浙江大学

数据库系统实验报告

实验名称：	图书管理系统
姓　　名：	蒋奕
学　　号：	3210103803
专　　业：	计算机科学与技术
课程名称：	数据库系统
实验地点：	紫金港机房
指导教师：	陈璐

2023 年 5 月 6 日

图书管理系统

1 实验目的和要求 **2**

2 实验内容和原理 **2**

2.1 系统架构描述 2

2.2 实验原理 2

3 实验设备和环境 **5**

4 实验实现方法、步骤与测试 **5**

4.1 初始化界面 5

4.2 管理员登录 6

4.3 图书入库 6

4.4 借书 8

4.5 还书 8

4.6 借书证管理 8

4.7 返回初始界面 9

4.8 排序图书 10

4.9 条件查询图书 10

5 附录 **11**

6 参考资料 **28**

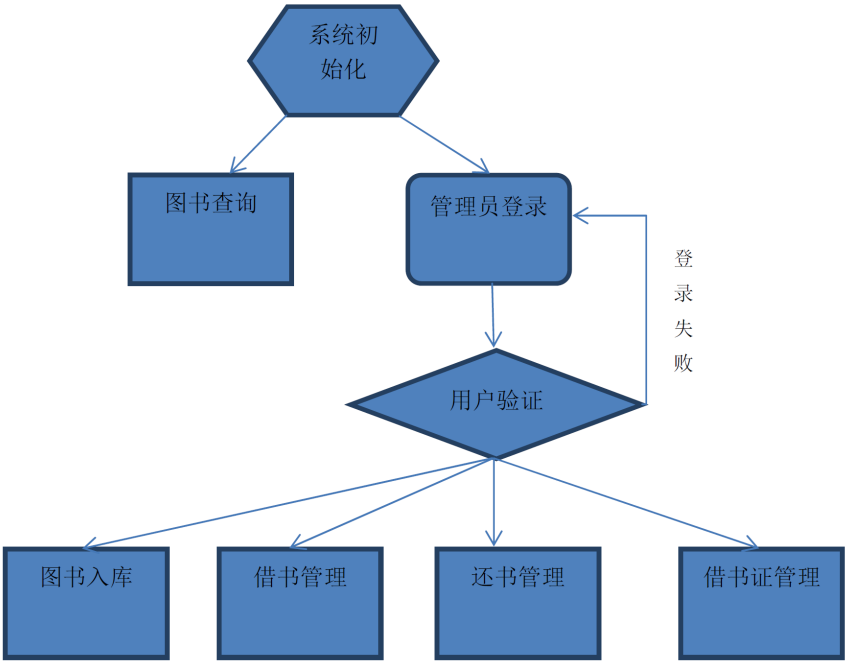
1 实验目的和要求

- 设计并实现一个精简的图书管理系统，具有入库、查询、借书、还书、借书证管理等基本功能。
- 通过本次设计来加深对数据库的了解和使用，同时提高自身的系统编程能力。

2 实验内容和原理

2.1 系统架构描述

本系统主要包括管理员登录、图书入库、图书查询、借书管理、还书管理、借书证管理六大功能模块。系统处理基本流程如下：



系统初始时仅有图书查询和管理员登录两个选项，图书查询为公共功能模块，不需要登录也可操作。管理员成功登录后，便可以选择进入图书入库、借书管理、还书管理、借书证管理功能。

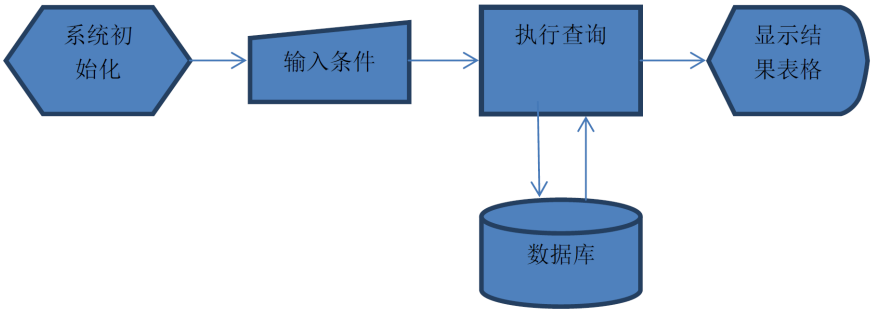
2.2 实验原理

各个功能模块说明如下：

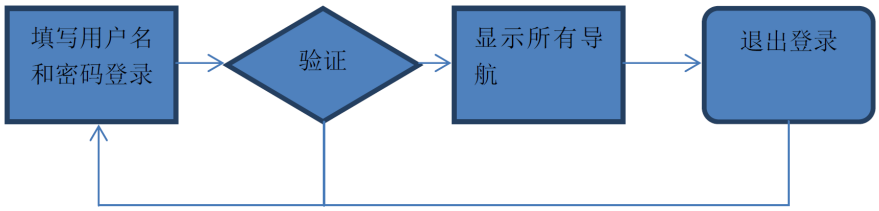
模块名称	功能描述
管理员登陆	根据管理员 ID 和密登录系统
图书入库	1. 单本入库 ，直接从程序界面上输入 2. 批量入库，从文本文件中批量导入图书数据

图书查询	1．按书的类别，书名，出版社，年份，作者，价格进行查询. 2．可以点击标题来对相应的字段进行排序
借书	1.输入借书证卡号，自动显示该借书证所有已借书籍 2.输入书号，如果该书还有库存，则提示借书成功，同时库存数减 1，否则输出该书无库存。
还书	1.输入借书证卡号，自动显示该借书证所有已借书籍 2.输入书号，如果该书在已借书籍列表内，则还书成功，同时库存加 1.否则输出出错信息。
借书证管理	增加或删除借书证.

