



**课 程 设 计**

课程名称\_\_数据库系统课程设计 \_

题目名称 图书借阅管理系统

专业班级 软件工程4班

学 号 3115005372

学生姓名 杨宇杰

指导教师 何晓桃

2018年 1 月 9 日

**目录**

[**一、引言 6**](#_Toc503266917)

[**1.1 课程设计选题 6**](#_Toc503266918)

[**1.2 课程设计目的 6**](#_Toc503266919)

[**1.3 本选题的设计背景 6**](#_Toc503266920)

[**1.4 相关技术介绍 6**](#_Toc503266921)

[**1.5 项目开发环境与工具 7**](#_Toc503266922)

[**二、需求分析 8**](#_Toc503266923)

[**2.1 信息需求 8**](#_Toc503266924)

[**2.2 系统需求概述 9**](#_Toc503266925)

[**2.3 功能需求 9**](#_Toc503266926)

[**2.4 安全性与完整性需求 10**](#_Toc503266927)

[**2.5 数据字典 11**](#_Toc503266928)

[**三、概念结构设计 15**](#_Toc503266929)

[**3.1 概念模型：ER图 15**](#_Toc503266930)

[**四、逻辑结构设计 16**](#_Toc503266931)

[**4.1 关系模型：所有表结构及范式分析等 16**](#_Toc503266932)

[**4.2 用户子模式 17**](#_Toc503266933)

[**4.3 应用程序功能模块图 17**](#_Toc503266934)

[**4.4 安全性（用户类别和权限）设计和实现说明 17**](#_Toc503266935)

[**4.5 完整性（主、外码和用户自定义的完整性约束）设计和说明 18**](#_Toc503266936)

[**五、数据库物理设计 18**](#_Toc503266937)

[**5.1 所有索引的定义 18**](#_Toc503266938)

[**5.2 数据的存放位置说明 19**](#_Toc503266939)

[**5.3 系统配置说明 19**](#_Toc503266940)

[**5.4 模块设计（模块IPO图） 20**](#_Toc503266941)

[**六、数据库实施 23**](#_Toc503266942)

[**6.1 数据库备份 23**](#_Toc503266943)

[**6.2 恢复方案 25**](#_Toc503266944)

[**七、数据库运行与维护 26**](#_Toc503266945)

[**7.1 系统使用说明 26**](#_Toc503266946)

[**7.2 系统测试方案和测试报告 31**](#_Toc503266947)

[**八、系统安装说明 35**](#_Toc503266948)

[**九、收获和体会 36**](#_Toc503266949)

[**十、附录 37**](#_Toc503266950)

# 引言

## 1.1 课程设计选题

《图书借阅管理系统》

## 1.2 课程设计目的

按照关系型数据库的基本原理，综合运用所学的基本知识，设计开发一个小型的管理信息系统。通过对一个实际问题的分析、设计与实现，将原理与应用相结合，使学生学会如何把书本上学到的知识用于解决实际问题，培养学生的动手能力，另一方面，使学生能深入理解和灵活掌握教学内容。

## 1.3 本选题的设计背景

当今时代是飞速发展的信息时代。在各行各业中离不开信息处理，这正是计算机被广泛应用于信息管理系统的环境。计算机的最大好处在于利用它能够进行信息管理。使用计算机进行信息控制，不仅提高了工作效率，而且大大的提高了其安全性。

尤其对于复杂的信息管理，计算机能够充分发挥它的优越性。计算机进行信息管理与信息管理系统的开发密切相关，系统的开发是系统管理的前提。本系统就是为了管理好图书馆信息而设计的。

图书馆作为一种信息资源的集散地，图书和用户借阅资料繁多，包含很多的信息数据的管理，现今己有很多图书馆都配备了图书借阅管理系统（在高校图书馆尤为常见），方便读者对所需图书的查找借阅及管理员对图书的管理。据调查，在未引入图书借阅管理系统之前，图书馆对信息管理的主要方式是基于文本、表格等纸介质的手工处理，对于图书借阅情况（如借书天数、超过限定借书时间的天数）的统计和核实等往往采用对借书卡的人工检查进行，对借阅者的借阅权限、以及借阅天数等用人工计算、手抄进行。数据信息处理工作量大，容易出错；由于数据繁多，容易丢失，且不易查找。总的来说，缺乏系统，规范的信息管理手段。因此，随着技术的发展，开发适合图书馆自身情况的图书借阅管理系统便成了一种趋势。本课设就是在这样的背景下，参考多数高校的图书借阅管理系统而进行开发的。

## 1.4 相关技术介绍

**1.数据库技术介绍：**数据库技术是信息系统的一个核心技术。是一种计算机辅助管理数据的方法，它研究如何组织和存储数据，如何高效地获取和处理数据。是通过研究[数据库](http://baike.baidu.com/view/1088.htm)的结构、存储、设计、管理以及应用的基本理论和实现方法，并利用这些理论来实现对[数据库](http://baike.baidu.com/view/1088.htm)中的[数据](http://baike.baidu.com/view/38752.htm)进行处理、分析和理解的技术。即：数据库技术是研究、管理和应用数据库的一门软件科学。

**2.Java开发技术：**[Java](http://baike.baidu.com/subview/29/12654100.htm)是一门[面向对象](http://baike.baidu.com/view/125370.htm)编程语言，不仅吸收了[C++](http://baike.baidu.com/view/824.htm)语言的各种优点，还摒弃了C++里难以理解的[多继承](http://baike.baidu.com/view/1160382.htm)、[指针](http://baike.baidu.com/view/159417.htm)等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。Java具有简单性、面向对象、[分布式](http://baike.baidu.com/subview/402382/19718738.htm)、[健壮性](http://baike.baidu.com/view/963152.htm)、[安全性](http://baike.baidu.com/view/421194.htm)、平台独立与可移植性、[多线程](http://baike.baidu.com/view/65706.htm)、动态性等特点，可编写[桌面应用程序](http://baike.baidu.com/view/2251864.htm)、[Web应用程序](http://baike.baidu.com/view/138452.htm)、[分布式系统](http://baike.baidu.com/view/991489.htm)和[嵌入式系统](http://baike.baidu.com/view/6115.htm)应用程序等项目，是当前使用最广的计算机语言之一。

**3.面向对象编程技术：**[面向对象编程](http://baike.baidu.com/view/10110.htm)（Object Oriented Programming，[OOP](http://baike.baidu.com/view/63596.htm)，面向对象程序设计）是一种计算机编程架构。OOP 的一条基本原则是计算机程序是由单个能够起到子程序作用的[单元](http://baike.baidu.com/subview/594567/8045349.htm)或对象组合而成。OOP 达到了[软件工程](http://baike.baidu.com/view/1659.htm)的三个主要目标：重用性、灵活性和扩展性。为了实现整体运算，每个对象都能够接收信息、处理数据和向其它对象发送信息。Java开发技术就是属于一种面向对象编程的技术。

**4.软件需求分析技术：**软件需求分析就是把软件计划期间建立的需求可行性分析求精和细化，分析各种可能的解法，并且分配给各个软件元素。需求分析是软件定义阶段中的最后一步，是确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求。

## 1.5 项目开发环境与工具

系统开发环境：Windows 10

编程开发语言：Java

Java JDK版本：JDK 1.7

编译软件： Eclipse

数据库软件：SQL Server 2014

# 需求分析

## 2.1 信息需求

用户信息

用户ID(ID)，

用户名(Username)，

用户密码(Password)，

用户权限(Power)

书籍信息

书籍ID(BookID)，

书名(BookName)，

出版社(Press)，

作者(Author)，

地址(Address)，

出版日期(PressDate)，

价格(Price)，

备注信息(Com)，

书本数目(books\_count)，

被借数(borrowed\_count)

书籍借阅信息

借阅序列号(ID)，

用户ID(ID),

学生姓名(StudentName)，

书籍ID(BookID),

书名(BookName)，

归还日期(ReturnDate)，

借书日期(BorrowDate)，

备注信息(Com)，

是否已归还(Is\_Returned)

