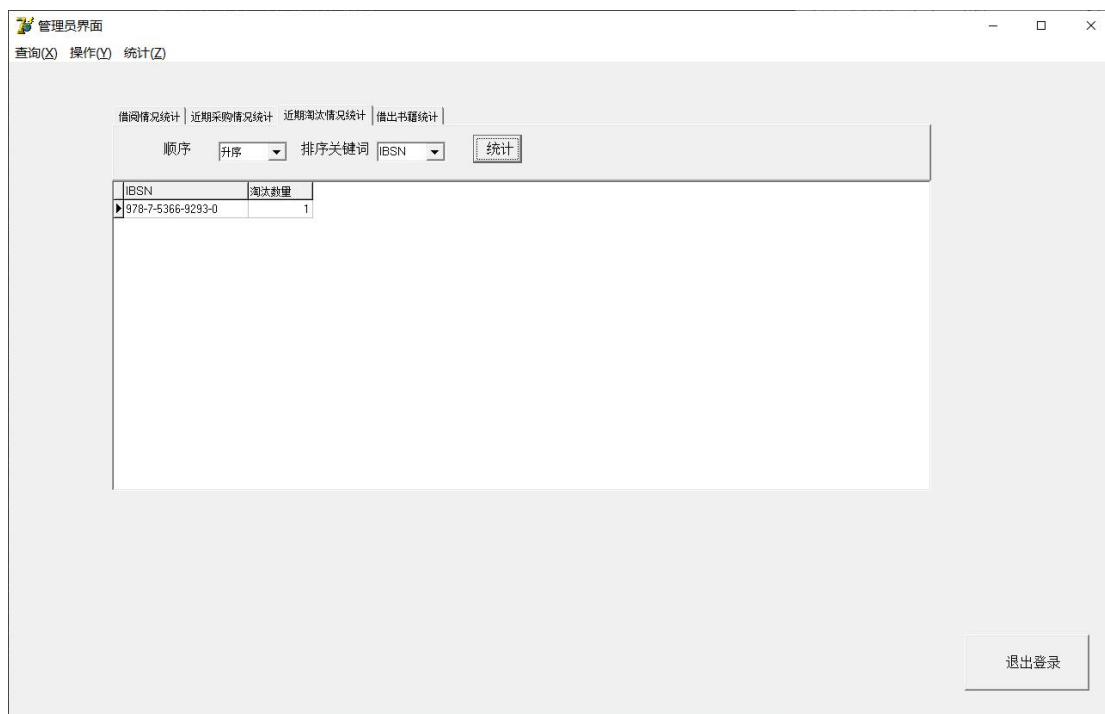


IBSN	书名	库存数量	剩余数量
7-03-009402-6	电磁学千题解	4	4
7-111-18777-6	算法导论	8	7
978-7-04-020849-8	力学第五版	2	2
978-7-301-21612-5	初等数论	3	3
978-7-5366-9293-0	三体	0	0

对于**近期采购情况的统计**，结果如下，可以看出每一种被采购过的书（仅显示 **IBSN**）及其近期采购的总数量。例如对于 **7-111-18777-6**，近期总共采购了 **6** 本，而我们在先前的采购信息查询中得知该书两次采购，一次 **5** 本，一次 **1** 本，合起来正好为 **6** 本。该统计主要实现的思路在于利用 **group by**，通过 **IBSN** 进行分组，并利用 **sum**（采购数量）对于采购数量进行求和，从而构造一个 **select** 语句获得结果，忽略 **order** 部分，代码为“**select IBSN,采购数量=sum（采购数量）from PB18000051buy group by IBSN**”

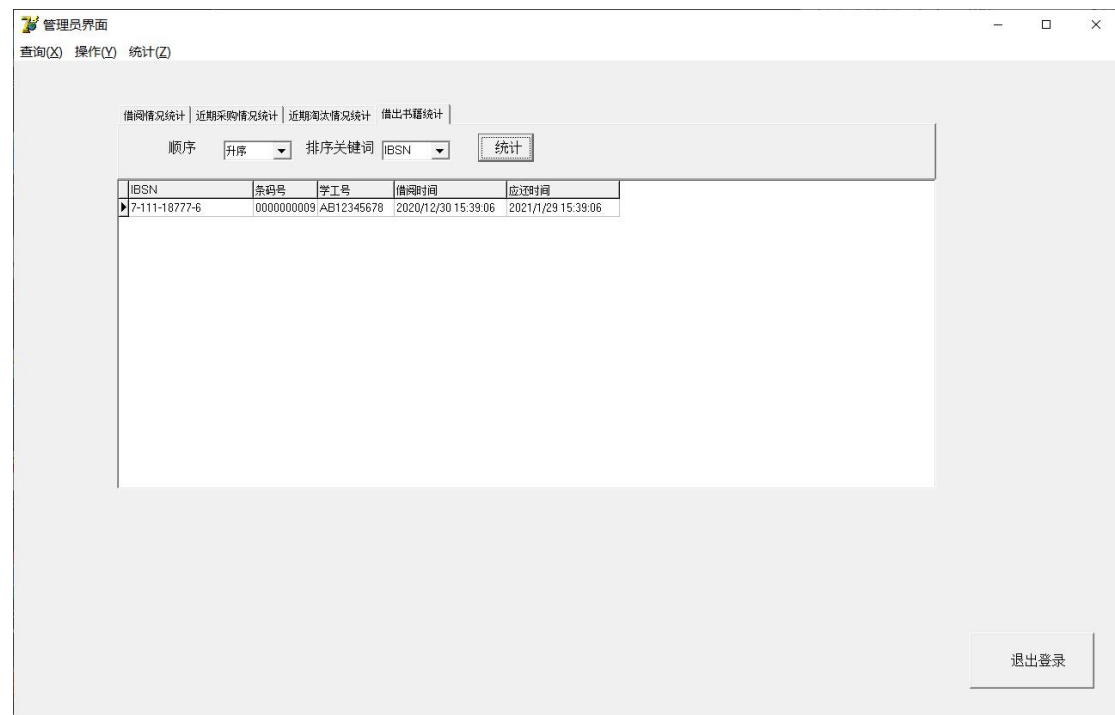


对于**近期淘汰情况统计**，结果如下，可以看出每一种被淘汰过的书（仅显示 ISBN）及其被淘汰的总量。其中 978-7-5366-9293-0 近期被淘汰了 1 本，与先前查询淘汰信息符合。具体的实现思路与近期采购情况实现思路相同，不过由于每一条淘汰信息对应一本书，故利用 `count(*)` 结合 `group by ISBN` 进行 `select` 查询，通过 ADOQuery 返回结果。



对于**借出书籍统计**，结果如下，显示出已经被借出的书籍条码号及相关的借阅操作信息，包括借阅人、借阅时间等。显示 AB12345678 借走了条码号为 000000009 的书，与查询信息相符。实现该功能的核心思想在于条件查询，利用

我们特别在特定某本书的表中加入的是否归还属性进行含 **where** 的查找，条件为“是否归还=‘否’”，用来查找那些未被归还的书。（后期写报告的时候，注意到其实是否归还可以不使用，直接利用学工号是否为空即可，之前设计的时候忘了这层逻辑推理了）



选择主菜单的**操作**按钮后，会显示管理员所允许进行的各种操作，包括借阅、归还、采购、淘汰、查询个人未还书籍、录入图书数据等操作。这些操作的主要实现方式在于利用已有的信息，通过 **update**、**delete**、**insert** 等语句对于表进行增删改。

为了方便借阅和归还的展示，首先介绍**查询个人未还书籍**，选择个人未还数据并点击查询后，显示当前使用该系统的用户所未归还的书籍，结果如下。结果与先前查询等途径获得的信息相符合。其实现思路为利用登录界面文本框中的学工号进行对于特定某本书表进行条件查询，查询对应学工号的结果

