

图书管理系统设计

课程：数据库系统原理
指导老师：陈根才
姓名：黄靖雅
学号：3110101593
专业：数字媒体技术

1 综述

1.1 实验内容

图书管理系统通过数据库系统来完成图书馆对图书和用户的管理，及用户对图书馆的利用，在图书数字化管理的现在有很大的现实意义。本次实验要求设计并实现一个具有图书入库、查询、借书、还书、借书证管理等功能的精简的图书管理系统，通过对数据库模型的设计规划，及整个程序的编写实现，加深对数据库原理和应用的理解。

1.2 实验要求

对象名称	包含属性
书	书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存
借书证	卡号, 姓名, 单位, 类别 (教师 学生等)
管理员	管理员 ID, 密码, 姓名, 联系方式
借书记录	卡号, 书号, 借期, 还期, 经手人 (管理员 ID)

模块名称	功能描述
管理员登陆	输入管理员 ID, 密码; 登入系统 或 返回 ID/密码 错误.

图书入库	<p>1. 单本入库</p> <p>2. 批量入库 (方便最后测试)</p> <p>图书信息存放在文件中, 每条图书信息为一行. 一行中的内容如下</p> <p>(书号, 类别, 书名, 出版社, 年份, 作者, 价格, 数量)</p> <p>Note: 其中 年份、数量是整数类型 ; 价格是两位小数类型 ; 其余为字符串类型</p> <p>Sample :</p> <p>(book_no_1, Computer Science, Computer Architecture, xxx, 2004, xxx, 90.00, 2)</p>
图书查询	<p>要求可以对书的 类别, 书名, 出版社, 年份(年份区间), 作者, 价格(区间) 进行查询. 每条图书信息包括以下内容:</p> <p>(书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存)</p> <p>可选要求: 可以按用户指定属性对图书信息进行排序. (默认是书名)</p>
借书	<p>1.输入借书证卡号</p> <p>显示该借书证所有已借书籍 (返回, 格式同查询模块)</p> <p>2.输入书号</p> <p>如果该书还有库存, 则借书成功, 同时库存数减一。</p> <p>否则输出该书无库存, 且输出最近归还的时间。</p>
还书	<p>1.输入借书证卡号</p> <p>显示该借书证所有已借书籍 (返回, 格式同查询模块)</p> <p>2.输入书号</p> <p>如果该书在已借书籍列表内, 则还书成功, 同时库存加一。</p>

	否则输出出错信息.
借书证管理	增加或删除一个借书证.

1.3 开发环境

这里使用了 Microsoft SQL Server 2008 R2 来搭建数据库,在 Microsoft Visual Studio 2010 中编写图书管理系统的整个程序,通过 ODBC 接口访问已经建好的本地数据库进行连接、查询、更新等操作,并用 MFC (Visual C++) 类库来实现图形化界面。程序开发测试环境是 Windows 7 (64 bits),在 VS2010 中进行了 Win32 平台的生成,此环境下可以正常地设计和运行。

1.4 技术介绍

1.4.1 Microsoft SQL Server

SQL Server 是一个关系数据库管理系统。它最初是由 Microsoft、Sybase 和 Ashton-Tate 三家公司共同开发的,后来由 Microsoft 移植到 Windows NT 系统上。这里使用了比机房预装的版本更为先进的 2008 R2,它的性能和用户交互更加出色。通过它的两个图形化界面 SQL Server 配置管理器及 SQL Server Management Studio,可以进行数据库访问方式的设置和数据库搭建、查询、更新的操作。

1.4.2 MFC (VC++)

VC++即 Visual C++,是由微软提供的用 C++语言编写程序的工具平台。VC++不仅是一个编译器,更是一个集成开发环境,包括编辑器、调试器和编译器等。MFC 全称 Microsoft

Foundation Classes ,也就是微软基础类库。它是 VC++ 的核心 ,是 C++ 与 Windows API 的结合 ,其中包含大量 Windows 句柄封装类和很多 Windows 的内建控件和组件的封装类 ,很彻底的用 C++ 封装了 Windows SDK(Software Development Kit 软件开发工具包) 中的结构和功能 ,还提供了一个应用程序框架 ,此应用程序框架为软件开发者完成了一些例行化的工作 ,比如各种窗口、工具栏、菜单的生成和管理等。使用 MFC 类库进行开发可以更容易地完成图形界面的实现 ,更多地关注系统的设计和功能的实现。