## 2.2 系统需求概述

设计图书馆信息管理系统要考虑的问题很多。本程序的实例对实际系统作了适当的简化。需求概述如下：

1. 系统能对书籍信息进行管理。书籍的信息包括：书籍名称、出版社、作者、出版社地址、出版日期、价格、书籍数量、书籍借出数量。书籍管理功能包括：添加书籍、修改书籍信息、删除书籍。添加书籍时,可设定新书数目，几本完全相同的书籍作为一条记录管理。
2. 系统不需要管理借阅者除姓名外的信息。对每一名借阅者最多可借阅的书籍数目不做限定。
3. 借书时，管理人员记录借书者ID、借书者姓名、书籍ID、借阅书籍名称和借阅日期。
4. 还书时，管理人员记录还书者姓名、归还书籍名和还书日期。系统暂时不设定书籍的最长借阅时间。
5. 操作人员可以根据书籍名称、作者、出版社三者之一或者三者的任意组合查询书籍信息。系统暂时不具备模糊查询的功能。
6. 操作人员可以查询借阅某书籍的所有借阅者姓名；也可以查询借阅者借阅的所有书籍名称；还可以根据借阅者姓名、书籍名称查询借阅者的借书时间、还书时间。
7. 系统由图书馆工作人员操作、图书馆工作人员分为系统管理员、书籍管理员和学生三类。系统管理员具有所有操作权限；书籍管理员具有书籍管理、查询信息权限但不具备管理借书还书事宜及管理用户权限；学生具有管理借书还书、查询信息权限，但不具备管理书籍和用户的权限。
8. 系统管理员负责管理系统用户，可以增减用户、修改用户密码和删除用户。
9. 所有操作人员凭账号和密码登陆系统。

## 2.3 功能需求

* 用户管理：该功能又包含以下三项子功能：
* 用户登录
* 添加用户
* 修改用户密码
* 删除用户
* 书籍管理：书籍管理功能包括以下三项子功能：
* 添加书籍
* 修改书籍
* 删除书籍
* 借书管理：该功能包括以下两项子功能：
* 书籍出借
* 出借信息修改
* 还书管理：该功能包括以下两项子功能：
* 书籍还入
* 书籍还入信息修改
* 信息查询：该功能包括以下三项子功能：
* 书籍信息查询
* 借阅信息查询
* 系统用户列表

## 2.4 安全性与完整性需求

1. 实体完整性约束：关系中的主属性不能为空，在数据库中使用Primary Key实现；
2. 参照完整性约束：读者与书籍、出版社与书籍、读者与读者身份、借还情况与读者和书籍表之间采用级联更新和级联删除；
3. 用户自定义的完整性约束：规定姓名、书名和借书日期不为空。
4. 设计表间关系：用户表通过用户ID与表联系，书籍表通过书籍ID与表联系，借阅表通过书籍ID、读者姓名与书籍表和用户表联系；
5. 限制的设置：只有管理员可实现对用户及书籍信息的管理，其他实体都不行，防止系统遭到管理员外的人员破坏；用户或管理员必需登录帐号才能进行操作，防止非法人员进入使用系统。
6. 实现自主存取控制：数据库创建了两个登录名Yujie\_Yang和UserLogin，用Yujie\_Yang登录SQL Server，具有超级管理员权限，用UserLogin登录，则只具有部分权限。两个用户都只能对LibraryBase数据库进行操作，无法操作其他数据库，保证了其他数据库的安全性。

## 2.5 数据字典

1）数据项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 别名 | 数据  类型 | 数据长度 | 取值范围 | 简述 |
| 用户ID | ID | 整型 | 5位 | 由不同的数字组成 | 唯一标识用户身份 |
| 用户名 | UserName | 字符型 | 40位 | 只能由汉字或者英文字母组成 | 用户登录时的用户名 |
| 用户密码 | Password | 字符型 | 40位 | 40位以内数字或英文字母 | 用户登录密码，与用户账号对应使用 |
| 用户权限 | Power | 字符型 | 40位 | 系统管理员、书籍管理员、学生 | 标识登录用户的权限 |
| 书籍ID | ID | 整型 | 5位 | 由不同的数字组成 | 书籍的唯一标识 |
| 书名 | BookName | 字符型 | 20位 | 由汉字或英文字母组成 | 书籍的名字 |
| 出版社 | Press | 字符型 | 20位 | 由汉字或英文字母组成 | 书籍的出版社信息 |
| 作者 | Author | 字符型 | 20位 | 由汉字或英文字母组成 | 书籍的作者名 |
| 地址 | Address | 字符型 | 20位 | 由汉字或英文字母组成 | 书籍的出版地址 |
| 出版日期 | PressDate | 字符型 | 10位 | 日期表示 | 书籍的出版日期 |
| 价格 | Price | 字符型 | 10位 | 由数字和汉字组成 | 书籍的价格 |
| 备注 | Com | 字符型 | 20位 | 由字符，数字以及汉字组成 | 书籍的备注信息 |
| 总数量 | books\_count | 字符型 | 2位 | 由数字组成 | 每本书籍的总数量 |
| 被借书数量 | borrow\_count | 字符型 | 2位 | 由数字组成 | 每本书籍被借阅数量 |
| 还书日期 | ReturnDate | 字符型 | 10位 | 日期表示 | 每本书籍被借阅时间 |
| 借书日期 | BorrowDate | 字符型 | 10位 | 日期表示 | 每本书籍被归还时间 |
| 已还 | Is\_Returned | 字符型 | 2位 | 是、否 | 标识书籍是否被归还 |

2）数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 别名 | 含义 | 组成 |
| 用户表 | UserTable | 用户的基本信息 | 用户ID，用户名，用户密码，用户权限 |
| 书籍表 | books | 书籍的基本信息 | 书籍ID，书名，出版社，作者，地址，出版日期，价格，备注，总数量，被借书数量 |
| 借阅表 | booksBrowse | 借阅信息 | 借阅信息ID，学生姓名，书名，还书日期，借书日期，备注，已还 |

3）数据流

|  |  |
| --- | --- |
| 名称 | 用户信息 |
| 含义 | 系统管理员、借阅管理员、学生的基本信息 |
| 来源 | 用户输入电脑的用户名、密码、用户权限 |
| 去向 | 系统主菜单 |
| 组成 | 用户名，用户密码、用户权限 |

|  |  |
| --- | --- |
| 名称 | 书籍信息 |
| 含义 | 管理员录入的书籍信息 |
| 来源 | 管理员在电脑上录入的书籍信息 |
| 去向 | 书籍记录 |
| 组成 | 书籍ID，书名，出版社，作者，地址，出版日期，价格，备注，总数量，被借书数量 |

|  |  |
| --- | --- |
| 名称 | 借阅信息 |
| 含义 | 管理员录入的借阅信息 |
| 来源 | 管理员在电脑上录入的借阅信息 |
| 去向 | 借阅记录 |
| 组成 | ID，学生姓名，书名，还书日期，借书日期，备注，已还 |

|  |  |
| --- | --- |
| 名称 | 查询书籍信息 |
| 含义 | 查询图书管理系统中的书籍信息 |
| 来源 | 管理员在电脑上录入的书籍信息 |
| 去向 | 输出框中的书籍信息 |
| 组成 | 书名，出版社，作者，地址，出版日期，价格，备注 |

|  |  |
| --- | --- |
| 名称 | 查询借阅信息 |
| 含义 | 查询图书管理系统中的书籍借阅信息 |
| 来源 | 管理员在电脑上录入的书籍借阅信息 |
| 去向 | 输出框中的书籍借阅信息 |
| 组成 | 借阅者，书名，借阅日期，归还日期，备注 |

|  |  |
| --- | --- |
| 名称 | 查询用户信息 |
| 含义 | 查询该系统的所有用户信息 |
| 来源 | 数据库和系统管理员输入的信息 |
| 去向 | 输出框中的用户信息 |
| 组成 | 用户名，权限 |

4）数据存储

|  |  |
| --- | --- |
| 名称 | 用户记录 |
| 含义 | 用户的基本信息 |
| 组成 | 用户ID，用户名，用户密码，用户权限 |
| 关键字 | 用户ID |
| 关联处理 | 书籍借阅，书籍归还，查询用户信息 |

|  |  |
| --- | --- |
| 名称 | 书籍记录 |
| 含义 | 书籍的基本信息 |
| 组成 | 书籍ID，书名，出版社，作者，地址，出版日期，价格，备注，总数量，被借书数量 |
| 关键字 | 书籍ID |
| 关联处理 | 查询书籍信息，书籍出借，书籍归还 |

|  |  |
| --- | --- |
| 名称 | 借阅记录 |
| 含义 | 书籍借阅信息 |
| 组成 | ID，学生姓名，书名，还书日期，借书日期，备注，已还 |
| 关键字 | ID |
| 关联处理 | 查询借阅信息 |