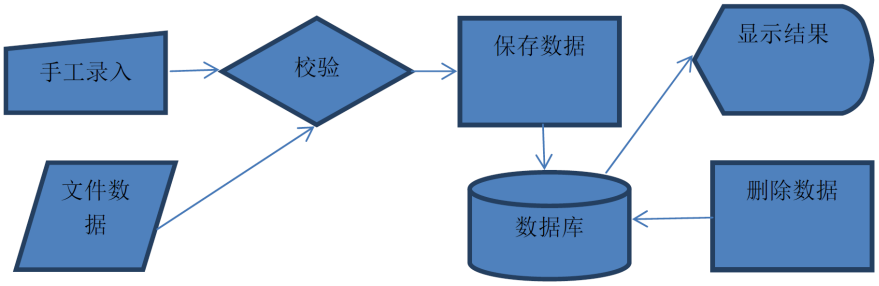
图书查询功能模块说明如下：



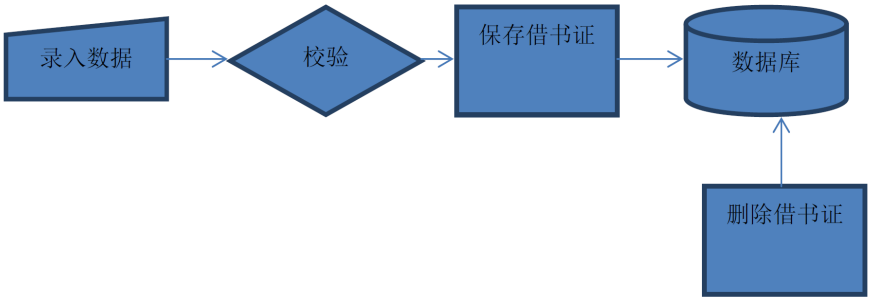
管理员登录功能模块说明如下：



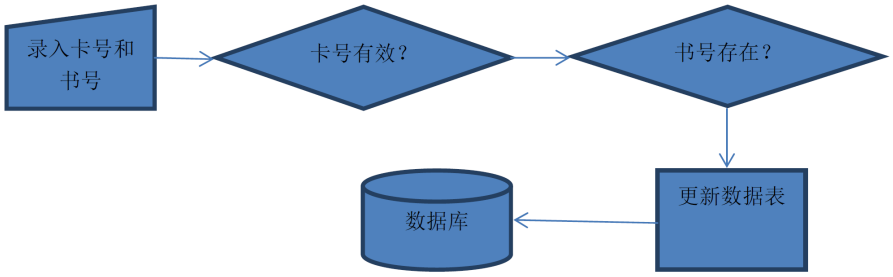
图书入库模块说明如下：



借书证管理功能模块说明如下：



还书功能模块说明如下：



数据库建立的代码如下所示:

```
create database test;
use test;
create table `admin`
(
`adminID`    int not null,
`password`   varchar(64) not null,
`name`       varchar(32) not null,
`contact`    varchar(64) not null,
primary key (adminID)
);
create table `books`
(
    `bookID`   int not null,
    `type`     varchar(32),
    `bookName`   varchar(64) not null,
    `publisher`  varchar(64),
    `publishYear` int,
    `author `  varchar(64),
    `price`    numeric(6, 2),
    `totalNum`   int not null,
    `storageNum`   int not null,
    `updateTime`   datetime,
    primary key (bookID),
    check (totalNum >= storageNum),
    check (storageNum >= 0),
    check (price >= 0.0)
);

create table `cards`
(
    `cardID`   int not null,
```

```
    `name`     varchar(64) not null,
    `department`   varchar(64),
    `type`     varchar(32) not null,
    `updateTime`   datetime,
    primary key (cardID)
);
create table records
(
    `recordID`   int not null,
    `cardID`   int not null,
    `bookID`   int not null,
    `borrowDate`   datetime,
    `returnDate`   datetime,
    `adminID` int not null,
    primary key (`recordID`),
    foreign key (`cardID`) references cards(`cardID`) on update cascade on delete cascade,
    foreign key (`bookID`) references books(`bookID`) on update cascade on delete cascade,
    foreign key (`adminID`) references admin(`adminID`) on update cascade on delete cascade
);
```

1: 建立数据库

数据库初始化的代码如下所示:

```
insert into `admin` values(1,'admin1','jiangyi','18957327377');
insert into `admin` values(2,'123456','jy','3210103803@zju.edu.cn');
insert into `books` values(4, 'computer', 'introduction to computer system', null, 2010, '
    patt', 12.5, 5, 0,'2017-12-20 10:01:00');
insert into `cards` values(1,'jiangyi','CS','student','2022-12-20 10:01:00');
insert into `books` values(7, 'db', 'db_textbook', null, 2010, 'xyz', 56.5, 9, 6,'2022-12-20
     18:01:30');
insert into `cards` values(2,'chenlu','cs','teacher',now());
insert into `records` values(1,1,7,now(),null,1);
insert into `records` values(3,1,4,now(),null,1);
```

2: 初始化数据库

3 实验设备和环境

开发工具：VS2019；数据库平台：MySQL 2019

4 实验实现方法、步骤与测试

本实验主要采用C++语言描述并且实现图书管理系统中的各个模块。然后自己设计测试数据进行调试直到功能正确。

4.1 初始化界面

输入登录模式。

C:\Users\jiangyi\Desktop\BookManagement_demo\x64\Debug\BookManagement_demo.exe

```
success
please input mode
0:only check books
1:admin login
```

4.2 管理员登录

输入ID和密码，如果错误则要求重新输入。

C:\Users\jiangyi\Desktop\BookManagement_demo\x64\Debug\BookManagement_demo.exe

```
success
please input mode
0:only check books
1:admin login
1
input ID:
1
input password:
wrongpassword
no such admin exist
input ID:
1
input password:
admin1
admin 1 successful login!
input things to do:
-1:exit login
0:books In
1:borrow Books
2:return Books
3:card Manage
```

4.3 图书入库

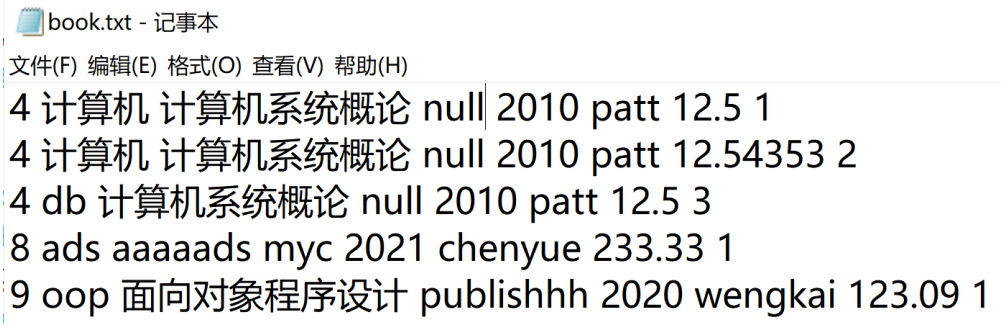
单本图书入库：输入图书信息。

```
number of selection:4
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
4        计算机  计算机系统概论  NULL    2010    patt    12.50   5       0       2017-12-20 10:01:00
7        db      db_textbook     NULL    2010    xyz     56.50   12      11      2023-03-23 22:32:25
8        ads     aaaaads myc     2021    chenyue 233.33  2       2       2023-05-05 10:13:25
9        oop     面向对象程序设计        publishhh       2020    wengkai 123.09  2       2       2023-05-05 10:13:25
please input mode
0:only get 1 book per time
1:get book from file
2:delete one kind of book
0
input book info(BookID,Type,BookName,Publisher,PublishYear,Author,Price,Num) seperated by space:
5 fds algorithm fffddd 2009 noaoo 45.67 4
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
4        计算机  计算机系统概论  NULL    2010    patt    12.50   5       0       2017-12-20 10:01:00
7        db      db_textbook     NULL    2010    xyz     56.50   12      11      2023-03-23 22:32:25
8        ads     aaaaads myc     2021    chenyue 233.33  2       2       2023-05-05 10:13:25
9        oop     面向对象程序设计        publishhh       2020    wengkai 123.09  2       2       2023-05-05 10:13:25
5        fds     algorithm       fffddd  2009    noaoo   45.67   4       4       2023-05-06 20:35:48
input things to do:
-1:exit login
0:books In
1:borrow Books
2:return Books
3:card Manage
```