1.4.3 ODBC

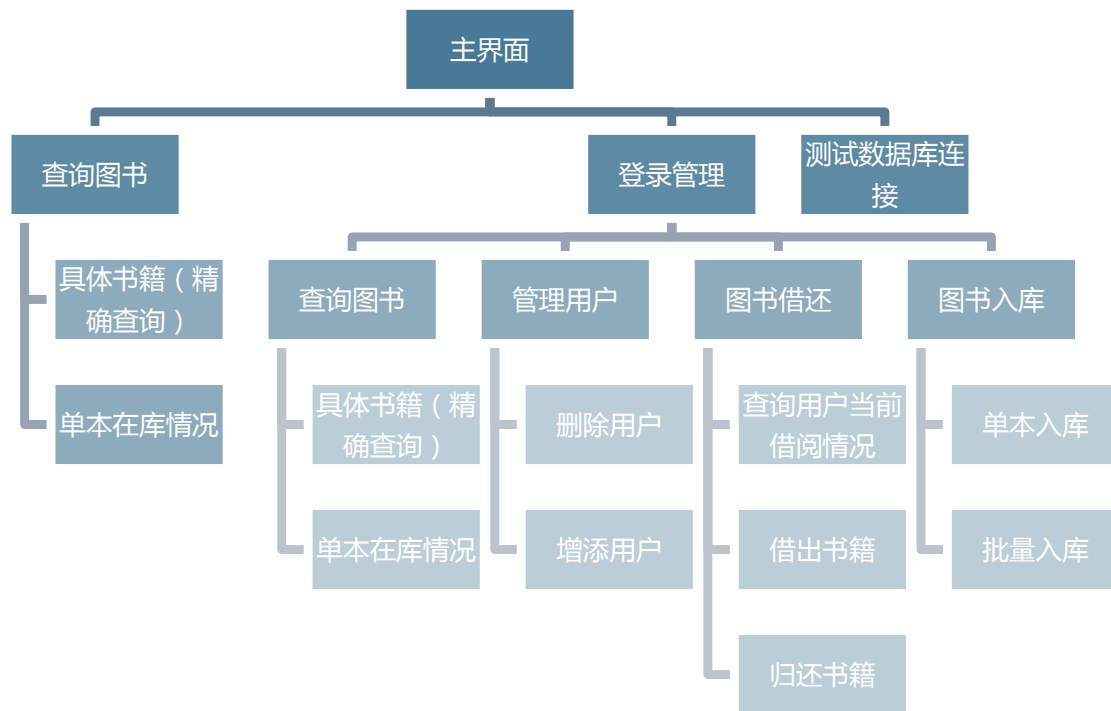
ODBC (Open Database Connectivity , 开放数据库互连) 是微软公司开放服务结构 (WOSA , Windows Open Services Architecture) 中有关数据库的一个组成部分 , 它建立了一组规范 , 并提供了一组对数据库访问的标准 API (应用程序编程接口) , 这些 API 利用 SQL 来完成其大部分任务。而 ODBC 本身也提供了对 SQL 语言的支持 , 用户可以直接将 SQL 语句传送给 ODBC。通过部署 DNS , 将本地数据库设为程序的数据源后 , 就能直接使用 ODBC 实现程序和数据库之间的通讯工作。

2 总体架构

2.1 系统架构

整个图书管理系统可以看作用户交互界面和后端处理模型两个部分。通过 Visual Studio 2010 提供的图形界面 , 创建基于对话框的用户交互界面 , 对每个对话框作为一个单独的类进行实例化 , 在用户进行操作的时候 , 读入对话框中的类成员变量并执行对应的类成员函数。而后端处理模型则是通过 ODBC 接口与 SQL Server 连接 , 完成数据库通讯、数据处理和界面交互的工作。