管理员界面

查询(X) 操作(Y) 统计(Z)

借阅 | 归还 | 采购 | 淘汰 | 查询个人未还书籍 | 录入图书数据

查询

ISBN	条码号	学工号	借阅时间	应还时间	是否归还
7-111-18777-6	0000000009	AB12345678	2020/12/30 15:39:06	2021/1/29 15:39:06	否

退出登录

对于**借阅**，我们仅需输入所要借阅的书的条码号，就可自动对于各个表进行修改，从而成功借阅。输入条码号为 0000000005 后，借阅成功，结果如下。具体的实现思路为，利用 ADOQuery 的 `execsql` 执行，首先利用该条码号从特定某本书中找到该条码号所对应的 ISBN，并利用 `Update` 语句将图书种类表中该 ISBN 对应的剩余数量减 1，表示有一本已经被借走，从而实现对于图书种类表的更新。然后利用在已经被隐藏的登录界面中未被消去的学工号，利用 `Update` 语句，将该条码号的学工号赋值，令借阅时间为 `getdate()` 获得的当前时间、归还时间为借阅时间利用 `dateadd` 函数加 30 天、是否归还为否，从而实现对于特定某本书的更新。最后利用 `Update` 语句及获得的学工号，将用户表该用户的借阅数量加 1，表明该用户借了一本书，从而实现对于用户表的更新。

管理员界面

查询(X) 操作(Y) 统计(Z)

借阅 | 归还 | 采购 | 淘汰 | 查询个人未还书籍 | 录入图书数据

请输入您的条码号，确认准确无误后，按下借阅键

条码号

借阅

Project1

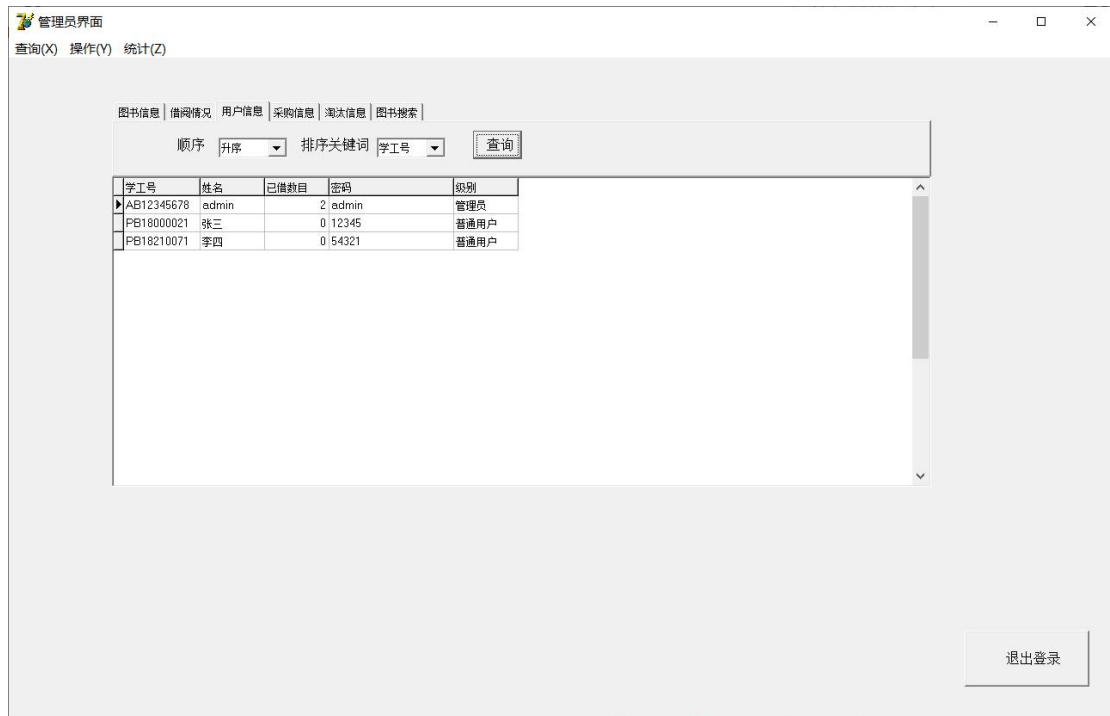
借阅成功

OK

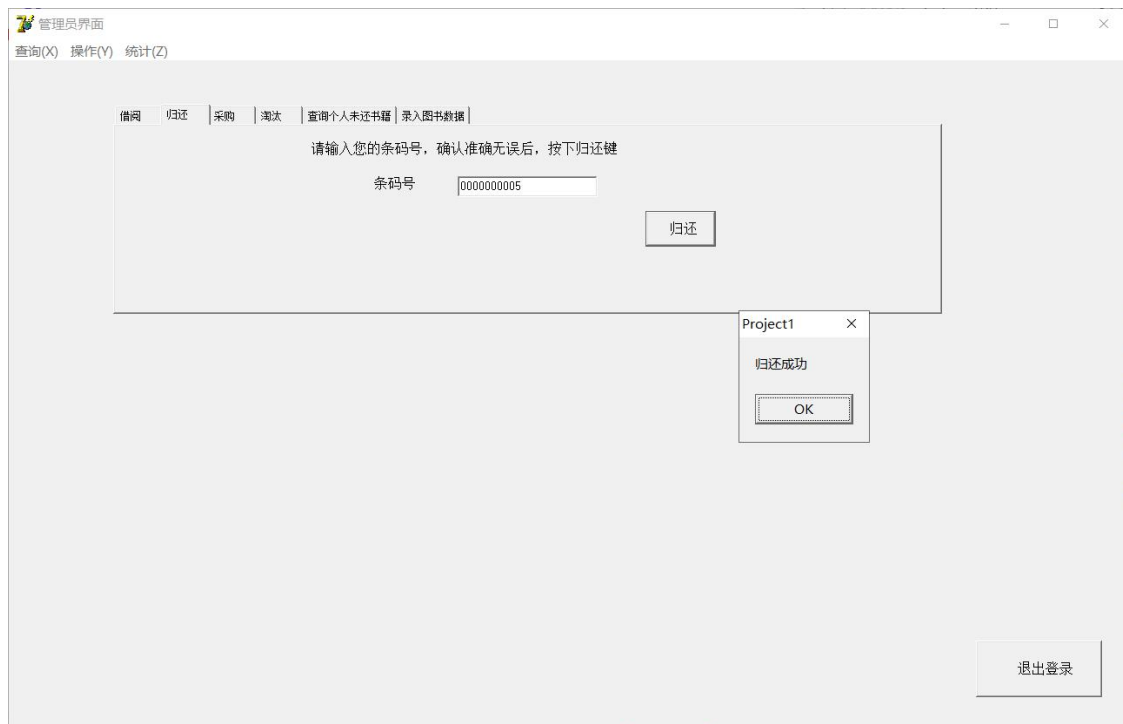
退出登录

此时，查询个人未还书籍，发现已经多了 0000000005 条码号对应的借阅信息，查询图书信息发现剩余的力学减少了 1 本，查询用户信息发现 AB12345678 对应的已借数目也加 1，均说明借阅成功。