5）处理过程

|  |  |
| --- | --- |
| 名称 | 查询用户信息 |
| 含义 | 查询系统中的用户信息 |
| 输入数据流 | 用户信息 |
| 处理 | 在数据库中查询所有的用户信息，显示到用户信息表格 |
| 输出数据流 | 用户信息表格 |

|  |  |
| --- | --- |
| 名称 | 查询书籍信息 |
| 含义 | 查询系统中的书籍信息 |
| 输入数据流 | 书籍信息 |
| 处理 | 在数据库中查询所有的书籍信息，显示到书籍信息表格 |
| 输出数据流 | 书籍信息表格 |

|  |  |
| --- | --- |
| 名称 | 查询借阅信息 |
| 含义 | 查询系统中的书籍借阅信息 |
| 输入数据流 | 借阅信息，学生名字，书名 |
| 处理 | 在数据库中查询所有的借阅信息，显示到书籍借阅信息表格 |
| 输出数据流 | 书籍借阅信息表格 |

|  |  |
| --- | --- |
| 名称 | 书籍出借 |
| 含义 | 书籍出借 |
| 输入数据流 | ID，学生姓名，书名，还书日期，借书日期，备注，已还 |
| 处理 | 将输入信息整理汇总到书籍借阅记录中 |
| 输出数据流 | 书籍借阅记录 |

|  |  |
| --- | --- |
| 名称 | 书籍归还 |
| 含义 | 书籍归还 |
| 输入数据流 | 还书者姓名，书名，日期，备注 |
| 处理 | 将输入信息整理汇总到书籍借阅记录中 |
| 输出数据流 | 书籍借阅记录 |

# 概念结构设计

## 3.1 概念模型：ER图

实体及属性：

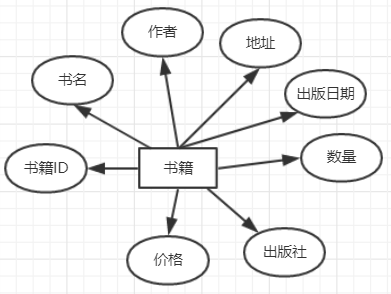
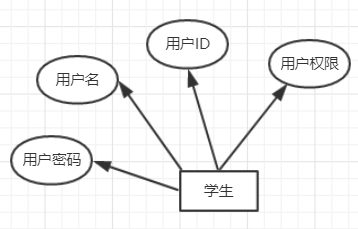
学生（用户ID，用户名，用户密码，权限），

管理者（用户ID，用户名，用户密码，权限），

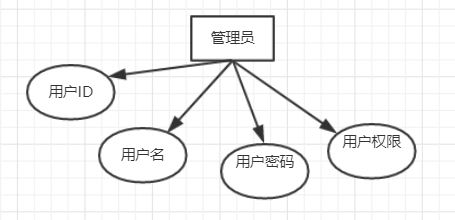
图书（书籍ID，书名，作者，数量，出版社，价格，地址，出版日期）

关系及属性：帐号管理，书籍管理，借阅（借书日期、还书日期）

根据以上信息，编制如下的E-R模型图：

书籍ER图 学生ER图



管理员ER图

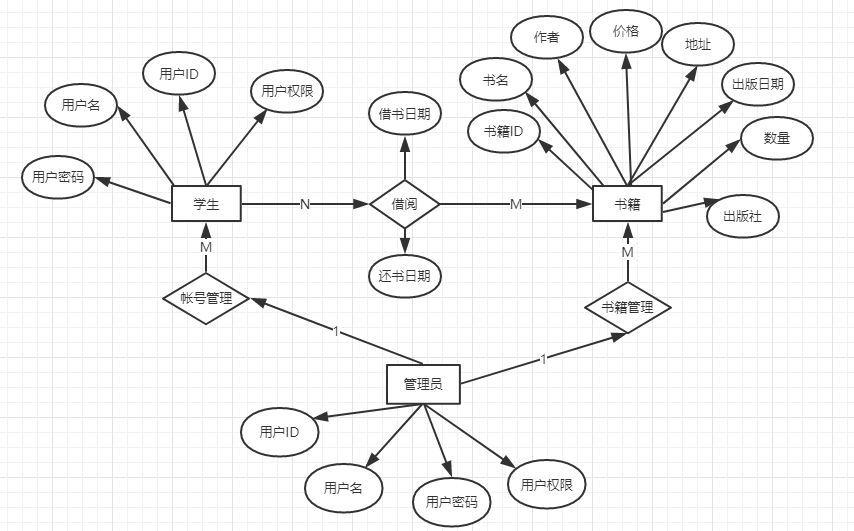


图3.1 总E-R模型图

# 逻辑结构设计

## 4.1 关系模型：所有表结构及范式分析等

采用C/S逻辑结构的设计。管理员直接通过数据库登录密码登录，其他每一个用户登录都要输入用户名和密码以验证身份。然后在后台数据库系统查询用户是否存在，返回给用户一个结果。用户根据自己的权限进行相应的操作。

关系模型为（满足第三范式3NF）：

用户表UserTable(用户ID，用户名，用户密码，用户权限)

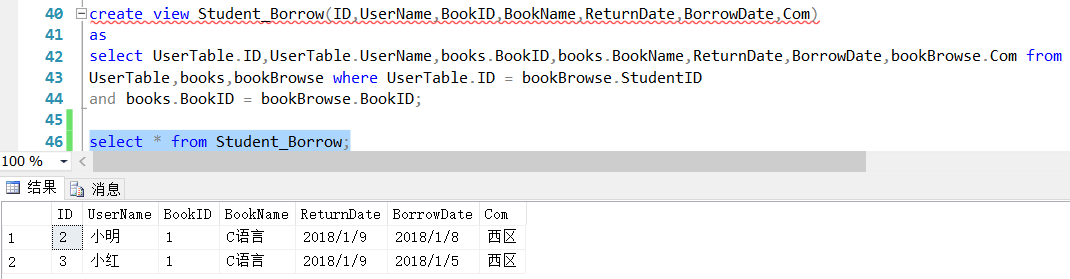
书籍表books（书籍ID，书名，出版社，作者，地址，出版日期，价格，备注，总数量，已借走数量）

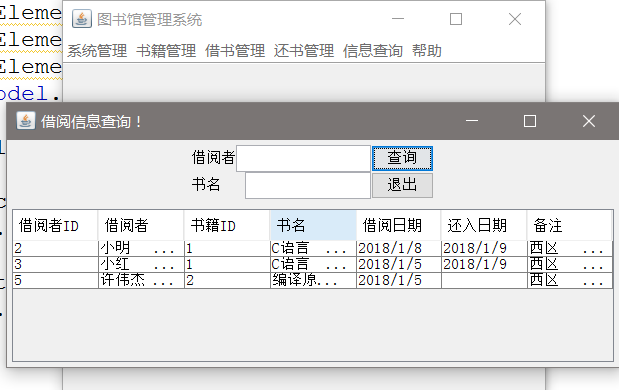
借阅表booksBrowse(ID，书名，学生姓名，出版社，作者，地址，借/还书日期，价格，备注，总数量，已借走数量)

**备注：**a. ID字段均被设置为“标识”，标识种子为1，增量为1。被设置为“标志”的字段的数据必须为数值类型；b. 下划线表示主键。

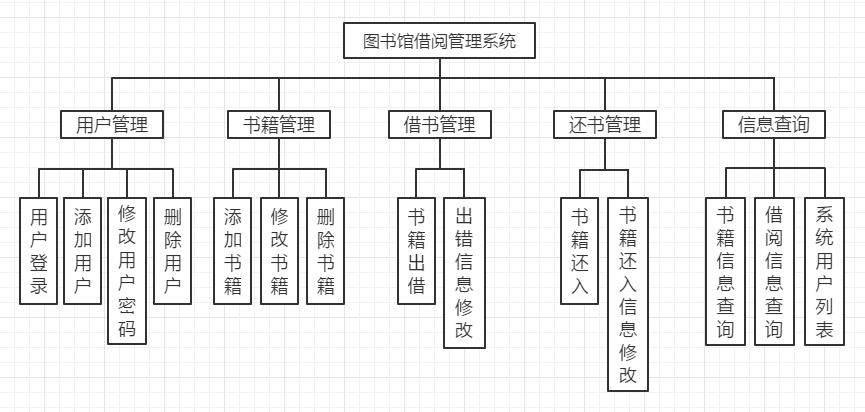
## 4.2 用户子模式

该管理系统只有一个图书借阅子模式，如下所示：

程序中应用:



## 4.3 应用程序功能模块图



**应用程序功能模块图**

## 4.4 安全性（用户类别和权限）设计和实现说明

一、系统管理员

拥有系统的所有权限(即系统管理、书籍管理、借书管理、还书管理、信息查询、帮助)