2.2 功能图

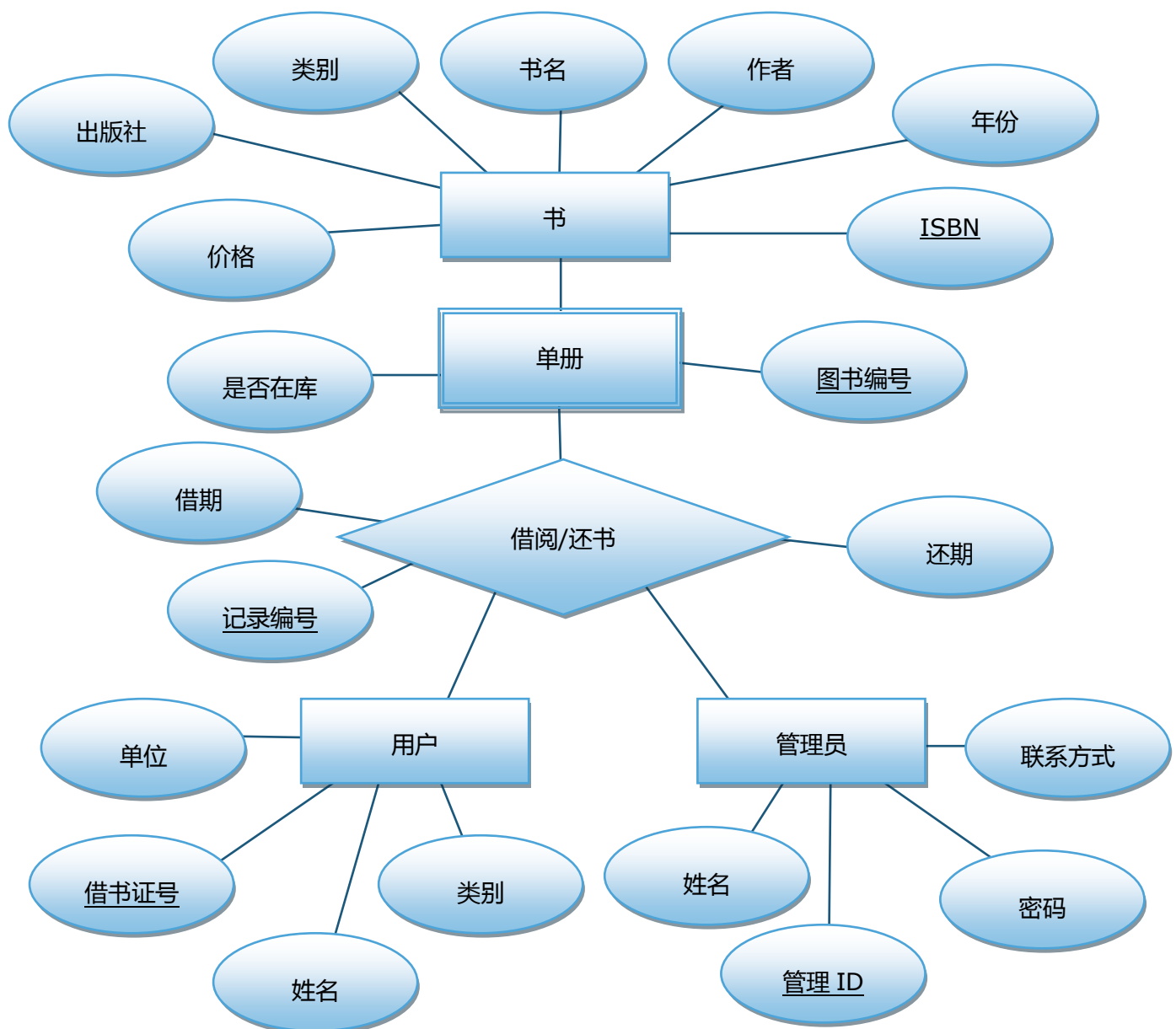


3 数据库设计

数据库采用 E-R 模型构建，包括 5 个表和 1 个视图，所有表和视图的命名均加上前缀 lib_

作为区分，以便数据库以后可以被用于其他应用程序。

3.1 数据库结构图



3.2 表和视图

3.2.1 表 lib_book_info

储存不同的书籍独有的信息，包括 ISBN、类别、书名、出版社、年份、作者、价格。

字段名	数据类型	长度	允许空	属性	说明
ISBN	char	17		Primary	
category	varchar	15			
title	varchar	60			
publisher	varchar	50			
pub_year	int				
author	varchar	60			
price	numeric(12,2)				

3.2.2 表 lib_book_list

把相同书籍的不同册别看作单独的个体，储存不同册别的全馆唯一编号和在库情况。

字段名	数据类型	长度	允许空	属性	说明
bno	int			Primary	
ISBN	char	17		Foreign	
inStore	int				0=借出,1=在库

3.2.3 表 lib_admin

储存图书馆系统管理员的信息，包括管理员 ID、密码、姓名、联系方式。

字段名	数据类型	长度	允许空	属性	说明
lib_aid	int			Primary	
lib_psw	varchar	32			
name	varchar	30			
contact	varchar	30			不限于电话