文件导入图书入库：输入文件路径。如果图书信息有错误，则显示insertion fail，否则插入并且合并图书信息。

```
1
input filepath
D:\\book.txt
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
7        db     db_textbook     NULL    2010     xyz      56.50   12      11      2023-03-23 22:32:25
8        ads    aaaaads myc     2021    chenyue 233.33   2       2       2023-05-05 10:13:25
9        oop    面向对象程序设计        publishhh       2020    wengkai 123.09  2       2       2023-05-05 10:13:25
5        fds    algorithm       fffddd  2009     noaoo    45.67   4       4       2023-05-06 20:35:48
4        计算机  计算机系统概论  NULL    2010     patt     12.50   6       1       2023-05-06 20:38:35
insertion fail
insertion fail
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
7        db     db_textbook     NULL    2010     xyz      56.50   12      11      2023-03-23 22:32:25
9        oop    面向对象程序设计        publishhh       2020    wengkai 123.09  2       2       2023-05-05 10:13:25
5        fds    algorithm       fffddd  2009     noaoo    45.67   4       4       2023-05-06 20:35:48
4        计算机  计算机系统概论  NULL    2010     patt     12.50   6       1       2023-05-06 20:38:35
8        ads    aaaaads myc     2021    chenyue 233.33   3       3       2023-05-06 20:38:35
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
7        db     db_textbook     NULL    2010     xyz      56.50   12      11      2023-03-23 22:32:25
5        fds    algorithm       fffddd  2009     noaoo    45.67   4       4       2023-05-06 20:35:48
4        计算机  计算机系统概论  NULL    2010     patt     12.50   6       1       2023-05-06 20:38:35
8        ads    aaaaads myc     2021    chenyue 233.33   3       3       2023-05-06 20:38:35
9        oop    面向对象程序设计        publishhh       2020    wengkai 123.09  3       3       2023-05-06 20:38:35
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
7        db     db_textbook     NULL    2010     xyz      56.50   12      11      2023-03-23 22:32:25
5        fds    algorithm       fffddd  2009     noaoo    45.67   4       4       2023-05-06 20:35:48
4        计算机  计算机系统概论  NULL    2010     patt     12.50   6       1       2023-05-06 20:38:35
8        ads    aaaaads myc     2021    chenyue 233.33   3       3       2023-05-06 20:38:35
9        oop    面向对象程序设计        publishhh       2020    wengkai 123.09  3       3       2023-05-06 20:38:35
input things to do:
```

文件导入图书入库信息如下，显而易见第2,3条数据是错误的。



删除一种图书：如果还有未还书籍则不能够删除。

```
0
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
7        db     db_textbook     NULL    2010     xyz      56.50   12      11      2023-03-23 22:32:25
5        fds    algorithm       fffddd  2009     noaoo    45.67   4       4       2023-05-06 20:35:48
4        计算机  计算机系统概论  NULL    2010     patt     12.50   6       1       2023-05-06 20:38:35
8        ads    aaaaads myc     2021    chenyue 233.33   3       3       2023-05-06 20:38:35
9        oop    面向对象程序设计        publishhh       2020    wengkai 123.09  3       3       2023-05-06 20:38:35
please input mode
0:only get 1 book per time
1:get book from file
2:delete one kind of book
2
input bookid to be deleted:
4
not returned book--delete book fails!
```

可以看到，ID=4的书籍是未还书籍，不能够删除。

```
mysql> select * from `records`;
+----------+--------+--------+---------------------+---------------------+---------+
| recordID | cardID | bookID | borrowDate          | returnDate          | adminID |
+----------+--------+--------+---------------------+---------------------+---------+
|        1 |      1 |      7 | 2023-03-24 11:27:19 | 2023-03-24 16:18:54 |       1 |
|        2 |      2 |      7 | 2023-03-24 16:23:01 | 2023-03-24 16:24:25 |       2 |
|        3 |      1 |      4 | 2023-05-06 16:24:31 | NULL                |       1 |
|        4 |      2 |      8 | 2023-05-05 10:21:20 | 2023-05-05 10:24:30 |       1 |
+----------+--------+--------+---------------------+---------------------+---------+
4 rows in set (0.00 sec)
```

4.4 借书

输入bookID和cardID，如果cardID有没有还的书则不可以借书。

```
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
4        计算机  计算机系统概论  NULL    2010    patt    12.50   6       1       2023-05-06 20:38:35
5        fds     algorithm       fffddd  2009    noaoo   45.67   4       4       2023-05-06 20:35:48
7        db      db_textbook     NULL    2010    xyz     56.50   12      11      2023-03-23 22:32:25
8        ads     aaaaads myc     2021    chenyue 233.33  3       3       2023-05-06 20:38:35
9        oop     面向对象程序设计        publishhh       2020    wengkai 123.09  3       3       2023-05-06 20:38:35
please input bookID
4
please input cardID
1
not returned book--borrow fails!
```

输入bookID和cardID，成功借书。

```
1
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
4        计算机  计算机系统概论  NULL    2010    patt    12.50   6       1       2023-05-06 20:38:35
5        fds     algorithm       fffddd  2009    noaoo   45.67   4       4       2023-05-06 20:35:48
7        db      db_textbook     NULL    2010    xyz     56.50   12      11      2023-03-23 22:32:25
8        ads     aaaaads myc     2021    chenyue 233.33  3       3       2023-05-06 20:38:35
9        oop     面向对象程序设计        publishhh       2020    wengkai 123.09  3       3       2023-05-06 20:38:35
please input bookID
4
please input cardID
2
number of selection:5
recordID   cardID   bookID   borrowDate   returnDate   adminID
1        1       7       2023-03-24 11:27:19     2023-03-24 16:18:54     1
2        2       7       2023-03-24 16:23:01     2023-03-24 16:24:25     2
3        1       4       2023-05-06 16:24:31     NULL    1
4        2       8       2023-05-05 10:21:20     2023-05-05 10:24:30     1
5        2       4       2023-05-06 20:54:57     NULL    1
```

4.5 还书

输入bookID和cardID，如果cardID没有借该书则不可以还该书。

```
2
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
4        计算机  计算机系统概论  NULL    2010    patt    12.50   6       1       2023-05-06 20:38:35
5        fds     algorithm       fffddd  2009    noaoo   45.67   4       4       2023-05-06 20:35:48
7        db      db_textbook     NULL    2010    xyz     56.50   12      11      2023-03-23 22:32:25
8        ads     aaaaads myc     2021    chenyue 233.33  3       3       2023-05-06 20:38:35
9        oop     面向对象程序设计        publishhh       2020    wengkai 123.09  3       3       2023-05-06 20:38:35
please input bookID
4
please input cardID
2
no records exist--fail
```

输入bookID和cardID，成功还书。

```
2
number of selection:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
4        计算机  计算机系统概论  NULL    2010    patt    12.50   6       0       2023-05-06 20:38:35
5        fds     algorithm       fffddd  2009    noaoo   45.67   4       4       2023-05-06 20:35:48
7        db      db_textbook     NULL    2010    xyz     56.50   12      11      2023-03-23 22:32:25
8        ads     aaaaads myc     2021    chenyue 233.33  3       3       2023-05-06 20:38:35
9        oop     面向对象程序设计        publishhh       2020    wengkai 123.09  3       3       2023-05-06 20:38:35
please input bookID
4
please input cardID
2
number of selection:5
recordID   cardID   bookID   borrowDate   returnDate   adminID
1        1       7       2023-03-24 11:27:19     2023-03-24 16:18:54     1
2        2       7       2023-03-24 16:23:01     2023-03-24 16:24:25     2
3        1       4       2023-05-06 16:24:31     NULL    1
4        2       8       2023-05-05 10:21:20     2023-05-05 10:24:30     1
5        2       4       2023-05-06 20:54:57     2023-05-06 20:56:43     1
input things to do:
```

4.6 借书证管理

删除借书证：输入cardID，如果cardID有借书没还则不可以删除。

```
number of selection:4
cardID   name   department   type   updateTime
1        jiangyi CS      student 2022-12-20 10:01:00
2        chenlu  cs      teacher 2023-03-22 18:12:42
3        jy      CS      student 2023-03-24 16:28:48
4        jjjyyy  cs      student 2023-05-05 10:20:05
please input mode
0:add new card
1:delete card
1
please input cardID to be deleted
1
not returned book--delete card fails!
```