二、书籍管理员

拥有管理书籍的权限(即书籍管理、信息查询、帮助)

三、学生

拥有借阅书籍的权限(借书管理、还书管理、信息查询、帮助)

用户权限说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 用户  权限  功能 | 系统管理员 | 书籍管理员 | 学生 |
| 用户登录 | √ | √ | √ |
| 添加用户 | √ | × | × |
| 修改用户密码 | √ | × | × |
| 删除用户 | √ | × | × |
| 添加书籍 | √ | √ | × |
| 修改书籍 | √ | √ | × |
| 删除书籍 | √ | √ | × |
| 书籍出借 | √ | × | √ |
| 修改书籍借阅信息 | √ | × | √ |
| 书籍还入 | √ | × | √ |
| 修改书籍还入信息 | √ | × | √ |
| 书籍信息查询 | √ | √ | √ |
| 借阅信息查询 | √ | √ | √ |
| 查看用户列表 | √ | × | × |

**用户权限说明**

## 4.5 完整性（主、外码和用户自定义的完整性约束）设计和说明

1. 实体完整性约束：关系中的主属性不能为空，在数据库中使用Primary Key实现；

2. 参照完整性约束：借阅表中设置用户ID和书籍ID为外码，限制借阅表中出现非法的数据；

3. 用户自定义的完整性约束：规定姓名、书名和借书日期不为空。

# 数据库物理设计

## 5.1 所有索引的定义

1. 在用户编号上建立索引，加快查询速度：

createuniqueindexuser\_numonUserTable(ID);

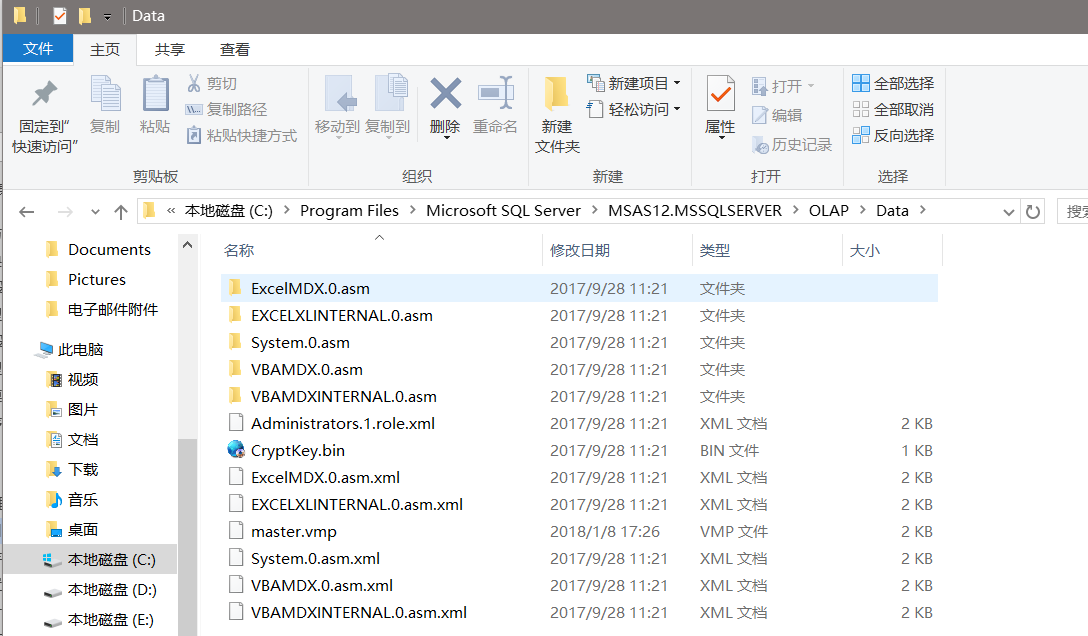
2. 在书籍编号上建立索引，加快查询速度：

createuniqueindexbook\_numonbooks(BookID);

## 5.2 数据的存放位置说明

本地磁盘：

C:\Program Files\Microsoft SQL Server\MSAS12.MSSQLSERVER\OLAP\Data



## 5.3 系统配置说明

系统开发环境：Windows 10

编程开发语言：Java

Java JDK版本：JDK 1.7

编译软件： Eclipse

数据库软件：Sql Server 2014

1. 操作环境：支持Windows XP 及以上的环境进行操作
2. 硬件：所需内存512Mb及以上
3. 响应时间：不超过1S
4. 可修改性：可对部分功能进行修改，如增加、删除某些功能
5. 可用性：正常情况下系统可以使用，出现故障会报错及能程序员能及时找出故障原因并修复
6. 可维护性：系统可维护，不容易瘫痪
7. 易用性：容易操作，操作专业知识不高或没有，有操作教程更好
8. 安全性：需满足不泄露系统使用者的信息等安全需求

## 5.4 模块设计（模块IPO图）



**模块IPO图**

根据系统结构图设计如下各模块具体功能（部分类似的省略）：

|  |
| --- |
| 场景名称：用户登录 |
| 参与执行者实例：系统用户、本软件系统界面、数据库 |
| 前置条件：软件已经打开、数据库已经连接 |
| 事件流：1 用户单击“用户登录”菜单项  2 系统弹出一个子窗口，用户在子窗口输入登录账号和密码  3 系统查询数据库用户列表，判断此账号是否存在，账号和密码是否吻合  4 如果账户存在并且账号和密码吻合，系统将根据该账号锁对应的操作权限设定用户所能操作的菜单选项 |
| 后置条件：用户已经登录，用户可以操作有权使用的菜单项 |
|  |

用户登录

|  |
| --- |
| 场景名称：添加用户 |
| 参与执行者实例：系统用户、本软件系统界面、数据库 |
| 前置条件：软件已经打开、数据库已经连接、用户具有添加新用户的权限 |
| 事件流：1 用户单击“用户添加”菜单项  2 系统弹出一个子窗口，用户在子窗口输入新增用户的登入名、登入密码和权限  3 系统向数据库用户表新增一条记录 |
| 后置条件：已新增用户，新用户可以使用指定登入名和密码登入 |
|  |

添加用户

|  |
| --- |
| 场景名称：修改用户密码 |
| 参与执行者实例：系统用户、本软件系统界面、数据库 |
| 前置条件：软件已经打开、数据库已经连接、该用户具有修改其他用户密码的权限 |
| 事件流：1 用户单击“修改用户密码”菜单项  2 系统弹出一个子窗口，用户在子窗口输入欲修改用户的登入名、原密码和新密码  3 系统向数据库用户表刷新一条记录 |
| 后置条件：用户密码已经修改，可进行后续操作 |
|  |

修改用户密码

|  |
| --- |
| 场景名称：删除用户 |
| 参与执行者实例：系统用户、本软件系统界面、数据库 |
| 前置条件：软件已经打开、数据库已经连接、该用户具有删除其他用户的权限 |
| 事件流：1 用户单击“删除用户”菜单项  2 系统弹出一个子窗口，用户在子窗口输入欲删除用户的登入名、密码  3 系统向数据库用户表删除此记录 |
| 后置条件：指定用户已经被删除 |
|  |

删除用户

|  |
| --- |
| 场景名称：添加书籍 |
| 参与执行者实例：系统用户、本软件系统界面、数据库 |
| 前置条件：软件已经打开、数据库已经连接、该用户具有添加书籍的权限 |
| 事件流：1 用户单击“添加书籍”菜单项  2 系统弹出一个子窗口，用户在子窗口输入欲添加书籍的书籍名称、出版社、作者、出版社地址、出版日期、价格、新增数目、备注  3 系统向数据库书籍表新增一条记录 |
| 后置条件：书籍已经添加，可以进行后续操作 |
|  |

添加书籍

|  |
| --- |
| 场景名称：修改书籍 |
| 参与执行者实例：系统用户、本软件系统界面、数据库 |
| 前置条件：软件已经打开、数据库已经连接、该用户具有修改书籍的权限 |
| 事件流：1 用户单击“修改书籍”菜单项  2 系统弹出一个子窗口，用户在子窗口输入欲修改书籍的书名  3 用户修改书籍的基本信息  4 系统向数据库书籍表刷新书籍记录 |
| 后置条件：书籍信息已经修改 |
|  |