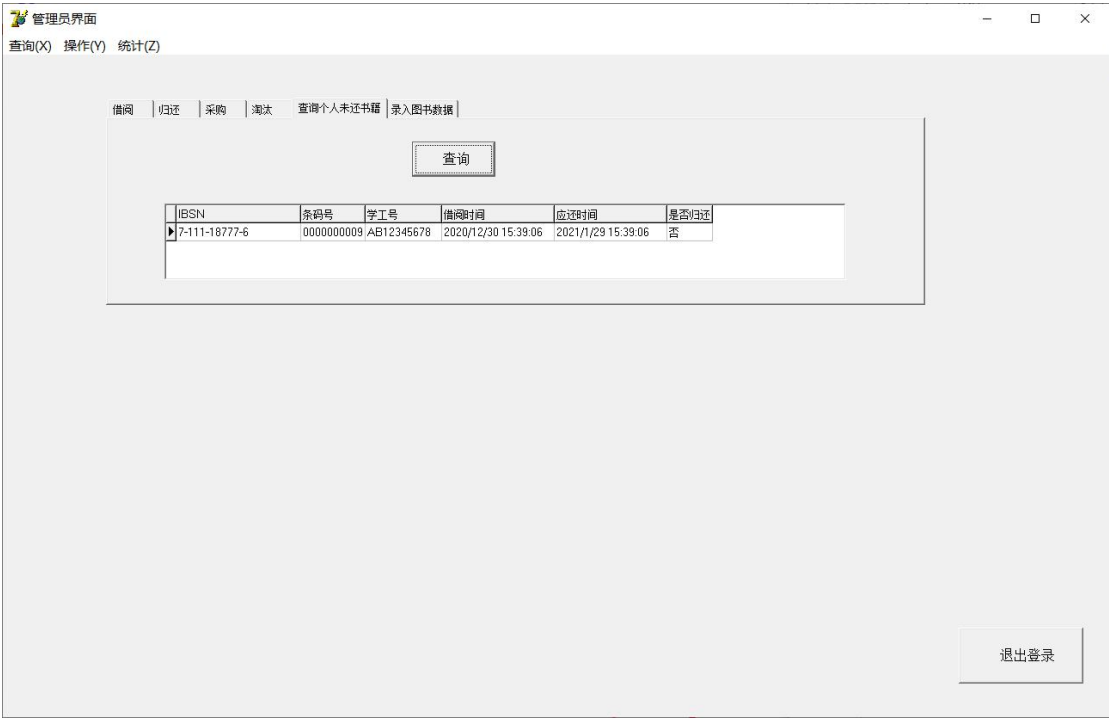




对于归还，我们仅需要输入所要归还的图书的条码号，将自动对各个表进行修改，从而自动实现归还的操作。输入条码号为 0000000005 后，归还成功，结果如下。具体的实现思路为，利用 ADOQuery 的 `execsql` 执行，首先利用该条码号从特定某本书中找到该条码号所对应的 IBSN，并利用 `Update` 语句将图书种类表中该 IBSN 对应的剩余数量减 1，表示有一本已经被归还，从而实现对于图书种类表的更新。然后利用 `Update` 语句，将该条码号的学工号、借阅时间、归还时间置为空，并将是否归还置为是，从而实现对于特定某本书表的更新。最后利用 `Update` 语句及获得的学工号，将用户表该用户的借阅数量减 1，表明该用户换了一本书，从而实现对于用户表的更新。



此时，查询个人未还书籍，发现刚才出现的 0000000005 条码号对应信息已经消失，查询图书信息发现对应的力学剩余数量相较于刚才加 1，查询用户信息发现已借数目减 1，均说明归还成功。



管理员界面

查询(X) 操作(Y) 统计(Z)

图书信息 | 借阅情况 | 用户信息 | 采购信息 | 淘汰信息 | 图书搜索 |

顺序 升序 排序关键词 学工号 查询

学工号	姓名	已借数目	密码	级别
AB12345678	admin	1	admin	管理员
PB18000021	张三	0	12345	普通用户
PB18210071	李四	0	54321	普通用户

退出登录

由于采购信息表 IBSN 作为外码，参考图书种类表，故如果进行采购时所采购的是没有在图书馆中储藏过的书籍，应当首先录入图书数据。

对于**录入图书数据**，我们需要输入一种图书的相关信息，然后点击录入图书数据的按钮，就可以将这种图书的相关信息插入图书种类表，输入及结果如下图。主要实现方式就是利用 insert 语句及各文本栏的数据构造一个图书种类表的元组及插入语句，然后利用 ADOQuery 的 execsql 插入图书种类表。其中，对于库存数量和剩余数量这两个属性的值不需要从文本栏获得，由于此时仅录入图书数据，故均置为 0。

管理员界面

查询(X) 操作(Y) 统计(Z)

借阅 | 归还 | 采购 | 淘汰 | 查询个人未还书籍 | 录入图书数据 |

IBSN 978-7-5021-7505-4 作者 苏建军

索书号 K826.16-49/1 出版社 石油工业出版社

书名 钱学森人生故事全集 出版年份 2010

类型 传记类 价格 24.8

录入图书数据

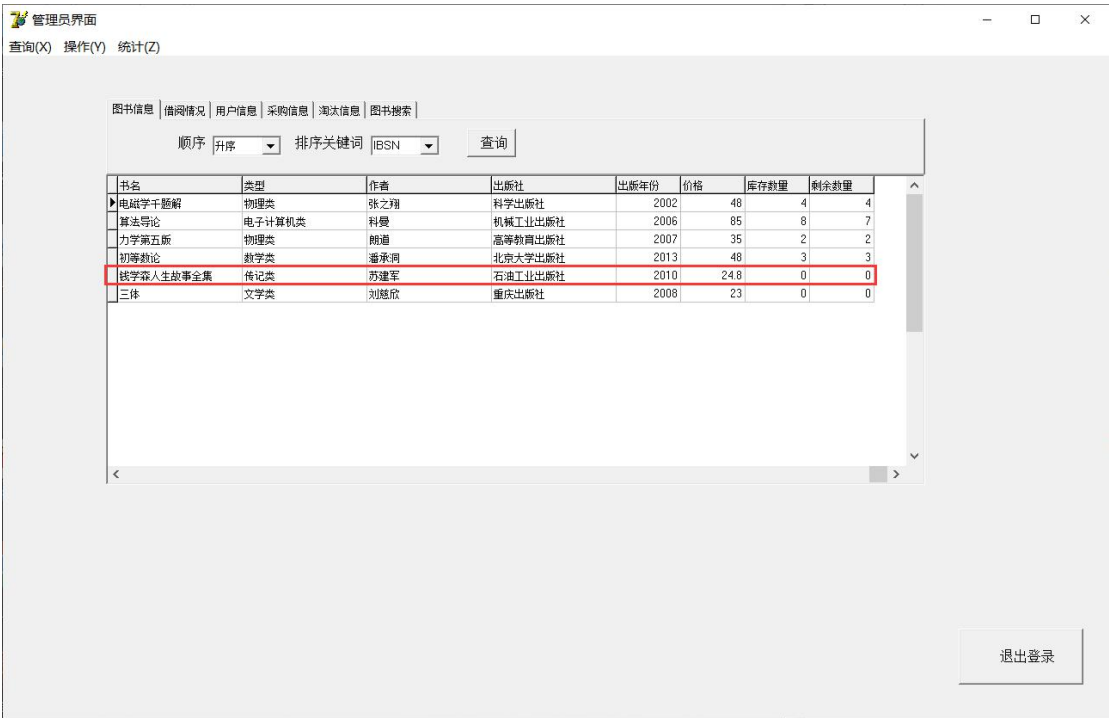
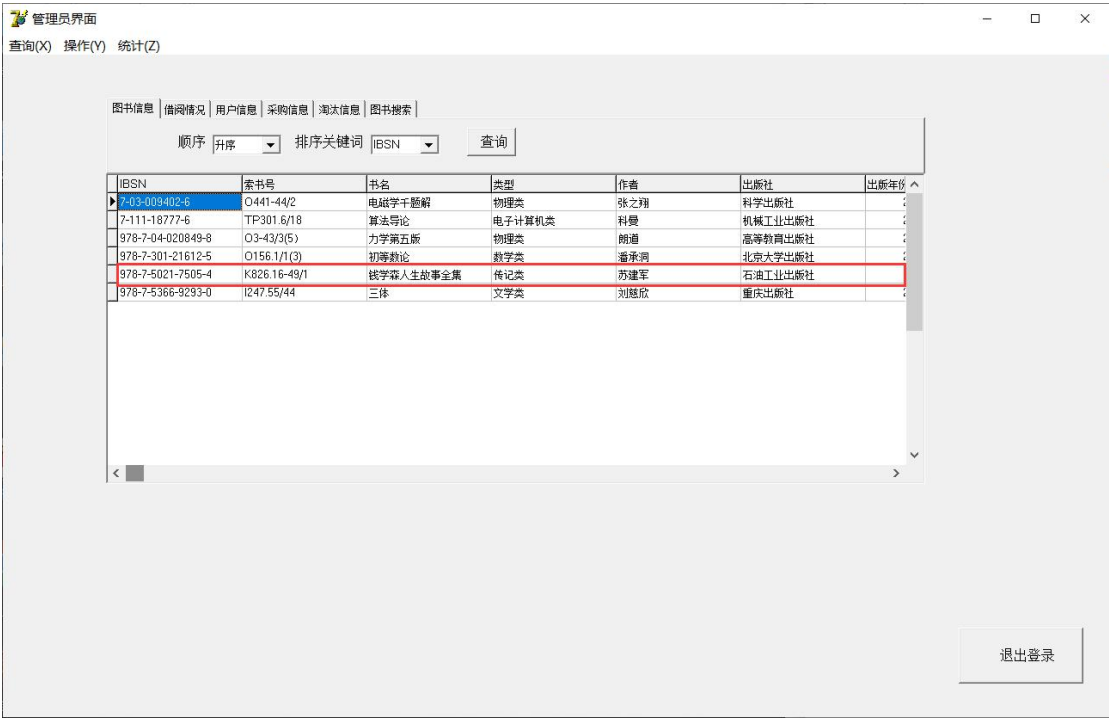
Project1