增加借书证：输入相关信息。

```
3
number of selection:4
cardID   name   department   type   updateTime
1        jiangyi CS      student 2022-12-20 10:01:00
2        chenlu  cs      teacher 2023-03-22 18:12:42
3        jy      CS      student 2023-03-24 16:28:48
4        jjjyyy  cs      student 2023-05-05 10:20:05
please input mode
0:add new card
1:delete card
0
please input cardID, name, department, Type
6 jjyy ee teacher
insert into `cards` values(6,'jjyy','ee','teacher',now());
number of selection:5
cardID   name   department   type   updateTime
1        jiangyi CS      student 2022-12-20 10:01:00
2        chenlu  cs      teacher 2023-03-22 18:12:42
3        jy      CS      student 2023-03-24 16:28:48
4        jjjyyy  cs      student 2023-05-05 10:20:05
6        jjyy    ee      teacher 2023-05-06 21:03:14
```

4.7 返回初始界面

```
input things to do:
-1:exit login
0:books In
1:borrow Books
2:return Books
3:card Manage
-1
Please choose whether to go on or exit:
0:exit
1:goon
1
please input mode
0:only check books
1:admin login
```

4.8 排序图书

如下提示操作即可。（对价格降序排序为例子）

```
input sort or not
0:no sort and break directly
1:sort
1
input type of sort
0:bookID
1:type
2:bookName
3:publisher
4:publishYear
5:author
6:price
7:totalNum
8:storageNum
9:updateTime
6
input type of order
0:asc
1:desc
1
number of books selected:5
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
8        ads    aaaaads myc     2021    chenyue 233.33  3       3       2023-05-06 20:38:35
9        oop    面向对象程序设计        publishhh       2020    wengkai 123.09  3       3       2023-05-06 20:38:35
7        db     db_textbook     NULL2010        xyz     56.50   12      11      2023-03-23 22:32:25
5        fds    algorithm       fffddd  2009    noaoo   45.67   4       4       2023-05-06 20:35:48
4        计算机  计算机系统概论  NULL2010        patt    12.50   6       1       2023-05-06 20:38:35
```

4.9 条件查询图书

输入条件个数，依次输入条件要求如下：

```
Please input the number of conditions
3
input type of partial select:
1:type
2:bookName
3:publisher
4:publishYear
5:author
6:price
4
Please input low and upper bound of publishYear(range check) by space
1999 2020
input type of partial select:
1:type
2:bookName
3:publisher
4:publishYear
5:author
6:price
6
Please input low and upper bound of price(range check) by space
10 100.09
input type of partial select:
1:type
2:bookName
3:publisher
4:publishYear
5:author
6:price
1
Please input type(accurate check)
db
```

查询结果如下（可以进一步进行排序等等）：

```
number of books selected:1
bookID   type   bookName   publisher   publishYear   author   price   totalNum   storageNum   updateTime
7        db     db_textbook     NULL2010        xyz     56.50   12      11      2023-03-23 22:32:25
input partially sort or not
0:no sort and break directly
1:partially sort
```