修改书籍

|  |
| --- |
| 场景名称：书籍出借 |
| 参与执行者实例：系统用户、本软件系统界面、数据库 |
| 前置条件：软件已经打开、数据库已经连接、该用户具有出借书籍的权限 |
| 事件流：1 用户单击“书籍出借”菜单项  2 系统弹出一个子窗口，用户在子窗口输入借阅者姓名、借阅书籍名称、借阅日期、备注等信息  3 系统查询数据库书籍表，判断所借书是否还有库存本  4 若有，系统向数据库借阅表新增一条记录  5 系统修改数据库书籍表中该书籍的借出册数 |
| 后置条件：书籍借出，借出册数加1 |
|  |

书籍出借

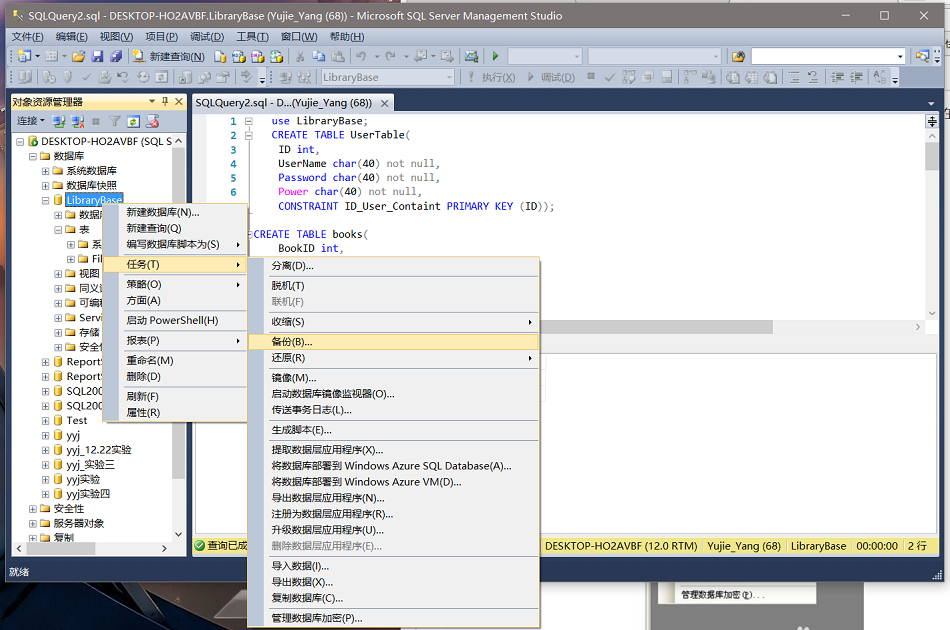
|  |
| --- |
| 场景名称：查看用户列表 |
| 参与执行者实例：系统用户、本软件系统界面、数据库 |
| 前置条件：软件已经打开、数据库已经连接、该用户具有查看用户列表的权限 |
| 事件流：1 用户单击“查看用户列表”菜单项  2 系统查询数据库用户表，以表格的形式显示所有用户登入名和权限 |
| 后置条件：满足条件的记录已经找到，可以进行其他查询或其他操作 |
|  |

