

《数据库系统》实验报告

实验名称： MIS 系统开发

成 绩：

专业班级： 1403106

姓 名： 张茗帅

学 号： 1140310606

实 验 日 期： 2017 年 5 月 28 日

实验报告日期： 2017 年 5 月 28 日

一、实验目的

通过高级语言和嵌入式 sql 开发 MIS 系统，对 MIS 应用系统进行分析和数据库设计，绘制出 E-R/IDEFIX 图

实验环境：

- ◆ Mysql 或 SqlServer 数据库
- ◆ 自行选择高级语言和开发工具，可以使用 C、Java、PHP 等语言

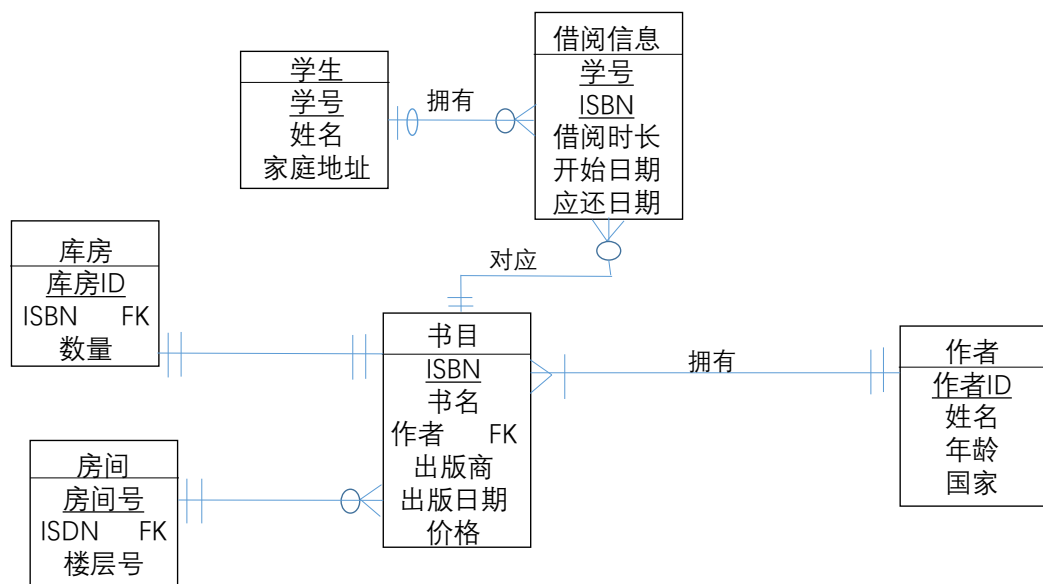
二、实验内容

- 1) MIS 系统的题目和内容自选(如学生学籍管理系统，医疗档案管理系统，图书管理系统等等)。
- 2) 至少包含 5 个以上的库表。
- 3) 有交互式界面，能通过界面插入、修改和删除数据，能够实现一些简单的查询操作。
- 4) 根据 MIS 系统的需求确定用到的实体，属性和联系
- 5) 将实体，属性和联系转化为 E-R 图

三、实验结果

- 给出 MIS 系统的 ER 图

图书管理系统ER图



➤ 列出所设计的数据库表结构。

按照我所设计的 ER 图可知，一共有六个表，如下：

```
mysql> use bookdb;
Database changed
mysql> show tables;
+-----+
| Tables_in_bookdb |
+-----+
| author            |
| book              |
| borrow            |
| house             |
| room              |
| student           |
+-----+
6 rows in set (0.02 sec)
```

注：其中 house 表对应“库房”实体；room 表对应“房间”实体；

author 表对应“作者”实体；book 表对应“书目”实体；

student 表对应“学生”实体；borrow 表对应“借阅信息”实体；

通过 show columns from 表名 和 show create table 表名字来展示该表的结构
author 表的结构如下：

```
mysql> show create table author;
+-----+-----+
| Table | Create Table |
+-----+-----+
| author | CREATE TABLE `author` (
  `AuthorID` smallint(5) unsigned NOT NULL DEFAULT '0',
  `Name` varchar(20) NOT NULL,
  `Age` smallint(5) unsigned DEFAULT NULL,
  `Country` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`AuthorID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.02 sec)

mysql> show columns from author;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| AuthorID | smallint(5) unsigned | NO | PRI | 0 | |
| Name | varchar(20) | NO | | NULL | |
| Age | smallint(5) unsigned | YES | | NULL | |
| Country | varchar(30) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

book 表的结构如下:

```

CREATE TABLE `book` (
  `ISBN` bigint(20) unsigned NOT NULL DEFAULT '0',
  `Title` varchar(30) NOT NULL,
  `AuthorID` smallint(5) unsigned NOT NULL,
  `Publisher` varchar(20) DEFAULT NULL,
  `PublishDate` date DEFAULT NULL,
  `Price` double(5,2) unsigned DEFAULT NULL,
  PRIMARY KEY (`ISBN`),
  KEY `AuthorID` (`AuthorID`),
  CONSTRAINT `book_ibfk_1` FOREIGN KEY (`AuthorID`) REFERENCES `author` (`AuthorID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;

```

```
1 row in set (0.00 sec)
```

```
mysql> show columns from book;
```

Field	Type	Null	Key	Default	Extra
ISBN	bigint(20) unsigned	NO	PRI	0	
Title	varchar(30)	NO		NULL	
AuthorID	smallint(5) unsigned	NO	MUL	NULL	
Publisher	varchar(20)	YES		NULL	
PublishDate	date	YES		NULL	
Price	double(5,2) unsigned	YES		NULL	

```
6 rows in set (0.02 sec)
```

student 表的结构如下:

```
mysql> show create table student;
```

```
+-----+
+-----+
| Table      | Create Table                                     |
+-----+-----+
+-----+-----+
+-----+-----+
| student    | CREATE TABLE `student` (  
  `snumber` varchar(10) NOT NULL,  
  `sname` varchar(20) NOT NULL,  
  `saddr` varchar(40) NOT NULL,  
  PRIMARY KEY (`snumber`)  
> ENGINE=InnoDB DEFAULT CHARSET=utf8 |
```

```
1 row in set (0.00 sec)
```

```
mysql> show columns from student;
```

Field	Type	Null	Key	Default	Extra
snumber	varchar(10)	NO	PRI	NULL	
sname	varchar(20)	NO		NULL	
saddr	varchar(40)	NO		NULL	

```
3 rows in set (0.01 sec)
```

borrow 表的结构如下:

```
mysql> show create table borrow;
+-----+-----+
| Table | Create Table
+-----+-----+
| borrow | CREATE TABLE `borrow` (
  `snumber` varchar(10) NOT NULL,
  `isbn` bigint(20) NOT NULL,
  `day` int(11) DEFAULT NULL,
  `beginday` date DEFAULT NULL,
  `endday` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
+-----+-----+
1 row in set (0.05 sec)

mysql> show columns from borrow;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| snumber | varchar(10) | NO | | NULL | |
| isbn | bigint(20) | NO | | NULL | |
| day | int(11) | YES | | NULL | |
| beginday | date | YES | | NULL | |
| endday | date | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)
```

room 表的结构如下:

```
mysql> show create table room;
+-----+-----+
| Table | Create Table
+-----+-----+
| room | CREATE TABLE `room` (
  `rnumber` int(11) NOT NULL,
  `isbn` bigint(20) unsigned NOT NULL DEFAULT '0',
  `floor` int(11) DEFAULT NULL,
  KEY `isbn` (`isbn`),
  CONSTRAINT `room_ibfk_1` FOREIGN KEY (`isbn`) REFERENCES `book` (`ISBN`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
+-----+-----+
1 row in set (0.00 sec)

mysql> show columns from room;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| rnumber | int(11) | NO | | NULL | |
| isbn | bigint(20) unsigned | NO | MUL | 0 | |
| floor | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

house 表的结构如下：

```
mysql> show create table house;
+-----+
| Table | Create Table
+-----+
| house | CREATE TABLE `house` (
  `hnumber` int(11) NOT NULL,
  `isbn` bigint(20) unsigned NOT NULL DEFAULT '0',
  `count` int(11) DEFAULT NULL,
  KEY `isbn` (`isbn`),
  CONSTRAINT `house_ibfk_1` FOREIGN KEY (`isbn`) REFERENCES `book` (`ISBN`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
+-----+

1 row in set (0.00 sec)

mysql> show columns from house;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| hnumber | int(11)              | NO   |     | NULL    |       |
| isbn    | bigint(20) unsigned | NO   | MUL | 0        |       |
| count   | int(11)              | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+

3 rows in set (0.02 sec)
```