5 附录

程序源代码如下：

```
# define _CRT_SECURE_NO_WARNINGS
# include <winsock.h>
# include <mysql.h>
# include <iostream>
# include <fstream>
# include <string>
# include <cstdlib>
# include <cstdio>
# define OnlyCheckBooks 0
# define AdminLogin 1
# define bookID 0
# define type 1
# define bookName 2
# define publisher 3
# define publishYear 4
# define author 5
# define price 6
# define totalNum 7
# define storageNum 8
# define updateTime 9
# define ascend 0
# define descend 1
# define exitLogin -1
# define booksIn 0
# define borrowBooks 1
# define returnBooks 2
# define cardManage 3
# define OneBook 0
# define ManyBooks 1
# define DeleteOneKind 2
# define AddCard 0
# define DeleteCard 1
# define NULLRepresent "null"
# define Accurate 0
# define Fuzzy 1
# define Range 2
using namespace std;
MYSQL mysql;
const char* host = "localhost";
const char* user = "root";
const char* password = "jjh2002jy";
const char* database = "test";
const int port = 3306;
int mode;
void set(string& s)
{
  if (s == "null")
  {
    s = "null";
```

```
  }
  else {
    s = "'" + s + "'";
  }
}
void sortPartialBooksByOrder(MYSQL_RES* result, string condition)/*partial books*/
{
  int DoSort = 0;
  int sorttype = 0;
  int order = 0;/*order == 0; ascend   order == 1;descend*/
  printf("input partially sort or not\n0:no sort and break directly\n1:partially sort\n");
  cin >> DoSort;//scanf("%d", &DoSort);
  string sql = "select * from `books` order by ";
  while (DoSort == 1)
  {
    sql = "select * from `books`" + condition + " order by ";
    if (result)
    {
      printf("input type of sort\n0:bookID\n1:type\n2:bookName\n3:publisher\n4:publishYear\
          n5:author\n6:price\n7:totalNum\n8:storageNum\n9:updateTime\n");
      cin >> sorttype;//scanf("%d", &sorttype);
      printf("input type of order\n0:asc\n1:desc\n");
      cin >> order;//scanf("%d", &order);
      switch (sorttype)
      {
        case bookID: sql += "bookID "; break;
        case type:sql += "type "; break;
        case bookName:sql += "bookName "; break;
        case publisher:sql += "publisher "; break;
        case publishYear:sql += "publishYear "; break;
        case author:sql += "author "; break;
        case price:sql += "price "; break;
        case totalNum:sql += "totalNum "; break;
        case storageNum:sql += "storageNum "; break;
        case updateTime:sql += "updateTime "; break;
        default:break;
      }
      if (order == ascend) sql += "ASC;";
      else if (order == descend) sql += "DESC;";
      int query = mysql_query(&mysql, sql.c_str());
      if (query == 0){
        MYSQL_RES* partialResult = mysql_store_result(&mysql);
        if (partialResult){
          unsigned long partialNumber = (unsigned long)mysql_num_rows(partialResult);
          cout << "number of books selected:" << partialNumber << endl;
          MYSQL_FIELD* fd = mysql_fetch_field(partialResult);
          while (fd){
            cout << fd->name << "   ";
            fd = mysql_fetch_field(partialResult);
          }
          cout << endl;
          MYSQL_ROW row;
          while (row = mysql_fetch_row(partialResult)){
            unsigned long* lens = mysql_fetch_lengths(partialResult);
            for (int i = 0; i < mysql_num_fields(partialResult); ++i)
              if (row[i]) cout << row[i] << "\t";
              else cout << "NULL";
            cout << endl;
          }
```

```
108         }
            mysql_free_result(partialResult);
110       }
      }
112     cout << "input sort or not\n0:no sort and break directly\n1:sort\n";
      cin >> DoSort;//scanf("%d", &DoSort);
114   }
    return;
116 }
  void sortAllBooksByOrder(MYSQL_RES* result)/*all books*/
118 {
    int DoSort = 0;
120   int sorttype = 0;
    int order = 0;/*order == 0; ascend    order == 1;descend*/
122   printf("input sort or not\n0:no sort and break directly\n1:sort\n");
    cin >> DoSort;
124   string sql = "select * from `books` order by ";
    while (DoSort == 1){
126     sql = "select * from `books` order by ";
      if (result)
128     {
        printf("input type of sort\n0:bookID\n1:type\n2:bookName\n3:publisher\n4:publishYear\
            n5:author\n6:price\n7:totalNum\n8:storageNum\n9:updateTime\n");
130     cin >> sorttype;//scanf("%d", &sorttype);
        printf("input type of order\n0:asc\n1:desc\n");
132     cin >> order;//scanf("%d", &order);
        switch (sorttype)
134     {
          case bookID: sql += "bookID ";break;
136       case type:sql += "type ";break;
          case bookName:sql += "bookName ";break;
138       case publisher:sql += "publisher ";break;
          case publishYear:sql += "publishYear ";break;
140       case author:sql += "author ";break;
          case price:sql += "price ";break;
142       case totalNum:sql += "totalNum ";break;
          case storageNum:sql += "storageNum ";break;
144       case updateTime:sql += "updateTime ";break;
          default:break;
146     }
        if (order == ascend) sql += "ASC;";
148     else if (order == descend) sql += "DESC;";
        int query = mysql_query(&mysql, sql.c_str());
150     if (query == 0)
        {
152       MYSQL_RES* partialResult = mysql_store_result(&mysql);
          if (partialResult)
154       {
            unsigned long partialNumber = (unsigned long)mysql_num_rows(partialResult);
156         cout << "number of books selected:" << partialNumber << endl;
            MYSQL_FIELD* fd = mysql_fetch_field(partialResult);
158         while (fd)
            {
160           cout << fd->name << "   ";
              fd = mysql_fetch_field(partialResult);
162         }
            cout << endl;
164         MYSQL_ROW row;
            while (row = mysql_fetch_row(partialResult))
```

```
166             {
                  unsigned long* lens = mysql_fetch_lengths(partialResult);
168               for (unsigned int i = 0; i < mysql_num_fields(partialResult); ++i)
                    if (row[i]) cout << row[i] << "\t";
170                 else cout << "NULL";
                  cout << endl;
172             }
              }
174           mysql_free_result(partialResult);
            }
176       }
      printf("input sort or not\n0:no sort and break directly\n1:sort\n");
178   cin >> DoSort;//scanf("%d", &DoSort);
    }
180 return;
}
182 void OnlyViewBooks()
{
184   int bookres = mysql_query(&mysql, "select * from `books`");
    if (bookres == 0)/*successful query*/
186   {
      MYSQL_RES* bookResult = mysql_store_result(&mysql);
188     int selectedType = 0;
      if (bookResult)
190     {
        unsigned long bookNumber = (unsigned long)mysql_num_rows(bookResult);
192       cout << "number of books:" << bookNumber << endl;
        MYSQL_FIELD* fd = mysql_fetch_field(bookResult);
194       while (fd)
        {
196       cout << fd->name << "\t"; fd = mysql_fetch_field(bookResult);
        }
198     cout << endl;
        MYSQL_ROW row;
200     while (row = mysql_fetch_row(bookResult))
        {
202       unsigned long* lens = mysql_fetch_lengths(bookResult);
          for (int i = 0; i < mysql_num_fields(bookResult); ++i)
204         if (row[i]) cout << row[i] << "\t";
            else cout << "NULL    " << "\t";
206       cout << endl;
        }
208     sortAllBooksByOrder(bookResult);

210     int conditionNumber = 0, id = -1;
        cout << "Please input the number of conditions" << endl; cin >> conditionNumber;
212     string sql = "select * from `books` where ", condition = " where ";
        string information, upperBound, lowBound;
214     if (conditionNumber == 1)
        {
216       printf("input type of partial select \n0: bookID\n1:type\n2:bookName\n3:publisher\n4
              :publishYear\n5:author\n6:price\n7:totalNum\n8:storageNum\n9:updateTime\n");
          cin >> selectedType;
218       int checkMode = Accurate;
          switch (selectedType)
220       {
            case bookID:sql += "bookID = ";printf("input bookID:\n");
222           cin >> information;
              sql += information + ";";
```

```
            condition += "bookID = ";
            condition += information;
            break;
        case type:sql += "type = ";printf("input type:\n");
            cin >> information;
            sql += "'" + information + "';";
            condition += "type = ";
            condition += " '" + information + "' ";
            break;
        case bookName:
            cout << "Please input check mode:" << endl << "0:accurate check" << endl << "1:
                fuzzy check" << endl;
            cin >> checkMode;
            if(checkMode == Accurate)
            {
                sql += "bookName = ";printf("input bookName:\n");
                cin >> information;
                sql += "'" + information + "';";
                condition += "bookName = ";
                condition += " '" + information + "' ";
            }
            else if (checkMode == Fuzzy)
            {
                sql += "bookName like"; printf("input bookName:\n");
                cin >> information;
                sql += "'" + information + "';";
                condition += "bookName like ";
                condition += " '" + information + "' ";
            }
            break;
        case publisher:
            cout << "Please input check mode:" << endl << "0:accurate check" << endl << "1:
                fuzzy check" << endl;
            cin >> checkMode;
            if (checkMode == Accurate)
            {
                sql += "publisher = ";printf("input publisher:\n");