查看用户列表

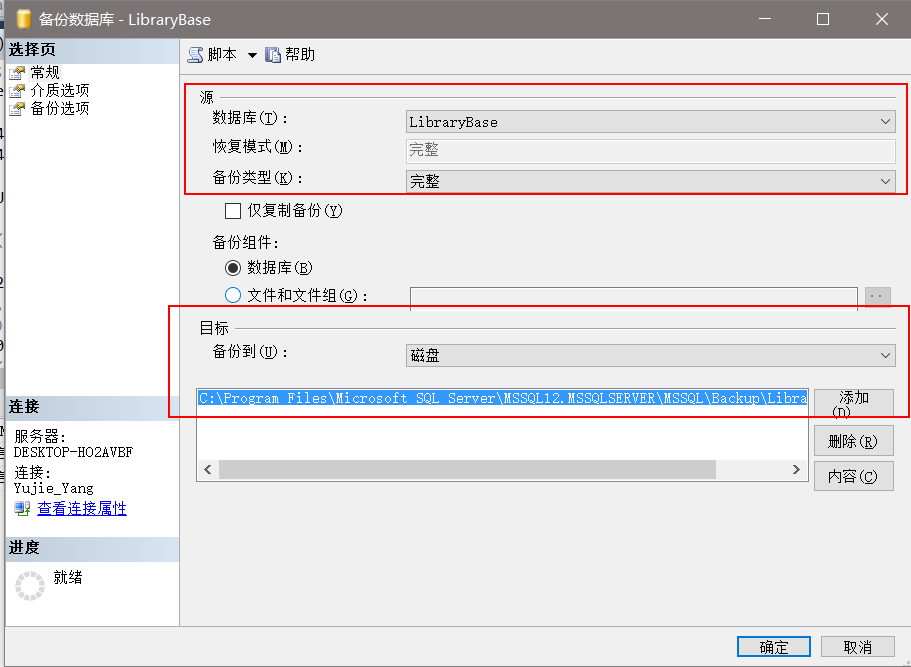
# 数据库实施

## 6.1 数据库备份

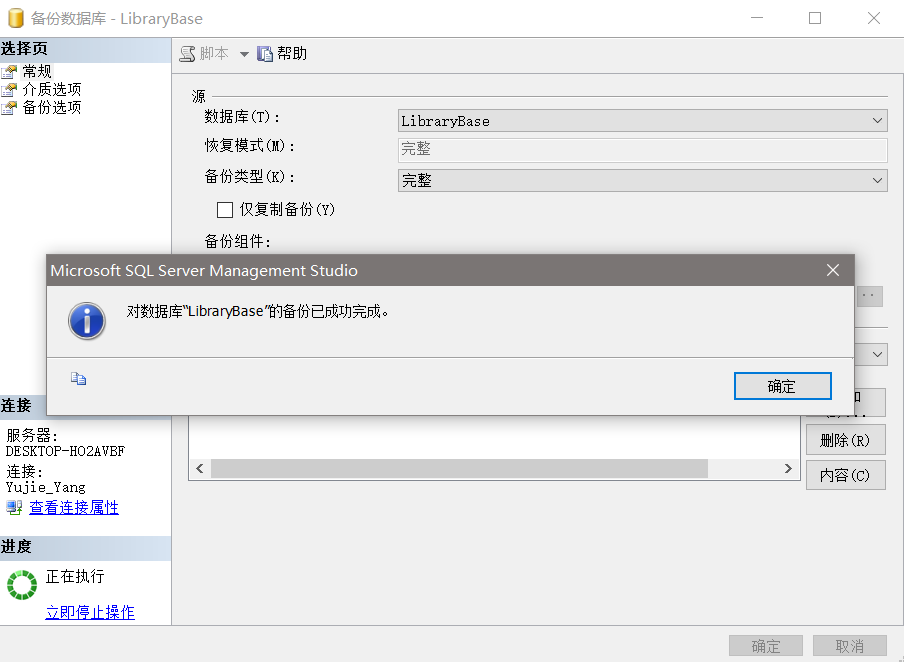
右键数据库->任务->备份：



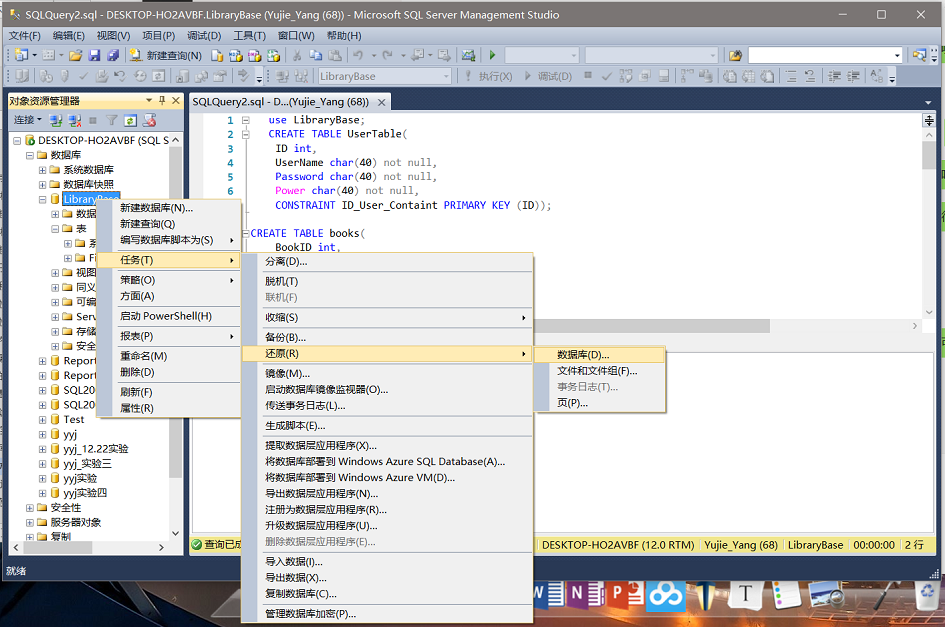
将完整的数据库进行备份，选择备份磁盘路径：



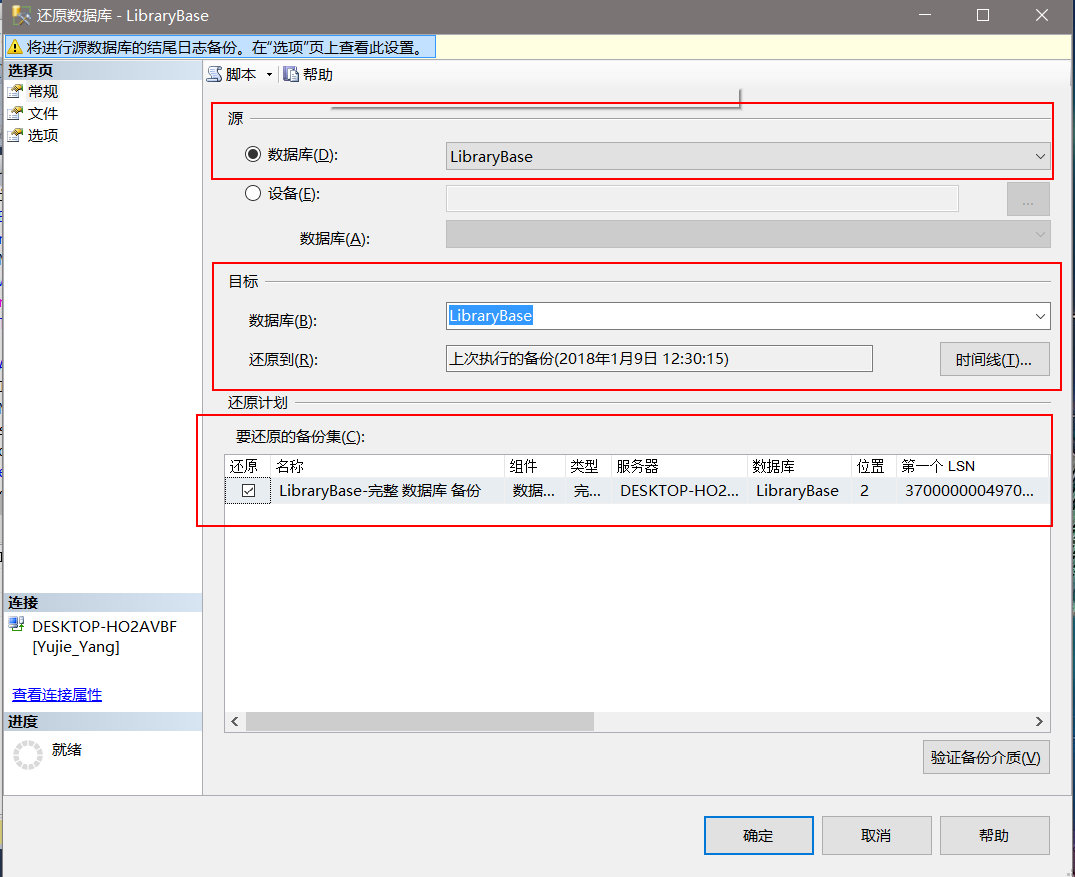
备份完成：



## 6.2 恢复方案



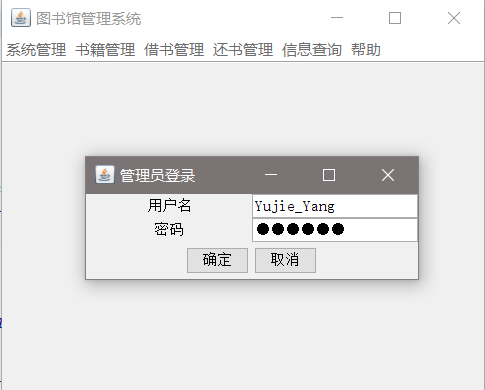
选择源数据库、目标数据库和备份集，进行备份。



# 数据库运行与维护

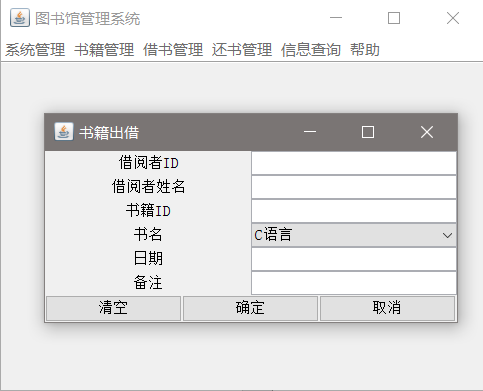
## 7.1 系统使用说明

1. 管理员登陆界面：



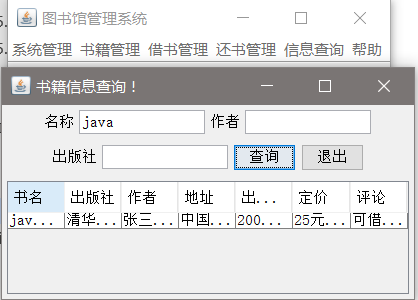
用户进行登录后可以获得相应的操作权限。

2.书籍出借：



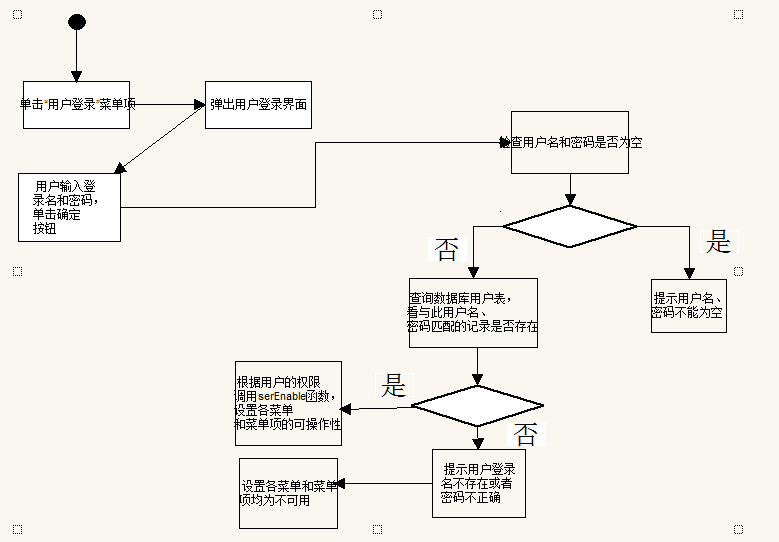
借书管理提供书籍借阅服务，借书者可通过该服务查询借阅相关书籍（还书管理与此类似，故不赘述）。

3.书籍查询：



信息查询功能可查询图书馆现有书籍。

用户登录：



用户登录流程图

**核心代码：**

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == cancelBtn) { //取消按钮

mainFrame.setEnable("else");

**this**.dispose();

}

**else** {

**char**[] password = passwordTextField.getPassword();

String passwordSTR = **new** String(password);

**if** (userTextField.getText().trim().equals("")){//用户名为空

JOptionPane.*showMessageDialog*(**null**, "用户名不可为空!");

**return**;

}

**if** (passwordSTR.equals("")) { //密码为空

JOptionPane.*showMessageDialog*(**null**, "密码不可为空!");

**return**;

}

String strSQL;

strSQL = "select \* from usertable where UserName='" +

userTextField.getText().trim() + "'and Password='" +

passwordSTR + "'";

rs = db.getResult(strSQL);

**boolean** isExist = **false**;

**try** {

isExist = rs.first();

}

**catch** (SQLException sqle) {

System.*out*.println(sqle.toString());

}

**if** (!isExist) {

JOptionPane.*showMessageDialog*(**null**, "用户名不存在或者密码不正确!");

mainFrame.setEnable("else");

}

**else** {

**try** {

rs.first();

mainFrame.setEnable(rs.getString("power").trim());

db.closeConnection();