3.2.4 表 lib_user

储存图书馆系统一般用户（持借书证者）的信息，包括卡号、姓名、单位、类别（教师、学生，但类别独有的权限如借阅上限等在这里并没有进行体现）。

字段名	数据类型	长度	允许空	属性	说明
lib_uid	int			Primary	
name	varchar	30			
organization	varchar	30			
lib_uclass	int				0=教师,1=学生

3.2.5 表 lib_record

储存书籍的借阅记录，包括卡号、书号、借期、还期、经手人（管理员 ID）、是否已还。这里的书号指的是单册图书的编号，而非该种图书的 ISBN。每条记录有唯一独立的 id，生成新记录的 id 比原先最大的记录 id 要大 1。

字段名	数据类型	长度	允许空	属性	说明
rid	int			Primary	
lib_uid	int			Foreign	
lib_aid	int			Foreign	
bno	int			Foreign	
date_borrow	datetime				自动获取系统当前时间
date_return	datetime		Yes		自动获取系统当前时间，初始值默认为 NULL
isReturned	int				0=未还,1=已还；初始值为 0

3.2.6 视图 lib_commonQuery

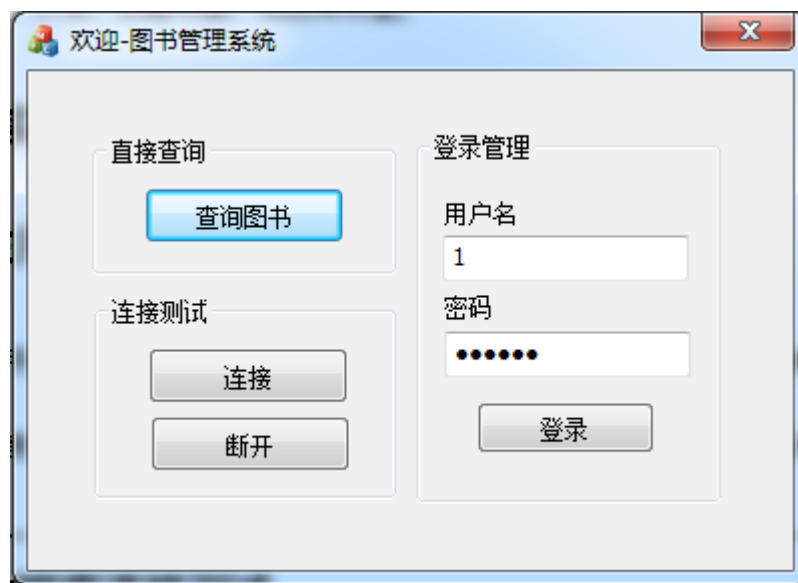
这个视图是为了比较方便地统计某种图书的总量和当前库存，将查询结果返还给用户，而根

据 lib_book_info 和 lib_book_list 这两张表创建的，包括书号、类别、书名、出版社、年份、作者、价格、总藏书量 (total_num)、当前库存量 (inventory)。其中原表中并没有的 total_num 和 inventory 两个属性是使用 COUNT() 语句，统计表 lib_book_list 中对应书籍的总数量和在庫状态为 1 的数量分别得到。

4 具体实现

4.1 主界面

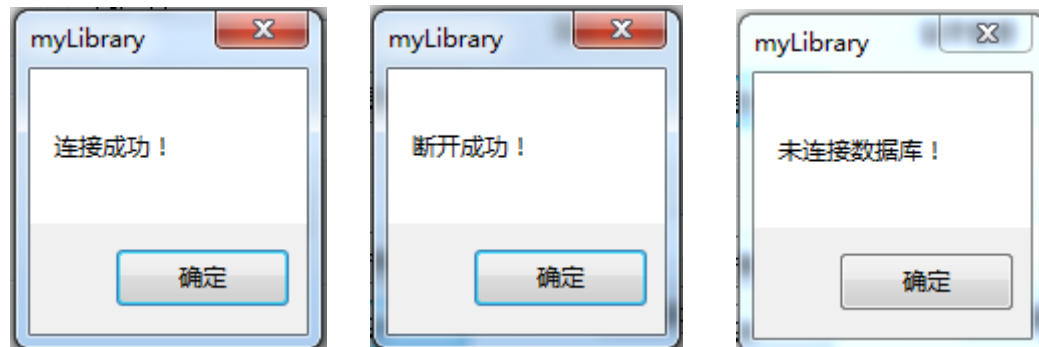
主界面主要用于为用户选择图书管理系统的使用方式提供入口。在不登录的情况下，可以直接进入图书查询的界面；而在登录的情况下，则可以进入管理界面。登录密码不明文显示。