                cin >> information;
                sql += information + ";";
                condition += "publisher = ";
                condition += "'" + information + "' ";
            }
            else if (checkMode == Fuzzy)
            {
                sql += "publisher like "; printf("input publisher:\n");
                cin >> information;
                sql += information + ";";
                condition += "publisher like ";
                condition += "'" + information + "' ";
            }
            break;
        case publishYear:
            cout << "Please input check mode:" << endl << "0:accurate check" << endl << "2:
                range check" << endl;
            cin >> checkMode;
            if (checkMode == Accurate)
            {
                sql += "publishYear = ";printf("input publishYear:\n");
                cin >> information;
```

```
280             sql += information + ";";
                condition += "publishYear = ";
282             condition += information ;
              }
284           else if (checkMode == Range)
              {
286             sql += "publishYear BETWEEN ";
                printf("input low and upper bounds of publishYear seperated by space:\n");
288             cin >> lowBound >> upperBound;
                sql += lowBound + " AND " + upperBound + ";";
290             condition += "publishYear BETWEEN " + lowBound + " AND " + upperBound + ";";
              }
292           break;
          case author:
294           cout << "Please input check mode:" << endl << "0:accurate check" << endl << "1:
                  fuzzy check" << endl;
              cin >> checkMode;
296           if (checkMode == Accurate)
              {
298             sql += "author = ";
                printf("input author:\n");
300             cin >> information;
                sql += "'" + information + "';";
302             condition += "author = ";
                condition += "'" + information + "' ";
304           }
              else if (checkMode == Fuzzy)
306           {
                sql += "author like ";
308             printf("input author:\n");
                cin >> information;
310             sql += "'" + information + "';";
                condition += "author like ";
312             condition += "'" + information + "' ";
              }
314           break;
          case price:
316           cout << "Please input check mode:" << endl << "0:accurate check" << endl << "2:
                  range check" << endl;
              cin >> checkMode;
318           if (checkMode == Accurate)
              {
320             sql += "price = ";
                printf("input price:\n");
322             cin >> information;
                sql += information + ";";
324             condition += "price = ";
                condition += information;
326           }
              else if (checkMode == Range)
328           {
                sql += "price BETWEEN ";
330             printf("input low and upper bounds of price seperated by space:\n");
                cin >> lowBound >> upperBound;
332             sql += lowBound + " AND " + upperBound + ";";
                condition += "price BETWEEN " + lowBound + " AND " + upperBound + ";";
334           }
              break;
336       case totalNum:
```

```
            cout << "Please input check mode:" << endl << "0:accurate check" << endl << "2:
                range check" << endl;
338         cin >> checkMode;
            if (checkMode == Accurate)
340         {
              sql += "totalNum = ";
342           printf("input totalNum:\n");
              cin >> information;
344           sql += information + ";";
              condition += "totalNum = ";
346           condition += information;
            }
348         else if (checkMode == Range)
            {
350           sql += "totalNum BETWEEN ";
              printf("input low and upper bounds of totalNum seperated by space:\n");
352           cin >> lowBound >> upperBound;
              sql += lowBound + " AND " + upperBound + ";";
354           condition += "totalNum BETWEEN " + lowBound + " AND " + upperBound + ";";
            }
356         break;
          case storageNum:
358         cout << "Please input check mode:" << endl << "0:accurate check" << endl << "2:
                range check" << endl;
            cin >> checkMode;
360         if (checkMode == Accurate)
            {
362           sql += "storageNum = ";
              printf("input storageNum:\n");
364           cin >> information;
              sql += information + ";";
366           condition += "storageNum = ";
              condition += information;
368         }
            else if (checkMode == Range)
370         {
              sql += "storageNum BETWEEN ";
372           printf("input low and upper bounds of storageNum seperated by space:\n");
              cin >> lowBound >> upperBound;
374           sql += lowBound + " AND " + upperBound + ";";
              condition += "storageNum BETWEEN " + lowBound + " AND " + upperBound + ";";
376         }
            break;
378       case updateTime:
            cout << "Please input check mode:" << endl << "0:accurate check" << endl << "2:
                range check" << endl;
380         cin >> checkMode;
            if (checkMode == Accurate)
382         {
              sql += "timestamp(updateTime) = ";
384           printf("input updateTime:\n");
              condition += "updateTime = ";
386           cin >> information; sql += "'" + information + " ";
              condition += "'" + information + " ";
388           cin >> information;
              sql += information + "';";
390           condition += information + "' ";
            }
392         else if (checkMode == Range)
```

```
            {
394           sql += "updateTime BETWEEN '";
              printf("input low and upper bounds of updateTime seperated by space:\n");
396           cin >> lowBound >> upperBound;
              sql += lowBound + "' AND '" + upperBound + "';";
398           condition += "updateTime BETWEEN '" + lowBound + "' AND '" + upperBound + "';"
                  ;
            }
400         break;
          default:
402         break;
        }
404   }
      else {
406     while (conditionNumber)
        {
408       printf("input type of partial select:\n1:type\n2:bookName\n3:publisher\n4:
              publishYear\n5:author\n6:price\n");
          cin >> id;
410       if ((id != type) && (id != bookName) && (id != publisher) && (id != publishYear)
              && (id != author) && (id != price))
            continue;
412       switch (id)
          {
414         case type:
              cout << "Please input type(accurate check)" << endl;
416           cin >> information;
              sql += " (type = '" + information + "') ";
418           condition += " (type = '" + information + "') ";
              break;
420         case bookName:
              cout << "Please input bookName(fuzzy check)" << endl;
422           cin >> information;
              sql += " (bookName like '" + information + "') ";
424           condition += " (bookName like '" + information + "') ";
              break;
426         case publisher:
              cout << "Please input publisher(fuzzy check)" << endl;
428           cin >> information;
              sql += " (publisher like '" + information + "') ";
430           condition += " (publisher like '" + information + "') ";
              break;
432         case publishYear:
              cout << "Please input low and upper bound of publishYear(range check) by space
                  " << endl;
434           cin >> lowBound >> upperBound;
              sql += " (publishYear BETWEEN " + lowBound + " AND " + upperBound + ") ";
436           condition += " (publishYear BETWEEN " + lowBound + " AND " + upperBound + ") "
                  ;
              break;
438         case author:
              cout << "Please input author(fuzzy check)" << endl;
440           cin >> information;
              sql += " (author like '" + information + "') ";
442           condition += " (author like '" + information + "') ";
              break;
444         case price:
              cout << "Please input low and upper bound of price(range check) by space" <<
                  endl;
```

```
446                 cin >> lowBound >> upperBound;
                    sql += " (price BETWEEN " + lowBound + " AND " + upperBound + ") ";
448                 condition += " (price BETWEEN " + lowBound + " AND " + upperBound + ") ";
                    break;
450               }
                  if (conditionNumber != 1)
452               {
                    sql += " AND ";
454                 condition += " AND ";
                  }
456               conditionNumber--;
                }
458             //sql += ";"; condition += ";";
              }
460
462           int query = mysql_query(&mysql, sql.c_str());
              if (query == 0)
464           {
                MYSQL_RES* partialResult = mysql_store_result(&mysql);
466             if (partialResult)
                {
468               unsigned long partialNumber = (unsigned long)mysql_num_rows(partialResult);
                  cout << "number of books selected:" << partialNumber << endl;
470               MYSQL_FIELD* fd = mysql_fetch_field(partialResult);
                  while (fd)
472               {
                    cout << fd->name << "    ";