**this**.dispose();

}

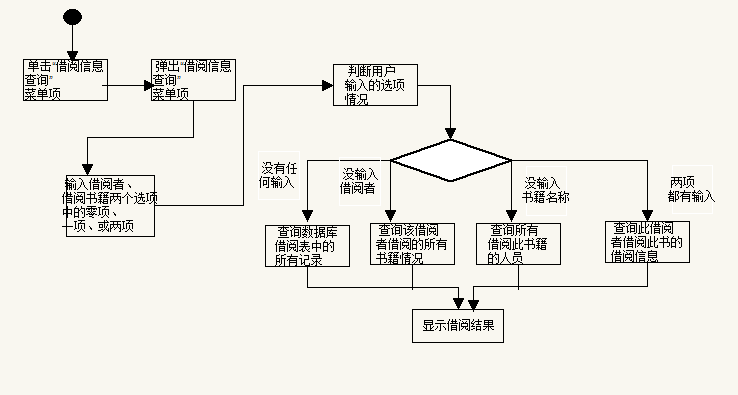
**catch** (SQLException sqle2) {

System.*out*.println(sqle2.toString());

}

}

借阅信息查询：



借阅信息查询流程图

**核心代码：**

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == clearBtn) {

borrowedBookStudentTextField.setText("");

borrowedBookNameTextField.setText("");

borrowedDateTextField.setText("");

borrowedCommentTextField.setText("");

}

**else** **if** (e.getSource() == cancelBtn) {

**this**.dispose();

}

**else** **if** (e.getSource() == yesBtn) {

**try** {

String strSQL =

"select studentName,bookName,borrowDate,com from BookBrowse where studentName='" +

borrowedBookStudentTextField.getText().trim() + "'and bookName='" +

borrowedBookNameTextField.getText().trim() + "'";

rs = db.getResult(strSQL);

**if** (!rs.first()) {

JOptionPane.*showMessageDialog*(**null**, "此学生没有借过书！或者没有此书！");

}

**else** {

borrowedBookStudentTextField.setText(rs.getString(1).trim());

borrowedBookNameTextField.setText(rs.getString(2).trim());

borrowedDateTextField.setText(rs.getString(3).trim());

borrowedCommentTextField.setText(rs.getString(4).trim());

updateBtn.setEnabled(**true**);

}

}

**catch** (Exception ex) {

System.*out*.println(ex.toString());

}

}

**else** **if** (e.getSource() == updateBtn) {

String strSQL = "update bookBrowse set borrowDate='" +

borrowedDateTextField.getText().trim() + "',com='" +

borrowedCommentTextField.getText().trim() + "' where studentName='" +

borrowedBookStudentTextField.getText().trim() + "'and bookName='" +

borrowedBookNameTextField.getText().trim() + "'";

**if** (db.updateSql(strSQL)>0) {

JOptionPane.*showMessageDialog*(**null**, "更新成功！");

db.closeConnection();

**this**.dispose();

}

**else** {

JOptionPane.*showMessageDialog*(**null**, "更新失败！");

db.closeConnection();