录入数据成功

OK

退出登录

此时再去查询图书信息，发现图书种类表中多了我们刚刚插入的图书的信息，和输入的信息一致，说明插入成功，库存数量、剩余数量也均按设计思路置为了 0。



对于一种信息已经被录入的图书的**采购**，在各文本栏中输入采购信息并点击提交采购数据按钮，就可以对一系列表格进行修改，从而自动实现采购数据的提交。其中，为了方便观察采购信息，我在采购这一栏加了一个表格用于显示采购信息表。实现采购的主要方法在于，首先，利用获得的信息构造一个采购信息表的元组。其中，日期利用 SQL 的 `getdate()` 获得，采购号将利用 ADOQuery 返回 `select`

语句对采购信息表的无条件查找得到的元组数，利用该元组数修改为采购号格式，从而实现顺序生成采购号。然后，将其利用 insert 语句插入到采购信息表中，表明这次采购的信息。此后，根据输入的 ISBN，利用 update 语句及 where 条件判断，找到图书种类表中 ISBN 属性为输入值的元组，然后将其库存数量和剩余数量均增加我们输入的采购数量，表明这种书采购了若干本。最后，将会查询淘汰信息表和特定某本书表中的书的总数，从而确定当前采购录入的这几本书是图书馆历史上的第几本书，然后顺序生成条码号。利用一个循环，循环采购数量次，利用 insert 语句将 ISBN 和按先前方法生成的条码号插入特定某本书表，其中学工号、借阅归还时间均置空，是否归还置‘是’，从而为每一本采购的书分配一个“身份”。从而实现对于图书种类表的更新、将相关信息加入采购信息表和特定某本书表，实现了采购功能。

如下为提交了 ISBN 为 978-7-5021-7505-4，采购数量为 5，单价为 28，采购原因为学生需求的采购结果。

管理员界面

查询(X) 操作(Y) 统计(Z)