474                 fd = mysql_fetch_field(partialResult);
                  }
476               cout << endl;
                  MYSQL_ROW row;
478               while (row = mysql_fetch_row(partialResult))
                  {
480                 unsigned long* lens = mysql_fetch_lengths(partialResult);
                    for (int i = 0; i < mysql_num_fields(partialResult); ++i)
482                   if (row[i]) cout << row[i] << "\t";
                      else cout << "NULL";
484                 cout << endl;
                  }
486               sortPartialBooksByOrder(partialResult,condition);
                }
488             mysql_free_result(partialResult);
              }
490           else {
                cout << "error";
492           }
            }
494         mysql_free_result(bookResult);
          }
496     }
      void output(MYSQL_RES *result)
498   {
        if (result)
500     {
          unsigned long Number = (unsigned long)mysql_num_rows(result);
502       cout << "number of selection:" << Number << endl;
          MYSQL_FIELD* fd = mysql_fetch_field(result);
504       while (fd)
```

```
    {
506   cout << fd->name << "    ";fd = mysql_fetch_field(result);
    }
508 cout << endl;
    MYSQL_ROW row;
510 while (row = mysql_fetch_row(result))
    {
512   unsigned long* lens = mysql_fetch_lengths(result);
      for (int i = 0; i < mysql_num_fields(result); ++i)
514     if (row[i]) cout << row[i] << "\t";
        else cout << "NULL" << "\t";
516   cout << endl;
    }
518 }
}
520 void dealWithOneBookIn(string &BookID , string &Type, string &BookName,
            string& Publisher, string &PublishYear, string &Author
522         , string& Price, string &Num)
{
524 string sql = "select * from `books` where ((bookID = "
    + BookID + ") and (bookName = '" + BookName + "')";
526 if (Type != NULLRepresent) sql += " and (type is null or type = '" + Type + "')";
  if (Publisher != NULLRepresent) sql += " and (publisher is null or publisher = '" +
      Publisher + "')";
528 if (PublishYear != NULLRepresent) sql += " and (publishYear is null or publishYear = " +
      PublishYear + ")";
  if (Author != NULLRepresent) sql += " and (author is null or author = '" + Author + "')";
530 if (Price != NULLRepresent) sql += " and (price is null or price = " + Price + ")";
  sql += ");";
532 /*
    + " and (type is null or type = " + "'" + Type + "')"
534   + " and (publisher is null or publisher = " + "'" + Publisher + "')"
    + " and (publishYear is null or publishYear = " + PublishYear + ")"
536   + " and (author is null or author = " + "'" + Author + "')"
    + " and (price is null or price = " + Price + "));";
538 */
  int res = mysql_query(&mysql, sql.c_str());
540 if (res == 0)
  {
542   MYSQL_RES* result = mysql_store_result(&mysql);
    if (result)
544   {
      unsigned long number = (unsigned long)mysql_num_rows(result);
546   if (number == 0)/*unfound*/
      {
548     mysql_free_result(result);
        set(Type); set(BookName); set(Publisher); set(Author);
550     string insertsql = "insert into `books` values("
          + BookID + "," + Type + "," + BookName + ","
552       + Publisher + "," + PublishYear + "," + Author + ","
          + Price + "," + Num + "," + Num + ",now());";
554     int error = mysql_query(&mysql, insertsql.c_str());
        if (error)/*information error*/
556     {
          printf("insertion fail\n");
558     }
        else {
560       int temp = mysql_query(&mysql, "select * from `books` order by updateTime ASC;");
          if (temp == 0){
```

```
562             result = mysql_store_result(&mysql);
                output(result);mysql_free_result(result);
564           }
            }
566       }
          else if (number == 1)/*found existed*/
568       {
            mysql_free_result(result);
            string updatesql = "update `books` set totalNum = totalNum + " + Num + " where
                bookID = " + BookID + ";";
570         mysql_query(&mysql, updatesql.c_str());
            updatesql = "update `books` set storageNum = storageNum + " + Num + " where bookID =
                " + BookID + ";";
572         mysql_query(&mysql, updatesql.c_str());
            updatesql = "update `books` set updateTime = now() where bookID = " + BookID + ";";
574         mysql_query(&mysql, updatesql.c_str());
            int temp = mysql_query(&mysql, "select * from `books` order by updateTime ASC;");
576         if (temp == 0){
              result = mysql_store_result(&mysql);
578           output(result); mysql_free_result(result);
            }
580       }
      }
582 }
}
584 void BooksIn()
{
586   //select all book and show
    int bookres = mysql_query(&mysql, "select * from `books` order by updateTime ASC;");
588   if (bookres == 0)/*successful query*/
    {
590   MYSQL_RES* bookResult = mysql_store_result(&mysql);
      output(bookResult);
592   mysql_free_result(bookResult);
    }
594 int bookInMode = 0;
    printf("please input mode\n0:only get 1 book per time\n1:get book from file\n2:delete one
        kind of book\n");
596 cin >> bookInMode;
    if (bookInMode == OneBook)
598 {
      printf("input book info(BookID,Type,BookName,Publisher,PublishYear,Author,Price,Num)
          seperated by space:\n");
600   string BookID,Type,BookName, Publisher, PublishYear, Author, Price, Num;
      cin >> BookID >> Type >> BookName >> Publisher >> PublishYear >> Author >> Price >> Num
          ;
602   dealWithOneBookIn(BookID, Type, BookName, Publisher, PublishYear, Author, Price, Num);
    }
604 else if (bookInMode == ManyBooks)
    {
606   printf("input filepath\n");
      string filepath;  cin >> filepath;
608   ifstream input;   input.open(filepath.c_str(), ios::in);
      while (!input.eof())//bug
610   {
        string BookID, Type, BookName, Publisher, PublishYear, Author, Price, Num;
612     input >> BookID >> Type >> BookName >> Publisher >> PublishYear >> Author >> Price >>
            Num;
        dealWithOneBookIn(BookID, Type, BookName, Publisher, PublishYear, Author, Price, Num);
614
```

```
    }
    input.close();
    int temp = mysql_query(&mysql, "select * from `books` order by updateTime ASC;");
    if (temp == 0) {
      MYSQL_RES* result = mysql_store_result(&mysql);
      output(result); mysql_free_result(result);
    }
  }
  else if (bookInMode == DeleteOneKind)
  {

    printf("input bookid to be deleted:\n");
    string id; cin >> id;
    string checkBookNotReturned;
    /*check all book related to the card are returned or not*/
    checkBookNotReturned = "select * from `records` where (bookID = " + id
      + ") AND (returnDate is null);";
    int notReturned = mysql_query(&mysql, checkBookNotReturned.c_str());
    if (notReturned == 0)
    {
      MYSQL_RES* temp = mysql_store_result(&mysql);
      if (temp)
      {
        unsigned long number = (unsigned long)mysql_num_rows(temp);
        if (number != 0)
        {
          cout << "not returned book--delete book fails!" << endl;
          return;
        }
      }
      mysql_free_result(temp);
    }


    mysql_query(&mysql, "SET foreign_key_checks = 0; ");
    string sql = "delete from `books` where bookID = " + id + ";";
    int deleteres = mysql_query(&mysql, sql.c_str());
    mysql_query(&mysql, "SET foreign_key_checks = 1; ");
    if (deleteres == 0)
    {
      int temp = mysql_query(&mysql, "select * from `books`;");
      if (temp == 0)
      {
        MYSQL_RES* bookResult = mysql_store_result(&mysql);
        output(bookResult);
        mysql_free_result(bookResult);
      }
    }
    else {
      printf("delete book error\n");
    }
  }
}
void ManageCard()
{
  int res = mysql_query(&mysql, "select * from `cards`;");
  if (res == 0)
  {
    MYSQL_RES* cardResult = mysql_store_result(&mysql);
```

```
674     output(cardResult);
        mysql_free_result(cardResult);
676   }
      int cardMode = 0;
678   cout << "please input mode\n0:add new card\n1:delete card" << endl;
      cin >> cardMode;
680   if (cardMode == AddCard)
      {
682     string sql, cardID, name, department, Type;
        sql = "insert into `cards` values(";
684     printf("please input cardID, name, department, Type\n");
        cin >> cardID >> name >> department >> Type;
686     set(department); set(name); set(Type);
        sql += cardID + ","+ name + "," + department + ","
688       + Type + ",now());";
        cout << sql << endl;
690     int insertres = mysql_query(&mysql, sql.c_str());
        if (insertres == 0)
692     {
          int temp = mysql_query(&mysql, "select * from `cards`");
694       if (temp == 0)
          {
696         MYSQL_RES* cardResult = mysql_store_result(&mysql);
            output(cardResult);
698         mysql_free_result(cardResult);
          }
700     }
        else {
702       printf("insert card error\n");
        }
704   }
      else if (cardMode == DeleteCard)
706   {
        string sql, cardID, checkBookNotReturned;
708     cout << "please input cardID to be deleted" << endl;
        cin >> cardID;
710     /*check all book related to the card are returned or not*/
        checkBookNotReturned = "select * from `records` where (cardID = " + cardID
712       + ") AND (returnDate is null);";
        int notReturned = mysql_query(&mysql, checkBookNotReturned.c_str());
714     if (notReturned == 0)
        {
716       MYSQL_RES* temp = mysql_store_result(&mysql);
          if (temp)
718       {
            unsigned long number = (unsigned long)mysql_num_rows(temp);