**this**.dispose();

}

}

## 7.2 系统测试方案和测试报告

1. 系统测试方案：

这次课设的主要目的是数据库的设计与对数据库基本操作的实现，所以采用黑盒测试，测试系统的功能实现。

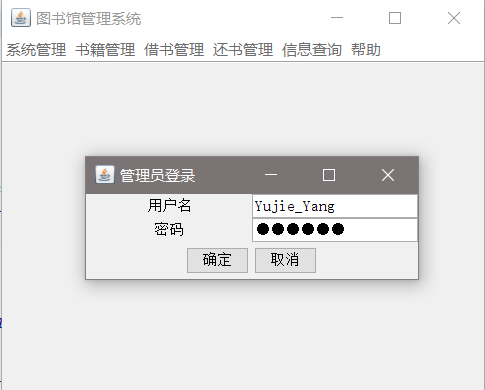
2. 测试报告：

在数据库中有系统管理员、书籍管理员、学生三类用户，方便操作。



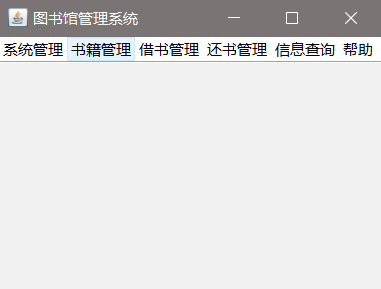
**系统用户**

管理员登录：



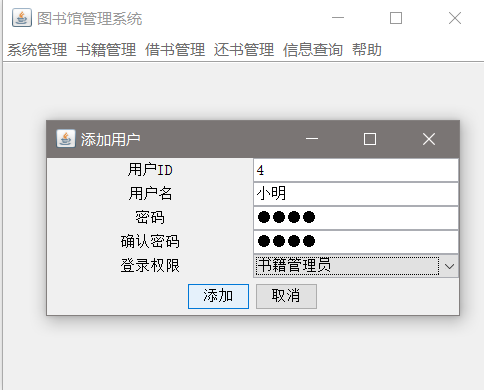
**管理员登录**

激活系统菜单：系统菜单全部激活，系统管理员可以进行所有操作。

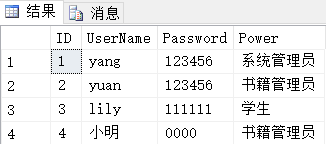


**激活系统菜单**

添加管理员：添加小明为书籍管理员，密码初始化为0000，数据库发生对应变化。

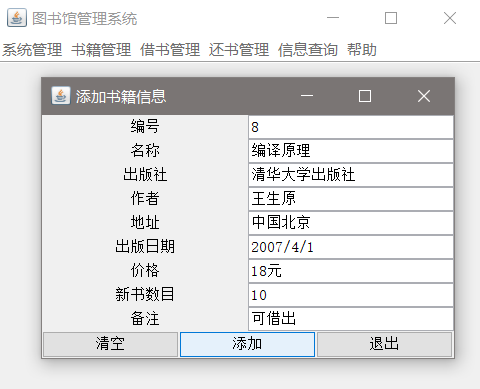


**添加管理员**

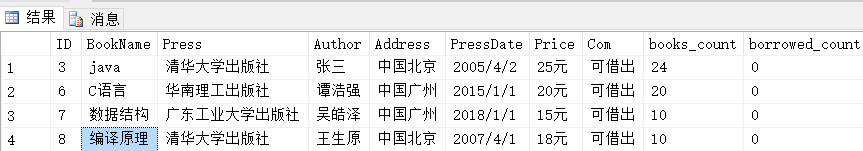


**用户数据库**

添加书籍：添加书籍《编译原理》及相应信息，数据库对应发生变化。

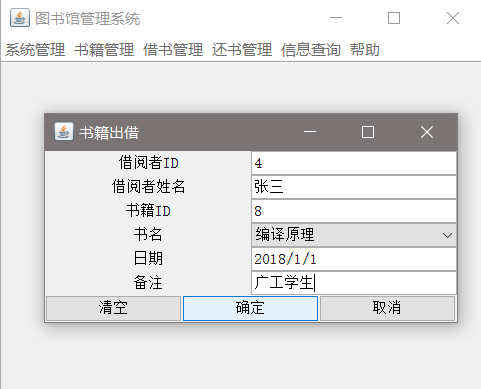


**添加书籍**

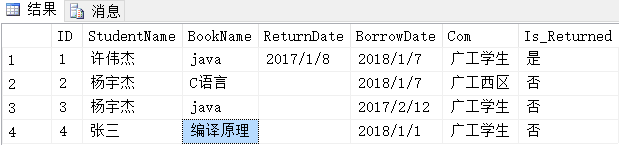


**书籍数据库**

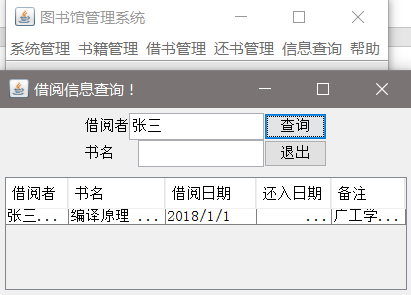
借阅书籍：书籍借出，点击下拉框可以选择要借的书籍。



**书籍借阅**

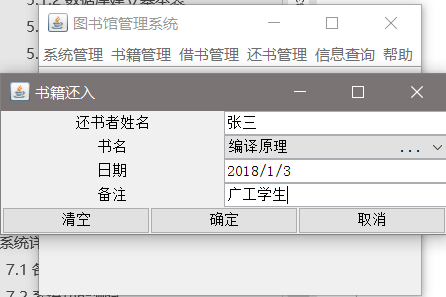


借阅信息查询：选择借阅信息查询，可以看到名叫张三的学生借了一本书。

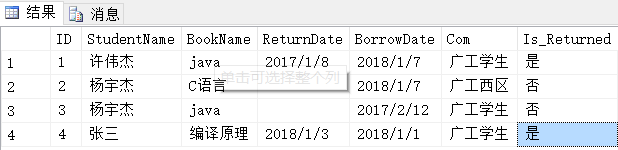


**借阅信息查询**

归还书籍：输入还书者姓名和日期进行书籍归还，数据库发生对应变化。

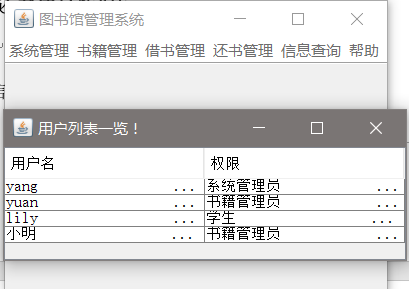


**归还书籍**



**还书信息数据库**

用户列表：管理员可查看用户列表信息。



**用户列****表**

**3. 测试评价**

1）总体评价

本图书借阅管理系统基本实现了一个图书馆中的基本功能。在界面上还是做得一般的，没有凸显出一个图书馆界面，但是总体上的感觉还行。各大功能模块的基本功能足够满足每一个用户的需要，能实现对数据库的增删改查功能，基本上实现了系统的总体功能。但由于自身的数据库水平和Java水平的有限，所以在开发一个图书借阅管理系统时还是显得比较乏力，在开发的过程中也凸显出许多不足，如UI设计不是很好等，以后会针对本次课设的不足和问题进行对应的学习，希望下次能开发出更好的系统出来。

2）缺陷与不足

本系统缺点也是比较多的，如UI做的不过好，精简了很多图书馆的实际管理功能，数据库设计的不够规范，数据库的逻辑结构和物理结构设计没有考虑完整。虽然只是模拟图书馆的借阅管理系统，但是实际还应该考虑更多因素，如发生故障如何处理、用户超期未归还书籍如何处理，等等。在逻辑结构优化方面做的也不够好，没有提供一个合理的逻辑结构，仍需做很多优化，这些留待以后对数据库及软件开发知识的深入学习后再继续改进。

# 系统安装说明

本系统需要在JDK环境下进行使用，建议使用MyEclipse或Eclipse进行安装使用，具体步骤如下：

1. MyEclipse或Eclipse导入工程文件，查看环境是否有错，有的话进行调整；
2. 用SQL Server导入数据库文件；
3. 创建一个数据源，该数据源必须属于某一个数据库用户，然后通过JDK本身自带的JDBC ODBC驱动器加载驱动然后指明路径，连上数据库；
4. 编译运行程序，弹出图书借阅管理系统界面，开始使用。

# 收获和体会

这次数据库的实验可以来说是所有课程设计中最难的了，因为要设计成一个小型数据库来实现某一个现实生活中存在的工作系统，涉及到数据库知识和数据库与某一些高级语言的知识，所以要设计好一个好的小型数据库是不容易的。

总体来说我选择的是我熟悉的Java搭配上实验课所学的SQL Server的知识来构建一个图书馆管理系统。在两个星期的设计过程中可以来说是困难重重，因为完全没用过Java的JDBC功能，所以要不断的查找Java的API文档才能了解它的工作原理，在重复无数次后最终实现了与数据库的链接。在图形界面上也就是GUI上面我用Java的图形界面功能，所以大部分时间主要是链接数据库方面，反而在图形界面上我做的不是很好，只是简单设计了一下图形界面，包括各个菜单选项的实现，比较简洁的风格。

这次在数据库的链接方面花了很大功夫，主要是不明白JDBC如何才能连上SQL Server，通过不断的找资料和上网查资料最后明白了，我们要先创建一个数据源，该数据源必须属于某一个数据库用户，然后通过JDK本身自带的JDBC ODBC驱动器加载驱动然后指明路径，就可以连上数据库了。虽然花了两天功夫才弄明白，不过我觉得还是值得的，毕竟我以后就不会再犯同样的错误了，而且对于Java有了更深的了解了。

最后终于成功了，还是觉得收获很大的。对于以后我们了解一个数据库软件的如何实现有很大的帮助。数据库在我们生活中到处都存在，所以要设计好一个数据库要在逻辑结构设计方面下够功夫才行，否则的话会在后面实现阶段吃很大的苦头!

改进建议：

1. 改进UI界面，使界面体验更良好，交互性更强；
2. 优化数据库，建立多几张表，使数据库功能不会过于集中在三张基本表上，加强各张表之间的联系；
3. 优化程序，加多必要的注释，方便他人阅读时理解程序；
4. 优化测试，解决系统潜在的漏洞。

# 附录

以附录形式给出创建数据库及数据库对象的SQL脚本文件；

useLibraryBase;

CREATETABLEUserTable(

IDint,

UserNamechar(40)notnull,

Passwordchar(40)notnull,

Powerchar(40)notnull,

CONSTRAINTID\_User\_ContaintPRIMARYKEY (ID));

CREATETABLEbooks(

BookIDint,

BookNamechar(20)notnull,

Presschar(20),

Authorchar(20),

Addresschar(20),

PressDatechar(10),

Pricechar(10),

Comchar(20),

books\_countchar(2),

borrowed\_countchar(2),

CONSTRAINTID\_Contraint\_PKPRIMARYKEY (BookID));

CREATETABLEbookBrowse(

IDintidentity,

StudentIDint,

StudentNamechar(40),

BookIDint,

BookNamechar(20),

ReturnDatechar(10),

BorrowDatechar(10),

Comchar(40),

Is\_Returnedchar(2),

CONSTRAINTID\_BookBrowse\_ContaintPRIMARYKEY (ID),

CONSTRAINTFK\_StudentIDforeignkey(StudentID)referencesUserTable(ID),

CONSTRAINTFK\_BookIDforeignkey(BookID)referencesbooks(BookID),

);

createuniqueindexuser\_numonUserTable(ID);

createuniqueindexbook\_numonbooks(BookID);

createviewStudent\_Borrow(ID,UserName,BookID,BookName,ReturnDate,BorrowDate,Com)

as

selectUserTable.ID,UserTable.UserName,books.BookID,books.BookName,ReturnDate,BorrowDate,bookBrowse.Comfrom

UserTable,books,bookBrowsewhereUserTable.ID=bookBrowse.StudentID

andbooks.BookID=bookBrowse.BookID;

select\*fromStudent\_Borrow;

select\*fromUserTable;

select\*frombooks;

select\*frombookBrowse;

INSERTINTOUserTablevalues (1,'yang','123456','系统管理员');

INSERTINTOUserTablevalues (2,'小明','1111','学生');

INSERTINTOUserTablevalues (3,'小红','6666','学生');

INSERTINTOUserTablevalues (4,'杨劲','6666','学生');

INSERTINTOUserTablevalues (5,'许伟杰','6666','学生');

INSERTINTOUserTablevalues (6,'yuan','6666','书籍管理员');

INSERTINTObooksvalues (1,'C语言','华南理工出版社','谭浩强','中国广州','2015/1/1','20元','可借出','20','0');

INSERTINTObooksvalues (2,'编译原理','清华大学出版社','王生原','中国北京','207/1/2','18元','可借出','10','0');

INSERTINTObooksvalues (3,'数据结构','华南理工出版社','张三','中国广州','2011/1/1','40元','可借出','20','0');

INSERTINTObooksvalues (4,'算法导论','电子工业出版社','李四','中国深圳','2012/1/1','30元','可借出','20','0');

INSERTINTObooksvalues (5,'Android','华南理工出版社','王五','中国广州','2014/1/1','40元','可借出','20','0');

INSERTINTObooksvalues (6,'计算机图形学','电子工业出版社','赵六','中国香港','2012/1/1','30元','可借出','20','0');

INSERTINTObookBrowsevalues (2,'小明',1,'C语言','2018/1/9','2018/1/8','西区','是');

INSERTINTObookBrowsevalues (3,'小红',1,'C语言','2018/1/9','2018/1/5','西区','是');

INSERTINTObookBrowsevalues (5,'许伟杰',2,'编译原理',null,'2018/1/5','西区','否');

createloginYujie\_Yangwithpassword='123456',default\_database=LibraryBase;

createuseruser1forloginYujie\_Yangwithdefault\_schema=dbo;

execsp\_addrolemember 'db\_owner','user1';

createloginUserLoginwithpassword='666',default\_database=LibraryBase;

createuseruser2forloginUserLoginwithdefault\_schema=dbo;

GRANTINSERT,UPDATE,DELETE,SELECT

ONbooks

TOuser2

GO

GRANTINSERT,UPDATE,DELETE,SELECT

ONbookBrowse

TOuser2

GO