4.2 数据库连接测试

该功能模块主要用于测试和断开当前数据库连接，便于登录和查询前的数据库通讯检验，以免在连接不上的时候花费多余的精力进行输入。要和数据库进行通讯，要在基类的头文件中包含 ODBC 类头文件“afxdb.h”及 SQL Server 提供的 SQL 类头文件“Sql.h”“Sqltext.h”及“Sqltypes.h”。

当数据库连接成功、断开连接成功或未连接数据库就进行断开操作时，跳出消息框对用户进行提示。



4.3 图书查询

图书查询是图书管理系统的基本功能。

对于某种图书的查询，根据提供的检索信息以 and 关系连接所有输入条件（类型为字符串的字段允许留空为“”，则不对该条件进行检索），对相应字段在视图 lib_commonQuery 中进行精确匹配（=）查询，最后以 ORDER BY title 的形式按书名的字典序进行排序，显示在 List Control 控件上。

而输入某种图书的查询，根据输入的 ISBN 在表 lib_book_list 和表 lib_book_info 中查询该种图书所有单册的在库情况，并在 List Control 控件上显示每一册的编号、ISBN、书名和是否在库（1 为在库、0 为借出），按照书籍编号排序。

这里为了得到返还的结果，要使用 CRecordSet（记录集）类来存放查询得到的记录。使用 CRecordSet::Open() 函数将输入的 SQL 语句传送给数据库，并打开结果记录集。记录集中数据以 CDBVariant 类型变量存储，通过 VariantToCString() 函数转换为 VC++ 可以使用的 CString 类型。这里通过一个 while 循环逐行、逐字段地遍历记录集，读入具体的数据并设置为 List Control 控件对应格的显示值。

SQL 语句：

查询图书：

```
select * from lib_commonQuery
where 1<=1 [and category = '%s'] [and title = '%s']
      [and publihser = '%s'] and pub_year between %d and %d
      [and author = '%s'] and price between '%f' and '%f' order by title
```

查询库存：

```
select L.bno, L.ISBN, I.title, L.inStore
from lib_book_list as L, lib_book_info as I
where L.ISBN = I.ISBN and L.ISBN = '%s' order by bno
```

书籍查询

书号	类别	书名	出版社	年
1	1	1	11	1
978-7-5327-3997-4	Fiction	1984	Shanghai Yiyen	200
978-7-5426-3054-4	Fiction	Animal Farm	Shanghai Sanlian	200
978-0-345-39180-3	Science Fiction	The Hitchhikers Guide to the...	Ballantine Books	200
978-7-80256-062-8	Education	TOEFL	QYpublish	200
2013	计算机	数据库系统	浙江大学	20

库存查询

书籍编号	书号	书名	在庫情况
16	2013	数据库系统	1
17	2013	数据库系统	1
18	2013	数据库系统	1
19	2013	数据库系统	1
20	2013	数据库系统	1

4.4 用户登录

这里只开放了管理员登录，因此只在表 lib_admin 中检查输入的用户名和密码是否存在且匹配，即 select 出所有 lib_aid 等于输入且 lib_psw 等于输入的元组，在记录集中打开，对它们进行计数，当数目大于 0 时判定该用户存在，登录成功。由于未考虑管理员注册的情况，因此暂不考虑得到多条记录的情况。当登录成功时，进入管理员操作界面，可以点击

按钮进入图书入库、借书/还书、借书证管理、图书查询四个功能窗口；当登录失败时，提示用户名或密码错误。其中，从管理员界面进入的“图书查询”与不登录时直接进入的“图书查询”为同一窗口，功能上没有区别。与此同时，全局变量 nowuid 记录当前登录的用户名，便于其他管理功能的操作。

SQL 语句：

```
select * from lib_admin where lib_aid = %d and lib_psw = '%s'
```



4.5 图书入库

管理员可以通过单本或批量的方式进行图书入库。

单本入库要求输入该种图书的书号、类别、书名、出版社、年份、作者、价格、数量，其中书号、类别、书名、出版社、作者不能为空（输入为""），且年份、数量不能为 0，年份、数量、价格不能小于 0。当输入不符合条件时，提示用户检查。同时，由于本系统并未设计删除入库图书的功能，在输入完成后，会提示用户再次确认，以免进行错误入库。

批量入库时构造一个打开文件对话框，通过 fstream 和 stringstream 的方式打开和读入 txt 文件。这里要求以每行为一种图书的信息输入，以半角逗号（,）为不同字段的分隔符，且遇到空格不停止输入，因此在 while 循环中，以 getline() 的方式读取 txt 文件，并以当前字