➤ 简述程序实现的具体过程

首先创建我所需要的六个表，然后开始设计 UI 界面，我的 GUI 界面通过 matisse 插件基于 swing 下的 JFrame 框架进行制作的，只需在面板上进行控件拖动，界面做好后保存自动生成代码，最后设计的 GUI 界面如下图：

Library Management System designed by ZhangMingshuai

☐ 图书: ISBN 书名 作者ID 出版社
出版日期 价格 数量 库房号
楼层 房间号

☐ 作者: 作者ID 姓名 年龄 国家

☐ 学生: 学号 姓名 地址

插入 更新 删除 返回信息:

学号 ISBN 借阅时长

开始日期 到期日期 借阅 查询全部图书

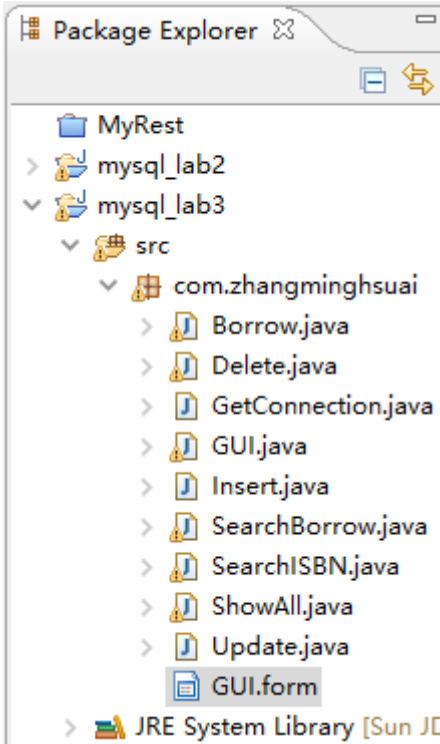
ISBN 查询 学号 查询

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存

< >

学号	姓名	地址	图书名	作者	借阅开始日期	借阅终止日期

然后就是 Java 程序代码的设计了，其实与实验二很像，只是实验二只做了查询，本次实验增删改查都要做，同样为每一个表的增删改查预先写好 SQL 语句，未知参数用?代替，等到获取了 UI 界面输入的参数后，再绑定给 SQL 语句并执行，获得的结果返回信息再传回给 UI 界面即可。总体视图如下，各个类的详细源代码见第四部分。



➤ 截屏给出程序界面，操作界面和操作结果

点击查询全部图书

Library Management System designed by ZhangMingshuai

☐ 图书: ISBN 书名 作者ID 出版社
出版日期 价格 数量 库房号
楼层 房间号

☐ 作者: 作者ID 姓名 年龄 国家

☐ 学生: 学号 姓名 地址

返回信息:

学号 ISBN 借阅时长

开始日期 到期日期

ISBN 学号

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	
1000000001	实现美国的...	奥巴马	40	美国	美国教育出...	2002-08-30	85.99	1	7	103
1000000002	何时回到祖...	蔡英文	35	中国	台湾民进党...	2014-11-12	50.33	1	8	102
1000000003	竞选之路	蔡英文	35	中国	台湾民进党...	2016-09-12	33.33	1	7	202
1000000004	当总理的那...	普京	52	俄罗斯	俄罗斯铁人...	2007-03-21	120.99	2	10	202
1000000005	论与美国的...	普京	52	俄罗斯	俄罗斯战争...	2010-05-01	99.99	2	3	212
1000000006	我的失败人生	马英九	40	中国	台湾自由出...	2010-03-28	39.99	2	7	212

学号	姓名	地址	图书名	作者	借阅开始日期	借阅终止日期
----	----	----	-----	----	--------	--------

输入我的学号进行查询我所借阅的书籍

Library Management System designed by ZhangMingshuai

☐ 图书: ISBN 书名 作者ID 出版社
出版日期 价格 数量 库房号
楼层 房间号

☐ 作者: 作者ID 姓名 年龄 国家

☐ 学生: 学号 姓名 地址

插入 更新 删除 返回信息: Search Successfully! -Remainin

学号 ISBN 借阅时长
开始日期 到期日期 借阅 查询全部图书

ISBN 查询 学号 1140310606 查询

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	房
1140310606	张帅	吉林松原	热力学传输理论	周玉	2008-08-09	2008-08-29				
1140310606	张帅	吉林松原	竞选之路	蔡英文	1999-06-06	1999-06-11				
1140310606	张帅	吉林松原	实现美国的伟大复兴	奥巴马	2001-06-07	2001-06-08				