借阅

归还

采购

淘汰

查询个人未还书籍

录入图书数据

ISBN

采购数量

单价

采购原因

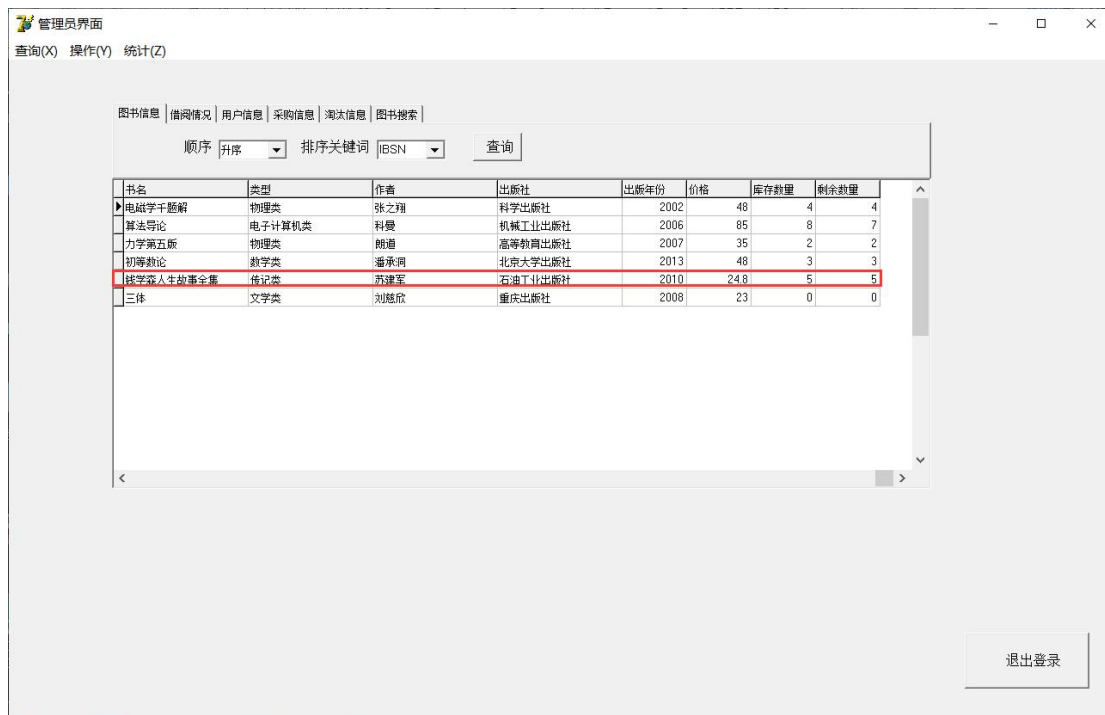
提交采购数据

采购信息

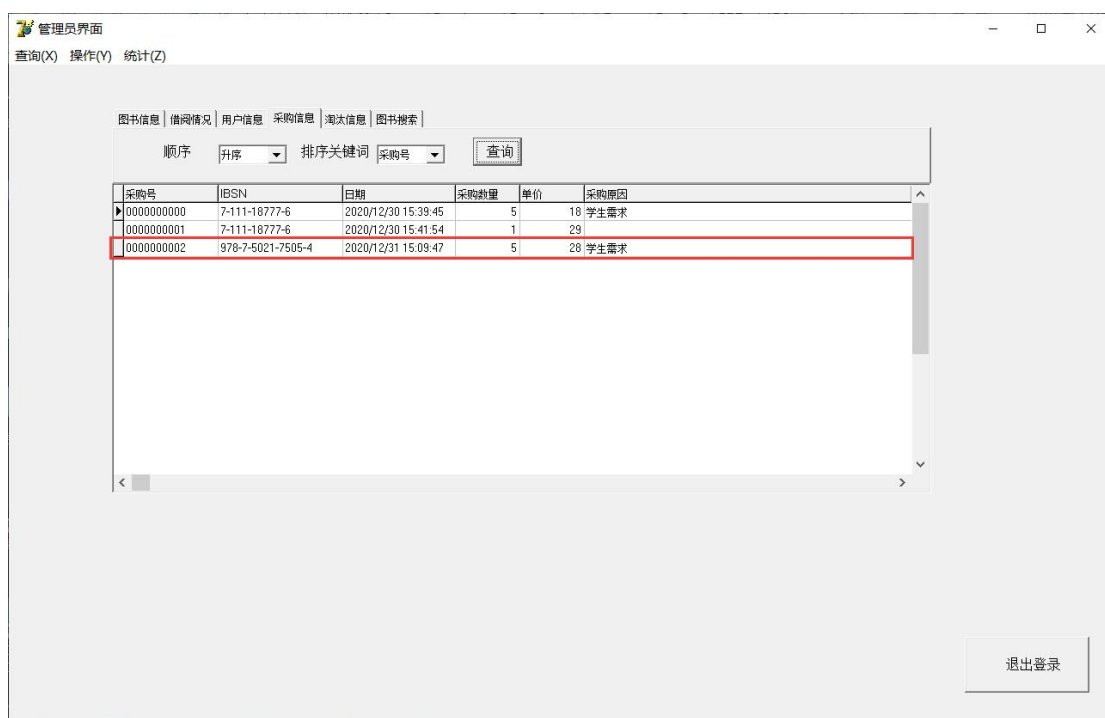
ISBN	日期	采购数量
7-111-18777-6	2020/12/30 15:39:46	5
7-111-18777-6	2020/12/30 15:41:54	1
978-7-5021-7505-4	2020/12/31 15:09:47	5

退出登录

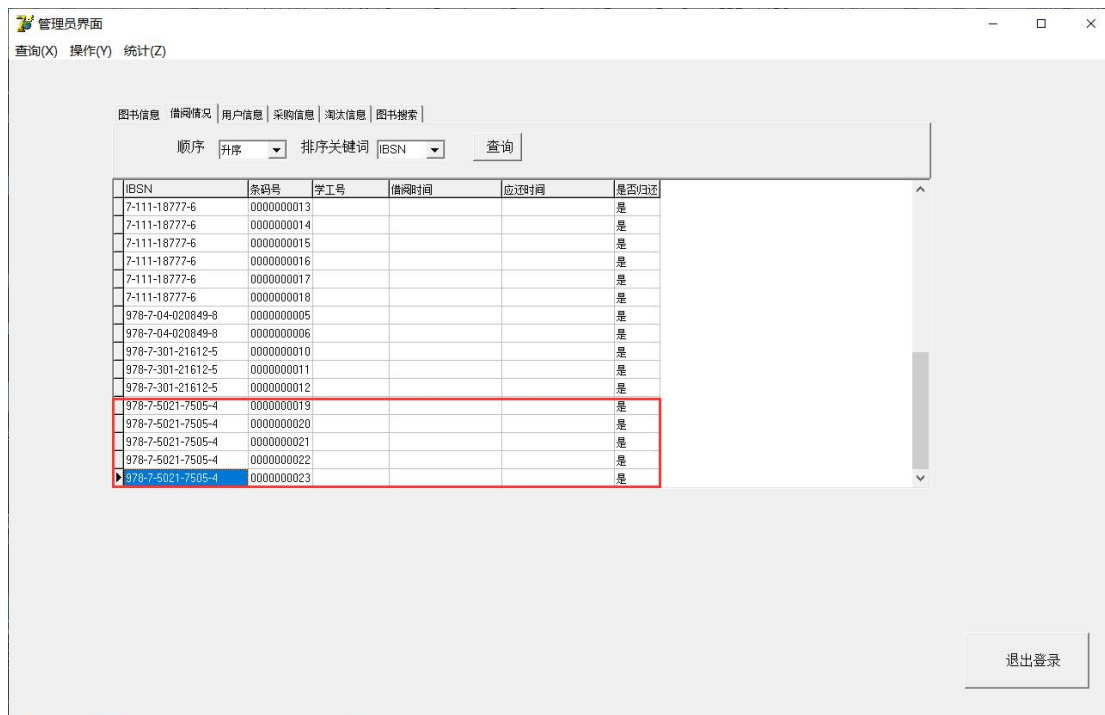
查看图书种类表，可以看到，我们采购的 ISBN 为 978-7-5021-7505-4 对应的钱学森人生故事全集的库存数量和剩余数量均为 5，相较于录入图书信息后的均为 0，增长的数量正为采购数量。



查看采购信息表，也增加了我们刚刚输入的采购信息。



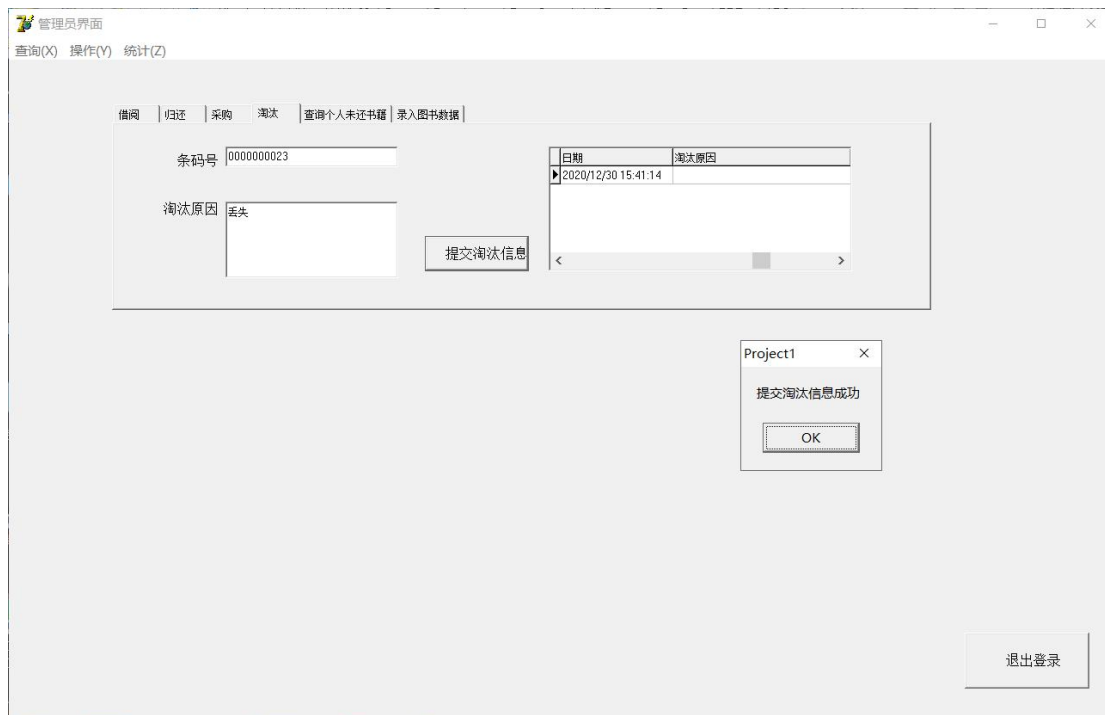
查看特定某本书表，此时多了条码号 19 到 23，其对应的 ISBN 和数量均符合我们之前采购时输入的信息。