符 类型为 char)是否半角逗号为分歧点 通过 for 循环将每一行的数据分别读入到 CString 数组 temp[]中。若出现信息有误 (和单本入库的判定标准一致) 则跳过该行, 不进行该行的数据库更新操作, 继续读取下一行。

每种图书的入库对表 lib_book_info 和 lib_book_list 都需要进行更新。在 lib_book_info 中, 插入一行该种图书的信息。而在 lib_book_list 中, 根据输入的数量, 在 for 循环中循环插入行, 其中在库情况被设置为 1 (在库), 图书编号设置为当前最大编号加一 (MAX(bno)+1)。

SQL 语句 :

更新书籍总表 lib_book_info :

```
insert into lib_book_info values  
( '%s', '%s', '%s', '%s', %d, '%s', %lf)
```

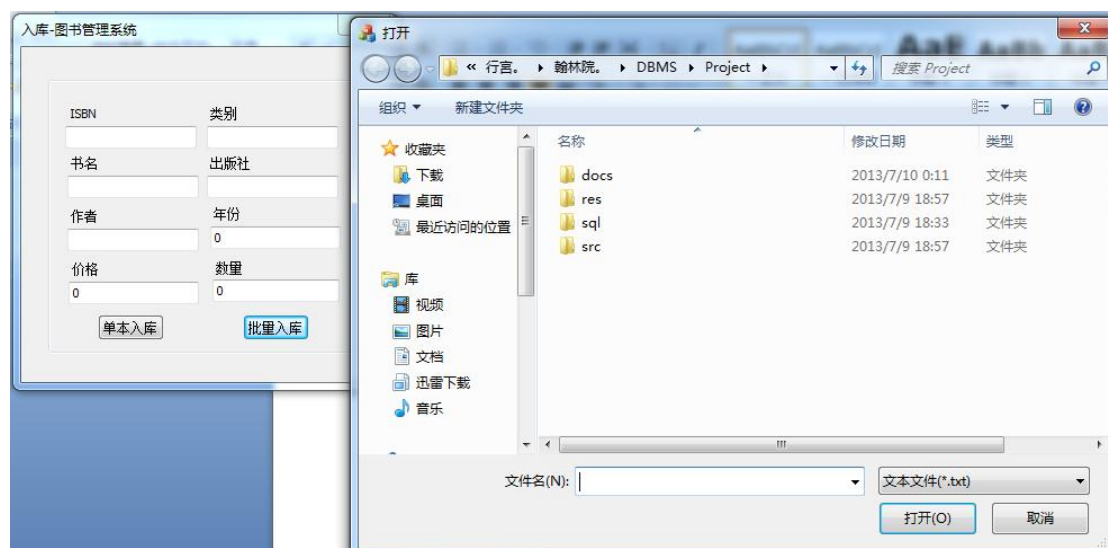
获得单册表 lib_book_list 中最大图书编号 :

```
select MAX(bno) from lib_book_list
```

更新单册表 lib_book_list :

```
insert into lib_book_list values (%d, '%s', 1)
```





4.6 借书证管理

管理员可以管理普通用户的注册信息，即增添或删除借书证。

增加借书证时，要求输入需要增加的借书证号、姓名、单位，并选择用户类别（有教师、学生两种选项，默认为教师）。新增借书证信息作为新的一行插入到表 lib_user 中。注意借书证号不能为 0，姓名、单位不能为空（""），否则提示用户输入相关信息。同时，若借书证号和已有冲突，则提示登记失败。

删除借书证时，为了慎重起见，要求输入当前管理账号的密码，以密文显示，同时在点击确定删除时弹出对话框进行确认。查询表 lib_admin 中是否有 lib_psw 与输入、lib_aid 与当前账号（由全局变量 nowuid 储存）同时吻合的记录，若有，则使用 delete 语句根据输入的借书证号删除表 lib_user 中对应的一行。

SQL 语句：

新增借书证：

```
insert into lib_user values (%d, '%s', '%s', %d)
```

确认管理密码：

```
select lib_psw from lib_admin where lib_aid = %d
```

删除借书证：

```
delete from lib_user where lib_uid = %d
```

借书证管理-图书管理系统

增加借书证 删除借书证

卡号
99999

姓名

单位

类别

教师

确定

myLibrary

请填写相关信息！

确定

借书证管理-图书管理系统

增加借书证 删除借书证

卡号
302

姓名
Kakyoin

单位
Literatruue Dept.