输入一本书的 ISBN 号码进行该书详细信息的查询

Library Management System designed by ZhangMingshuai

☐ 图书: ISBN 书名 作者ID 出版社
出版日期 价格 数量 库房号
楼层 房间号

☐ 作者: 作者ID 姓名 年龄 国家

☐ 学生: 学号 姓名 地址

插入 更新 删除 返回信息: Search Successfully! -Remainin

学号 ISBN 借阅时长
开始日期 到期日期 借阅 查询全部图书

ISBN 1000000007 查询 学号 查询

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	房
1000000007	我与工大的...	周玉	64	中国	哈工大出版社	2011-08-01	45.66	2	18	315

插入操作（这里只以插入作者进行示范，图书和学生的插入类似）

Library Management System designed by ZhangMingshuai

☐ 图书: ISBN 书名 作者ID 出版社
出版日期 价格 数量 库房号
楼层 房间号

☒ 作者: 作者ID 姓名 年龄 国家

☐ 学生: 学号 姓名 地址

学习使我快乐
Study always makes me happy !!!
-Remainin

返回信息:

学号 ISBN 借阅时长
开始日期 到期日期

ISBN 学号

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	房间
------	----	-----	------	------	-----	------	----	-----	----	----

学号	姓名	地址	图书名	作者	借阅开始日期	借阅终止日期
----	----	----	-----	----	--------	--------

去数据库进行查询，发现多了 authorID 为 20 的这个条目，于是成功插入

```
mysql> select * from author;
```

AuthorID	Name	Age	Country
1	奥巴马	40	美国
2	蔡英文	35	中国
3	普京	52	俄罗斯
4	马英九	40	中国
5	周玉	64	中国
6	帕克	37	法国
7	张茗帅	20	中国
8	阿德	35	奥地利
10	坏人	19	美国
11	飞机	16	意大利
15	老邓	16	日本
16	宋梅	36	中国
20	小明	20	日本

13 rows in set (0.00 sec)

更新操作（这里也只以作者进行示范，其他类似，修改更新刚才插入的 authorID 为 20 这个作者的年龄为 20->27）

Library Management System designed by ZhangMingshuai

☐ 图书: ISBN 书名 作者ID 出版社

出版日期

价格

数量

库房号

楼层

房间号

☒ 作者: 作者ID 20 姓名 小明 年龄 27 国家 日本

☐ 学生: 学号 姓名 地址

插入

更新

删除

返回信息: Success Update!

学号 ISBN 借阅时长

开始日期

到期日期

借阅

ISBN 查询

学号 查询

查询全部图书

学习使我快乐

Study always makes me happy !!!

-Remainin

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	房间
------	----	-----	------	------	-----	------	----	-----	----	----

学号	姓名	地址	图书名	作者	借阅开始日期	借阅终止日期
----	----	----	-----	----	--------	--------

去数据库进行查询，发现 authorID 为 20 的这个作者的年龄改为了 27，于是成功更新

```
mysql> select * from author;
```

AuthorID	Name	Age	Country
1	奥巴马	40	美国
2	蔡英文	35	中国
3	普京	52	俄罗斯
4	马英九	40	中国
5	周玉	64	中国
6	帕克	37	法国
7	张茗帅	20	中国
8	阿德	35	奥地利
10	坏人	19	美国
11	飞机	16	意大利
15	老邓	16	日本
16	宋梅	36	中国
20	小明	27	日本

13 rows in set (0.00 sec)

删除操作（这里也只以作者进行示范，其他类似，删除 authorID 为 20 这个作者）

Library Management System designed by ZhangMingshuai

☐ 图书: ISBN 书名 作者ID 出版社
出版日期 价格 数量 库房号
楼层 房间号

☒ 作者: 作者ID 姓名 年龄 国家

☐ 学生: 学号 姓名 地址

插入 更新 删除 返回信息: delete Successfully!