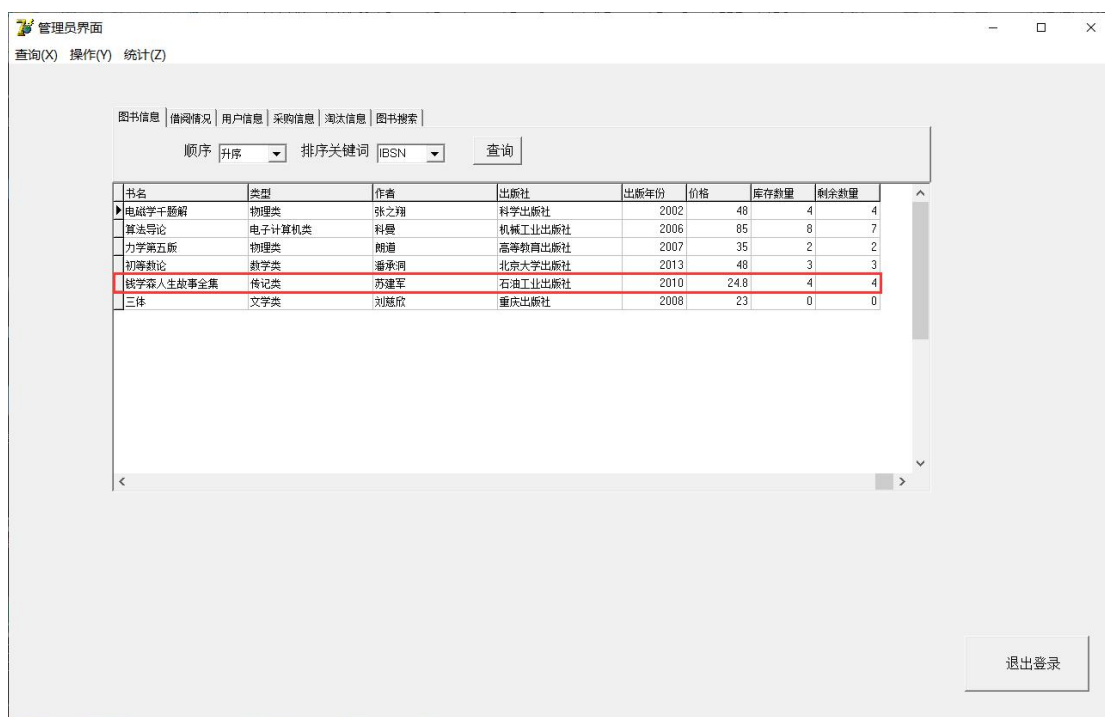
综上所述，成功实现了进行采购信息录入后对于各个表的修改。

对于**淘汰**，在各文本栏中输入条码号及淘汰原因，然后点击提交淘汰信息，将会自动对各个表进行修改，从而自动实现图书的淘汰功能。其中，为了方便观察，在淘汰栏加了一个表格用来显示淘汰信息。实现的主要思路在于，首先利用条码号，对特定某本书表进行搜索，获得该条码号对应的 ISBN。然后，利用 **update** 语句将该 ISBN 对应的图书种类表中的元组的库存数量和剩余数量均减一表示这种书被淘汰了一本。此后，利用条码号，使用 **delete** 语句和 **where** 条件判断从特定某本书表删除该条码号对应的元组，从而不再保存该被淘汰的书的借阅信息。最后利用 **insert** 语句，将本次淘汰信息插入淘汰信息表，其中条码号、淘汰原因为输入，淘汰号为查询淘汰信息表中的元组个数然后顺序生成的，日期为 **getdate()** 获得，ISBN 为先前查询的结果。最终实现对于淘汰信息表、图书种类表、特定某本书表的更新，实现了淘汰功能。

如下为输入及运行的结果，其中，我们淘汰了刚刚采购录入的对应条码号 0000000023 的一本书。



此时，查看图书种类表，发现录入的钱学森人生故事全集的库存数量和剩余数量均少了一本，说明有一本书被淘汰了。



再查看特定某本书表，发现表格中先前存在的 23 条码号的书已经消失了。

管理员界面

查询(X) 操作(Y) 统计(Z)

图书信息 | 借阅情况 | 用户信息 | 采购信息 | 淘汰信息 | 图书搜索 |

顺序 升序 排序关键词 条码号 查询

IBSN	条码号	学工号	借阅时间	应还时间	是否归还
978-7-04-020849-8	0000000006				是
7-111-18777-6	0000000008				是
7-111-18777-6	0000000009	AB12345678	2020/12/30 15:39:06	2021/1/29 15:39:06	否
978-7-301-21612-5	0000000010				是
978-7-301-21612-5	0000000011				是
978-7-301-21612-5	0000000012				是
7-111-18777-6	0000000013				是
7-111-18777-6	0000000014				是
7-111-18777-6	0000000015				是
7-111-18777-6	0000000016				是
7-111-18777-6	0000000017				是
7-111-18777-6	0000000018				是
978-7-5021-7505-4	0000000019				是
978-7-5021-7505-4	0000000020				是
978-7-5021-7505-4	0000000021				是
978-7-5021-7505-4	0000000022				是

退出登录

最后查看淘汰信息表，表中增加了一条淘汰信息，淘汰的条码号及原因正是我们先前输入的为丢失而淘汰的 23 号书。

管理员界面

查询(X) 操作(Y) 统计(Z)

图书信息 | 借阅情况 | 用户信息 | 采购信息 | 淘汰信息 | 图书搜索 |

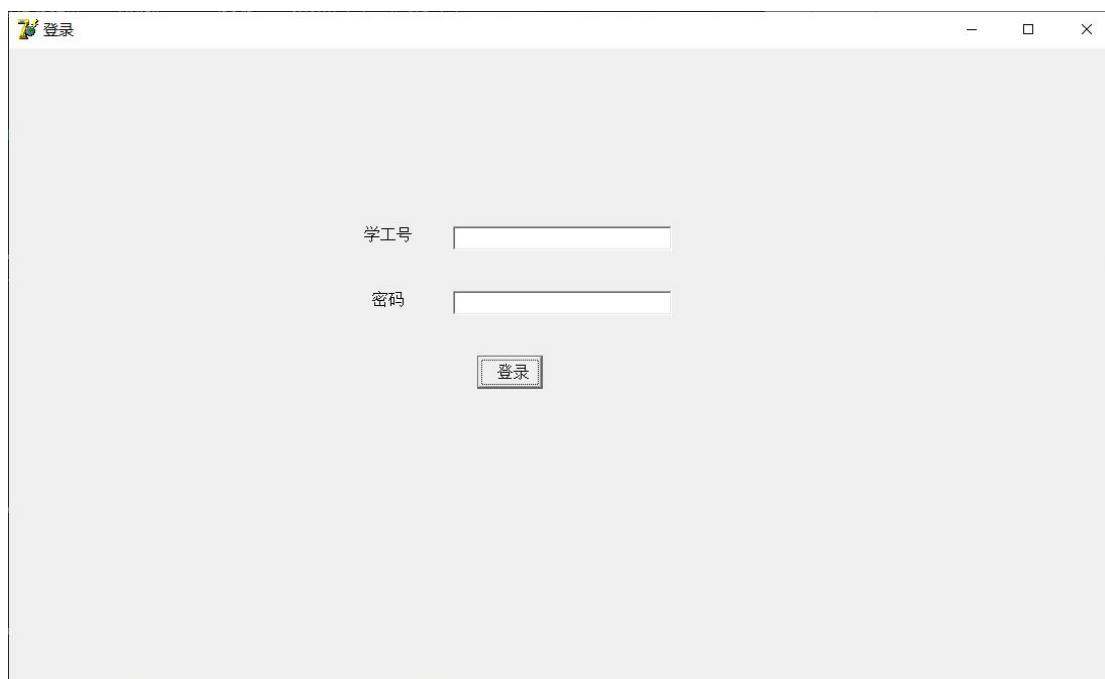
顺序 升序 排序关键词 淘汰号 查询

淘汰号	IBSN	条码号	日期	淘汰原因
0000000000	978-7-5366-9293-0	0000000007	2020/12/30 15:41:14	
0000000001	978-7-5021-7505-4	0000000023	2020/12/31 16:08:26	丢失