类别
学生

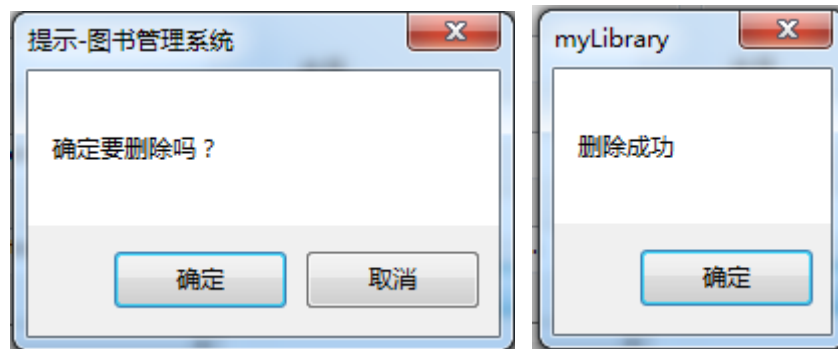
教师

确定

myLibrary

登记成功

确定



4.7 图书借还

和图书查询一样，图书借还也是图书管理系统的核心功能之一。在图书借还的窗口，管理员可以查询用户当前的借阅情况，并且为某一用户登记借书和还书。

查询借阅要求输入用户的借书证号，从表 `lib_record` 中找出 `lib_uid` 为该用户借书证号且在库状态为借出 (`isReturned = 0`) 的书籍，再到视图 `lib_commonQuery` 中取出该种书籍的信息，显示在 List Control 控件中。

借书要求输入借书证号和图书编号(单册)，其中图书编号可以在图书查询窗口中输入 ISBN 查询获得。在表 `lib_book_list` 中查询该册的在库情况 (`inStore`)，若为借出，则提示用户此书已借出；若为在库，则更新在库情况为借出，同时在表 `lib_record` 中插入一条记录。

还书同样也要求输入借书证号和图书编号。首先要检查该书是否在用户的借阅列表中，创建一个变量 `int exists = 0` 进行标记，`select` 出表 `lib_record` 中所有未归还且 `lib_uid` 等于该用户借书证号的图书编号，当其中有一条记录符合所还册编号时，变更 `exists` 的值为 1，即该书存在于用户的借阅列表中，否则认为不存在。若用户未借此书，则提示不在借阅列表中；若用户已借此书，则在表 `lib_book_list` 中更新该册为在库，同时在表 `lib_record` 中更新该册已还、还书时间为当前系统时间（使用 SQL 中 `GETDATE()` 语句得到）。

SQL 语句：

查询借书证借阅信息：

```
select distinct Q.*
from lib_commonQuery as Q, lib_book_list as L, lib_record as R
where R.bno = L.bno and R.lib_uid = %d and R.isReturned = 0 and Q.ISBN
= L.ISBN
order by title
```

检查该册是否被借走：

```
select instore from lib_book_list where bno = %d
```

借出单册：

```
insert into lib_record values
((select MAX(rid) from lib_record)+1, %d, %d,
%d, GETDATE(), default, 0)
update lib_book_list set instore = 0 where bno = %d;
```