学号 ISBN 借阅时长
开始日期 到期日期 借阅 查询全部图书

ISBN 查询 学号 查询

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	房号
------	----	-----	------	------	-----	------	----	-----	----	----

学号	姓名	地址	图书名	作者	借阅开始日期	借阅终止日期
----	----	----	-----	----	--------	--------

去数据库进行查询，发现 authorID 为 20 的这个作者已经不见了，于是成功删除

```
mysql> select * from author;
```

AuthorID	Name	Age	Country
1	奥巴马	40	美国
2	蔡英文	35	中国
3	普京	52	俄罗斯
4	马英九	40	中国
5	周玉王	64	中国
6	帕克	37	法国
7	张茗帅	20	中国
8	阿德	35	奥地利
10	坏人	19	美国
11	飞机	16	意大利
15	老邓	16	日本
16	宋梅	36	中国

12 rows in set (0.00 sec)

含外键关联则禁止删除（删除作者 ID 为 7 的作者，由于 book 表中有一项书对应的作业为 7 号作者，于是删除失败）

```
mysql> select * from book;
```

ISBN	Title	AuthorID	Publisher	PublishDate	Price
1000000001	实现美国的伟大复兴	1	美国教育出版社	2002-08-30	85.99
1000000002	何时回到祖国的怀抱	2	台湾民进党出版社	2014-11-12	50.33
1000000003	竞选之路	2	台湾民进党出版社	2016-09-12	33.33
1000000004	当总理的那些年	3	俄罗斯铁人出版社	2007-03-21	120.99
1000000005	论与美国的持久战	3	俄罗斯战争出版社	2010-05-01	99.99
1000000006	我的失败人生	4	台湾自由出版社	2010-03-28	39.99
1000000007	我与工大的那一份情	5	哈工大出版社	2011-08-01	45.66
1000000008	热力学传输理论	5	高等教育出版社	2006-01-22	120.99
1000000009	我的篮球故事	6	法国工业出版社	2011-04-01	45.79
1000000010	GDP传奇	6	NBA出版社	2016-04-30	77.79
1000000011	我的童年	1	美国农业出版社	2004-02-10	50.45
1000000012	小时候的事情	8	哈哈出版社	2004-09-04	33.33
1000000013	我的假期如此之短	7	痛苦出版社	2016-07-16	66.66

13 rows in set (0.00 sec)

Library Management System designed by ZhangMingshuai

☐ 图书: ISBN 书名 作者ID 出版社

出版日期 价格 数量 库房号

楼层 房间号

☒ 作者: 作者ID 姓名 年龄 国家

☐ 学生: 学号 姓名 地址

返回信息:

学号 ISBN 借阅时长

开始日期 到期日期

ISBN 学号

ISBN 书名 作者名 作者年龄 作者国家 出版商 出版日期 价格 库房号 库存 房间号

学号 姓名 地址 图书名 作者 借阅开始日期 借阅终止日期

借阅功能 首先查询一本书的库存，不妨查询 isbn 为 1000000005 这本书：

ISBN	1000000005	查询	学号		查询					
ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	房间号
1000000005	论与美国的... 普京	52	俄罗斯	俄罗斯战争...	2010-05-01	99.99	2	3	212	

可以看到，这本书的库存为 3

填写借阅信息并点击“借阅”按钮，如下图：

Library Management System designed by ZhangMingshuai

图书

ISBN

书名

作者ID

出版社

出版日期

价格

数量

库房号

楼层

房间号

作者

作者ID

7

姓名

年龄

国家

学生

学号

姓名

地址

插入

更新

删除

返回信息

borrow successfully!

学号

1140310606

ISBN

1000000005

借阅时长

10

开始日期

2017-05-05

到期日期

2017-05-15

借阅

查询全部图书

ISBN

1000000005

查询

学号

查询

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	房间
1000000005	论与美国的...	普京	52	俄罗斯	俄罗斯战争...	2010-05-01	99.99	2	3	212

学号

姓名

地址

图书名

作者

借阅开始日期

借阅终止日期

学习使我快乐

Study always makes me happy !!!