退出登录

说明我们的淘汰操作成功实现了对于书淘汰后各个表的修改。

此时为了展示普通用户界面，我们可以点击退出登录按钮，该按钮触发的进程会将管理员界面隐藏并显示登录界面，同时将登录界面的学工号和密码文本栏均置空。



登录

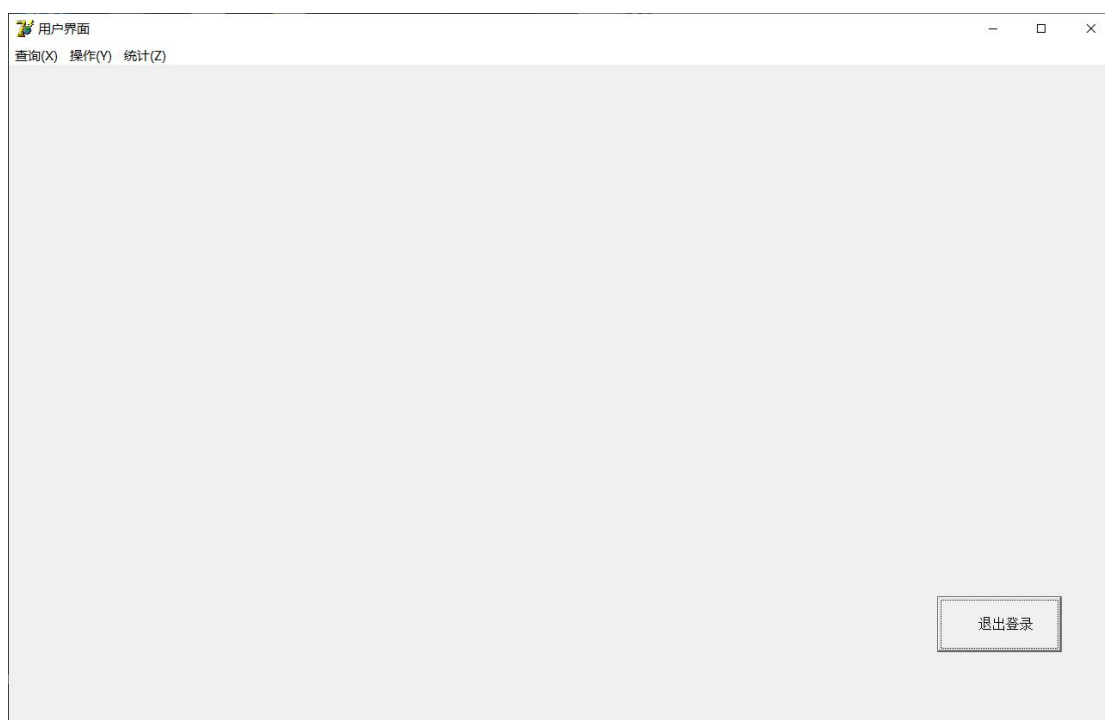
学工号

密码

登录

2.普通用户界面

利用普通用户账号 PB18000021 及密码 12345 登录，进入普通用户界面如下

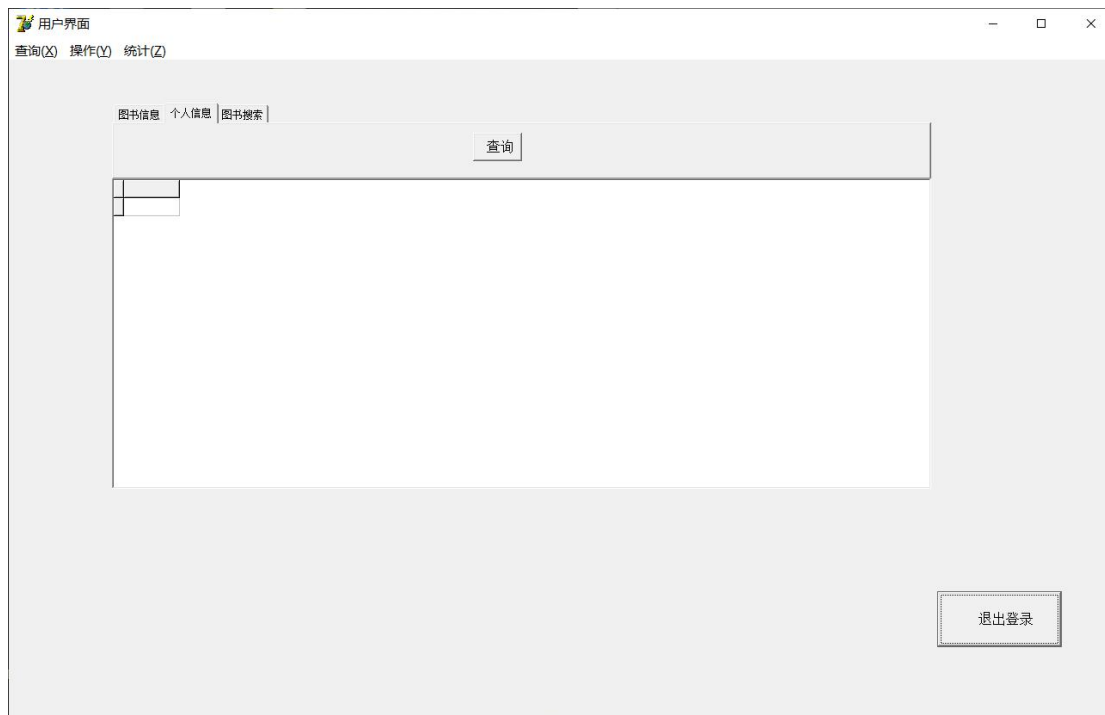


用户界面

查询(X) 操作(Y) 统计(Z)