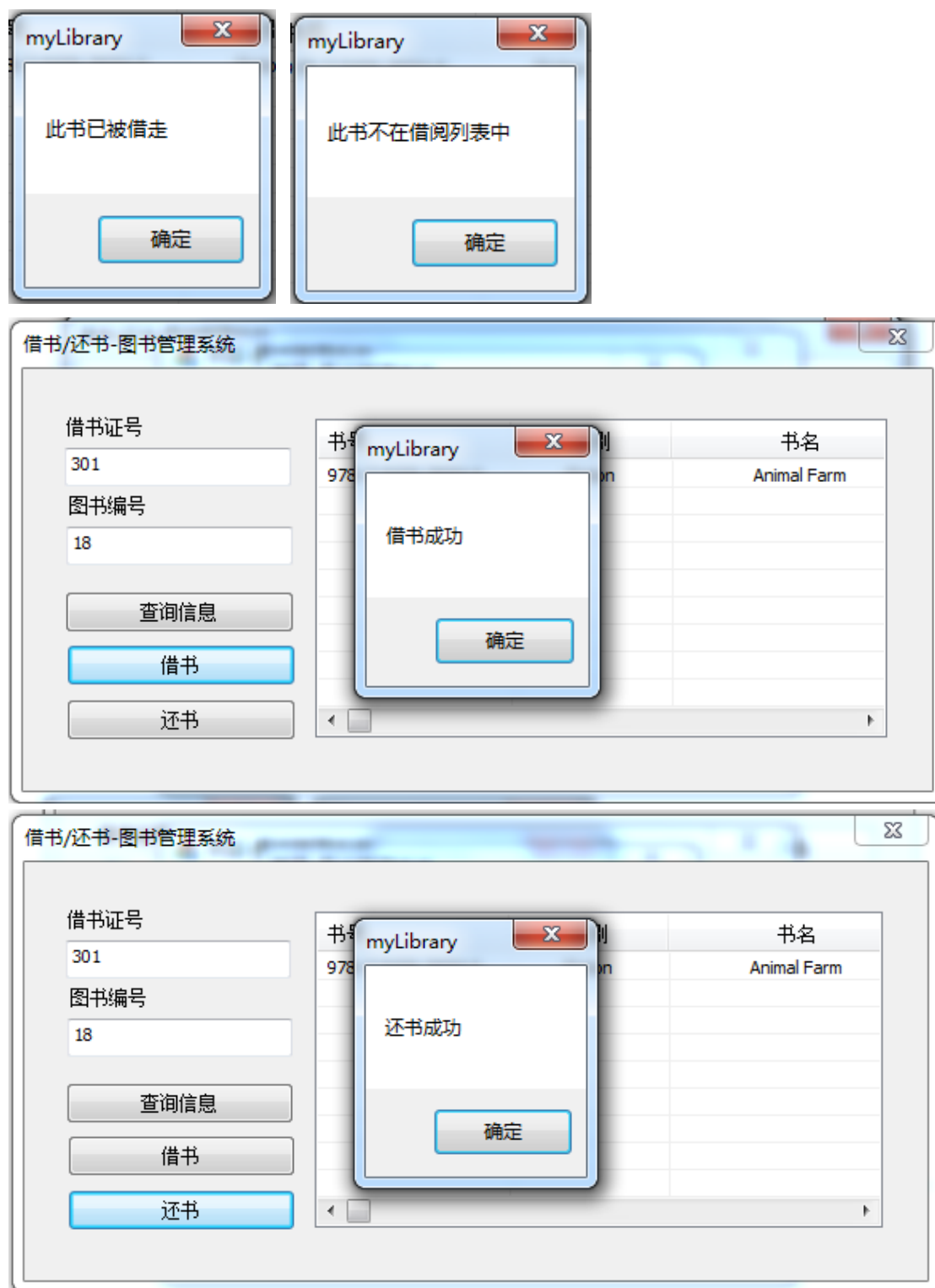
检查该册是否在用户借阅列表中：

```
select bno from lib_record
where lib_uid = %d and isReturned = 0
```

归还单册：

```
update lib_record set
date_return = GETDATE(), isReturned = 1
where rid = (select rid from lib_record where bno = %d and isReturned =
0)
update lib_book_list set instore = 1 where bno = %d;
```

书号	类别	书名
978-7-5426-3054-4	Fiction	Animal Farm



5 总结

5.1 程序优点

- 基本实现了实验要求的内容。
- 拥有图形界面，用户交互比较友好。
- 数据库结构与指导书略有不同，将每种书籍的单册当成不同个体处理，更符合实际情况，同时能保证借还记录唯一不被覆盖。也因此增加了查询单本库存的功能。
- 设置了一些输入限制，并向用户进行结果告知、操作提示和警告。

5.2 程序缺点

- 未能将实验要求的“否则输出该书无库存,且输出最近归还的时间”完全实现。实际上，相关代码已在文件 LAB.cpp 的 CLAB::OnBnClickedlendnew()函数的注释部分中写出，但实际运行时会出现“无可来源”的错误，故而舍弃。
- 一些应该在 SQL 中使用 Trigger 进行限制的条件，为了提供相应提示而放到程序实现中进行限制，使代码变得繁琐。
- 由于对 C++ 的运用尚不熟练，未能很好地利用类成员、外联函数等知识简化代码。

5.3 个人心得

这次大程带给我的收获很大。从搭建数据库、创建新的数据库角色和用户，到配置 SQL Server Browser 和 ODBC 数据源，再到用 C++ 连接数据库，我试了很多次、看了很多相关资料和例程才一点点地完成。在这个过程中，我对数据库的理解和应用水平大有提高。其实这也是我第一次独立用 MFC 写大程，在我学习和熟练使用 MFC 类库和 VC++ 的图形界面的过程中，我对 C++ 的理解和代码水平也有了很大的提高。