720         if (number != 0)
            {
722           cout << "not returned book--delete card fails!" << endl;
              return;
724         }
          }
726       mysql_free_result(temp);
        }
728
        string cardExist = "select * from `cards` where cardID = " + cardID + ";";//card exist
            or not
730     int cardExistenceCheck = mysql_query(&mysql, cardExist.c_str());//get card;
        if (cardExistenceCheck == 0)
```

```
732     {
          MYSQL_RES* temp = mysql_store_result(&mysql);
734       if (temp)
          {
736         unsigned long number = (unsigned long)mysql_num_rows(temp);
            if (number == 0)
738         {
              cout << "card not exist--delete fails!" << endl;
740           return;
            }
742       }
          mysql_free_result(temp);
744     }


746


748


750     mysql_query(&mysql, "SET foreign_key_checks = 0; ");
        sql = "delete from `cards` where cardID = ";
752     sql += cardID + ";";
        int deleteres = mysql_query(&mysql, sql.c_str());
754     mysql_query(&mysql, "SET foreign_key_checks = 1; ");// "SET foreign_key_checks = 1; "
        if (deleteres == 0)
756     {
          int temp = mysql_query(&mysql, "select * from `cards`;");
758       if (temp == 0)
          {
760         MYSQL_RES* cardResult = mysql_store_result(&mysql);
            output(cardResult);
762         mysql_free_result(cardResult);
          }
764     }
        else {
766       cout << "delete card error!" << endl;
        }
768   }
}
770 void BorrowBooks(string adminID)
{
772   int bookres = mysql_query(&mysql, "select * from `books`;");
      if (bookres == 0)
774   {
        MYSQL_RES* bookResult = mysql_store_result(&mysql);
776     if(bookResult) output(bookResult);
        mysql_free_result(bookResult);
778   }// show books left now
      string bookid,cardid;
780   cout << "please input bookID" << endl; cin >> bookid;
      cout << "please input cardID" << endl; cin >> cardid;
782   string sql;
      sql = "select * from `cards` where cardID = " + cardid + ";";//card exist or not
784   int flag = mysql_query(&mysql, sql.c_str());//get card;
      if (flag == 0)
786   {
        MYSQL_RES* temp = mysql_store_result(&mysql);
788     if (temp)
        {
790       unsigned long number = (unsigned long)mysql_num_rows(temp);
```

```
      if (number == 0)
      {
        cout << "card not exist--borrow fails!" << endl;
        return;
      }
    }
    mysql_free_result(temp);
  }
  sql = "select * from `records` where (cardID = " + cardid + " and returnDate is null);";//
      not returned books?
  flag = mysql_query(&mysql, sql.c_str());
  if (flag == 0)
  {
    MYSQL_RES* temp = mysql_store_result(&mysql);
    if (temp)
    {
      unsigned long number = (unsigned long)mysql_num_rows(temp);
      if (number != 0)
      {
        cout << "not returned book--borrow fails!" << endl;
        return;
      }
    }
    mysql_free_result(temp);
  }
  string storagenum;
  sql = "select storageNum from `books` where bookID = " + bookid + ";";
  flag = mysql_query(&mysql, sql.c_str());//get storage
  if (flag == 0)
  {
    MYSQL_RES* temp = mysql_store_result(&mysql);
    if (temp)
    {
      unsigned long number = (unsigned long)mysql_num_rows(temp);
      if (number == 0)
        cout << "get storage num fails" << endl;
      else {
        MYSQL_FIELD* fd = mysql_fetch_field(temp);
        while (fd)
          fd = mysql_fetch_field(temp);
        MYSQL_ROW row = mysql_fetch_row(temp);
          unsigned long* lens = mysql_fetch_lengths(temp);
          storagenum = row[0];
      }
    }
    mysql_free_result(temp);
  }
  if (storagenum == "0")/*no storage more*/
  {
    cout << "error--storage is 0!" << endl; return;
  }
  else {/*bug*/
    string updatesql = "update `books` set storageNum = storageNum - 1 where bookID = " +
        bookid + ";";
    mysql_query(&mysql, updatesql.c_str());
    //new a record
    int getcardreturn = mysql_query(&mysql, "select * from `records`"),cardnum = 0;
    if (getcardreturn == 0)
    {
```

```
848         MYSQL_RES* temp = mysql_store_result(&mysql);
            if (temp) cardnum = (unsigned long)mysql_num_rows(temp);
850         mysql_free_result(temp);
            cardnum++;
852         string insertsql = "insert into `records` values(" + to_string(cardnum)
              + "," + cardid + "," + bookid + ",now(),null," + adminID + ");";
854         mysql_query(&mysql, insertsql.c_str());
            int show = mysql_query(&mysql, "select * from `records`;");
856         if (show == 0)
            {
858           temp = mysql_store_result(&mysql);
              if (temp) output(temp);
860           mysql_free_result(temp);
            }
862       }
    }
864 }
  void ReturnBooks(string adminID)
866 {
    int bookres = mysql_query(&mysql, "select * from `books`;");
868 if (bookres == 0)
    {
870   MYSQL_RES* bookResult = mysql_store_result(&mysql);
      if (bookResult) output(bookResult);
872   mysql_free_result(bookResult);
    }// show books left now
874 string bookid, cardid;
    printf("please input bookID\n"); cin >> bookid;
876 printf("please input cardID\n"); cin >> cardid;
    string sql = "select * from `records` where (cardID = "
878   + cardid + " and bookID = " + bookid
      + " and returnDate is null);";
880 int num = 0;//num of records of borrow
    int flag = mysql_query(&mysql, sql.c_str());//get records of borrow
882 if (flag == 0)
    {
884   MYSQL_RES* temp = mysql_store_result(&mysql);
      if (temp)
886   {
        num = (unsigned long)mysql_num_rows(temp);
888     if (num == 0)//no records
        {
890       printf("no records exist--fail\n");
          return;
892     }
        else {/*records exist*/
894       string booksql = "update `books` set storageNum = storageNum + 1 where bookID = " +
              bookid + ";";
          mysql_query(&mysql, booksql.c_str());
896       string recordsql = "update `records` set returnDate = now() where (bookID = "
            + bookid + " and cardID = " + cardid + " and returnDate is null);";
898       mysql_query(&mysql, recordsql.c_str());
          int show = mysql_query(&mysql, "select * from `records`;");
900       if (show == 0)
          {
902         MYSQL_RES *showrecords = mysql_store_result(&mysql);
            if (showrecords) output(showrecords);
904         mysql_free_result(showrecords);
          }
```

```
906         }
      }
908     mysql_free_result(temp);
    }
910 }
int main()
912 {
  MYSQL_RES* result = NULL;//
914  mysql_init(&mysql);
  if (!mysql_real_connect(&mysql, host, user, password, database, port, 0, 0))
916  {
    cout << "fail" << endl; exit(1);
918  }
  else
920  {
    cout << "success" << endl;
922    mysql_query(&mysql, "SET NAMES GBK");
    int goon = 1;
924    while (goon)
    {
926      cout << "please input mode" << endl << "0:only check books" << endl << "1:admin login"
           << endl;
      cin >> mode;
928      if (mode == OnlyCheckBooks) {
        OnlyViewBooks();
930      }
      else if (mode == AdminLogin) {
932        unsigned long number = 0;/*check admin in or not in*/
        string adminID;
934        string password;
        cout << "input ID:" << endl; cin >> adminID;
936        cout << "input password:" << endl; cin >> password;/*password has no blank*/
        string sql = "select * from `admin` where adminID = " + adminID + " and password = '
            " + password + "';";
938        int adminFind = mysql_query(&mysql, sql.c_str());
        MYSQL_RES* adminResult = mysql_store_result(&mysql);
940        if (adminResult) number = (unsigned long)mysql_num_rows(adminResult);
        while (number == 0)
942        {
          cout << "no such admin exist\n";
944          cout << "input ID:" << endl; cin >> adminID;
          cout << "input password:" << endl; cin >> password;/*password has no blank*/
946          sql = "select * from `admin` where adminID = " + adminID + " and password = '" +
              password + "';";
          adminFind = mysql_query(&mysql, sql.c_str());
948          adminResult = mysql_store_result(&mysql);
          if (adminResult) number = (unsigned long)mysql_num_rows(adminResult);
950        }
        cout << "admin " + adminID + " successful login!" << endl;
952        int adminPower = 0;
        cout << "input things to do:\n-1:exit login\n0:books In\n1:borrow Books\n2:return
            Books\n3:card Manage\n";
954        cin >> adminPower;
        while (adminPower != exitLogin)
956        {
          switch (adminPower)
958          {
            case booksIn:
960            BooksIn();
```

```
                break;
962             case borrowBooks:
                BorrowBooks(adminID);
964             break;
                case returnBooks:
966             ReturnBooks(adminID);
                break;
968             case cardManage:
                ManageCard();
970             break;
                default:break;
972           }
              cout << "input things to do:\n-1:exit login\n0:books In\n1:borrow Books\n2:return
                  Books\n3:card Manage\n";
974           cin >> adminPower;
            }
976       }
          else {
            cout << "mode error" << endl;
          }
980       cout << "Please choose whether to go on or exit:" << endl << "0:exit" << endl << "1:
              goon" << endl;
          cin >> goon;
982     }
    }
984 return 0;
}
```

3: 完整源代码

6 参考资料

C++操作MySQL

VS2019 C++连接MySQL