退出登录

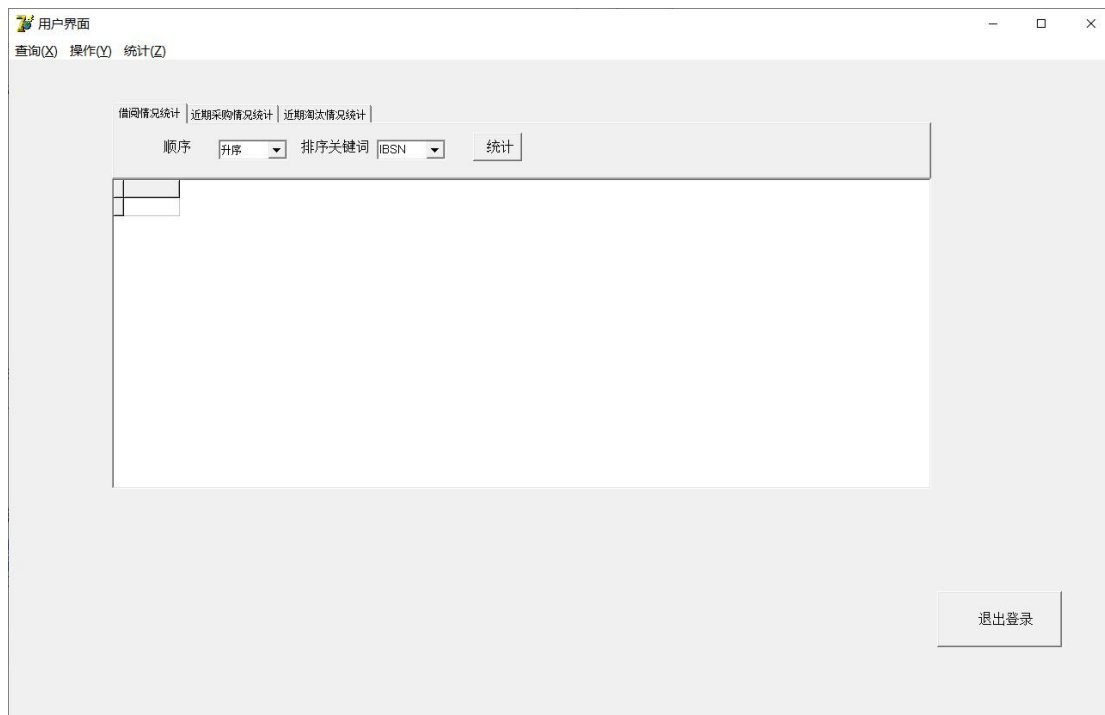
选择主菜单**查询**按钮后，将会显示如下图所示的普通用户所允许进行的查询，包括图书信息、个人信息、图书搜索，管理员所进行的淘汰、采购、特定某本书的查询均不对用户开放，同时将管理员的用户信息查询替换为个人信息查询，从而避免用户信息泄露。图书信息、图书搜索先前已经介绍过就不再赘述。



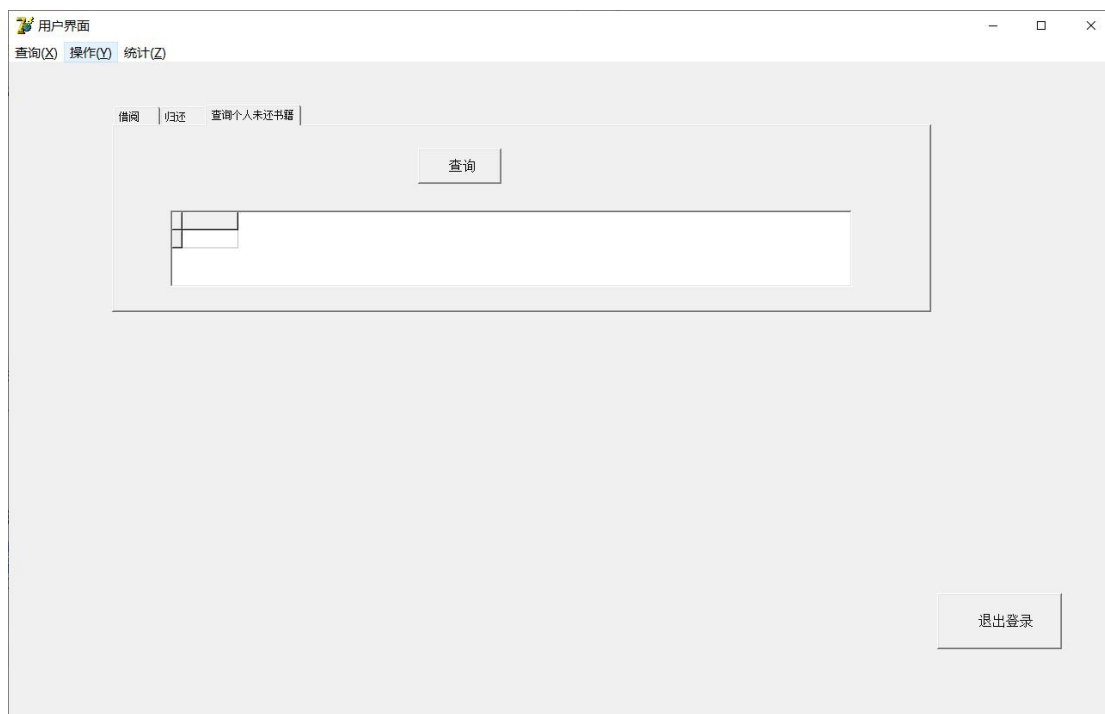
对于**个人信息**，查询结果如下，将会显示当前登录用户的在用户表中的信息。实现的主要思路在于利用隐藏的登录界面中的学工号一栏的文本对用户表进行 **select** 语句查找，查找属性学工号为输入的元组并返回到表单上。图中可以看出当前的姓名为张三学工号为 **PB18000021**，恰为我们登录的账号。



选择主菜单**统计**按钮后，会显示普通用户所允许进行的统计，包括借阅情况统计、近期采购情况统计、近期淘汰情况统计，均在管理员所能进行的统计中已经介绍过，不包含管理员能看到的具体的借出书籍统计。



选择主菜单**操作**按钮后，将会显示普通用户所允许进行的操作，包括借阅、归还、查询个人未还书籍，由于普通用户不应该能直接对采购、图书、淘汰信息进行操作，所以不包括管理员的淘汰、采购、录入图书信息的操作。这些操作均包含在之前的管理员操作中，已经介绍过，不再赘述。



综上所述，该程序实现了对于不同用户的权限的区分，并且成功实现了对于不同功能的实现。其中数据库本身对应着题目中图书管理系统的对于存储的要

求，查询的功能对应了要求中对于各类信息的查询要求，操作的功能对应了要求中对于图书的采购淘汰租借功能的要求，统计的功能对应了题目中对于各类信息的统计要求，故该数据库及应用程序成功实现了实验的要求。

五、反省与总结：

本次实验主要考察了对于整个学期数据知识的综合运用，以及对于 Delphi 的学习和掌握程度。在整个实验中，沿着数据库设计步骤——需求分析、概念结构设计、逻辑结构设计、数据库物理设计、数据库实施、数据库运行维护这一整条主线，从无到有的完成了一个数据库及其相关操作及应用程序的设计与实现。同时伴随各个步骤，复习了 SQL 代码、E-R 图、约束、权限划分、用户标识与鉴定等一系列所学的知识，并将其融会贯通。本次实验是我第一次体验将一门课所学的所有知识进行综合运用并落实到实践，对于数据库的知识是一个很好的复习，在实践中大大加深了对于所学知识的理解，收益颇丰。同时通过对于 Delphi 的学习，我了解了如何设计一个 UI 界面并将其与程序进行联动，还体会了如何在一个较短的时间内对于一门新的语言进行掌握。

在这次实验过程中，也出现了不少的问题。首先，遇到的最困难的一点在于对 Delphi 的使用，仅仅通过 PPT 不足以满足应用程序实现的需求，于是我在图书馆借到了相关资料进行了初步学习并了解了不同组件的大致功能，然后每当我需要使用一定的组件，便在网上查询其具体的属性及用法。由于 Delphi 虽然方便但不是特别热门，网上的资料较少，故我尽量尝试用常见的组件及代码去替代较为复杂的组件，辅之以一些猜测，最终成功实现了我的需求。我在 SQL 代码实现中也遇到了一系列问题，例如不确定如何获得时间，最终通过在网上查找资料解决。其中我遇到的最奇怪的问题在于字符串处理，查找了很多资料发现我将文本框中的字符串加入我所要的 SQL 代码以后，若在 SQL 代码中应作为字符串存在，则应在两端各加上两个单引号，以作为转义字符表示一个单引号。此外，数据库的分析与设计部分花了我不少时间，在之后进行实际操作中，我也发现了一些我的不合理冗余设计，通过这里我意识到了老师所讲的需求分析及设计的复杂性。

最后，感谢老师和助教这一个学期的教导与帮助，我在这个学期的课程中首次借触数据库，学习到了很多知识，收益颇丰。