-Remainin

借阅成功后查询 1140310606 学号所借阅的信息

学号	姓名	地址	图书名	作者	借阅开始日期	借阅终止日期
1140310606	张帅	吉林松原	论与美国的持久战	普京	2017-05-05	2017-05-15

然后再次查询 isbn 为 1000000005 这本书的相关信息，发现库存为 2，也就是少了一本书，说明借阅功能正确

ISBN

1000000005

查询

学号

1140310606

查询

ISBN	书名	作者名	作者年龄	作者国家	出版商	出版日期	价格	库房号	库存	房间
1000000005	论与美国的...	普京	52	俄罗斯	俄罗斯战争...	2010-05-01	99.99	2	2	212

学号

姓名

地址

图书名

作者

借阅开始日期

借阅终止日期

1140310606	张帅	吉林松原	论与美国的持久战	普京	2017-05-05	2017-05-15
------------	----	------	----------	----	------------	------------

四、程序代码

```
package com.zhangminghsuai;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.table.DefaultTableModel;

/**
 * CreateDate: 2017-6-3 下午 11:12:18
 * Location: HIT
 * Author: Zhang Mingshuai
 * TODO
 * return
 */
public class Borrow {

    private String snumber;
    private String isbn;
    private String day;
    private String begindate;
    private String enddate;

    public Borrow(String snumber, String isbn, String day, String begindate, String
enddate) {
        this.snumber = snumber;
        this.isbn = isbn;
        this.day = day;
        this.begindate = begindate;
        this.enddate = enddate;
    }

    public String goBorrow() {
        Connection con = GetConnection.get();
        try {
            PreparedStatement stmt = con.prepareStatement("insert into borrow"+"
values(?,?,?, ?, ?)");
            stmt.setString(1, snumber);
            stmt.setString(2, isbn);
            stmt.setString(3, day);
            stmt.setString(4, begindate);
            stmt.setString(5, enddate);
            stmt.executeUpdate();
            PreparedStatement stmt2 = con.prepareStatement("select count from
house where isbn = ?");
            stmt2.setString(1, isbn);
```

```

        ResultSet rs = stmt2.executeQuery();
        String countString = new String();
        while(rs.next()){
            countString = rs.getString(1);
            countString = String.valueOf(Integer.valueOf(countString)-1);
            PreparedStatement stmt3 = con.prepareStatement("update house set
count=? where isbn=?");
            stmt3.setString(1, countString);
            stmt3.setString(2, isbn);
            stmt3.executeUpdate();
        }
        return "borrow successfully!";
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "borrow failed!";
    }
}
}
}

```

```

.....

package com.zhangminghsuai;

```

```

import java.sql.Connection;
import java.sql.SQLException;

```

```

import com.mysql.jdbc.PreparedStatement;

```

```

/**
 * CreateDate: 2017-6-3 下午 10:53:38
 * Location: HIT
 * Author: Zhang Mingshuai
 * TODO
 * return
 */

```

```

public class Delete {

```

```

    private String isbn;
    private String authorid;
    private String snumber;

```

```

    public String goBook(String isbn1){
        this.isbn = isbn1;
        Connection con = GetConnection.get();
        java.sql.PreparedStatement stmt;
        try {
            stmt = con.prepareStatement("delete from book where isbn = ?");
            stmt.setString(1, isbn);

```

```

        stmt.executeUpdate();
        return "delete Successfully!";
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "delete failed!";
    }
}

public String goAuthor(String author){
    this.authorid = author;
    Connection con = GetConnection.get();
    java.sql.PreparedStatement stmt;
    try {
        stmt = con.prepareStatement("delete from author where authorid = ?");
        stmt.setString(1, authorid);
        stmt.executeUpdate();
        return "delete Successfully!";
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "delete failed!";
    }
}

public String goStudent(String number){
    this.snumber = number;
    Connection con = GetConnection.get();
    java.sql.PreparedStatement stmt;
    try {
        stmt = con.prepareStatement("delete from student where snumber = ?");
        stmt.setString(1, snumber);
        stmt.executeUpdate();
        return "delete Successfully!";
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "delete failed!";
    }
}
}

```

```

package com.zhangminghsuai;

import java.sql.Connection;
import java.sql.DriverManager;

```

```

/**
 * CreateDate: 2017-6-3 下午 08:13:52

```



```

    * Location: HIT
    * Author: Zhang Mingshuai
    * TODO
    * return
    */
public class GetConnection {
    static Connection connect;
    public static Connection get() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("Success loading Mysql Driver!");
        } catch (Exception e) {
            System.out.print("Error loading Mysql Driver!");
            e.printStackTrace();
        }
        try {
            connect =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bookdb", "root",
"1234");
            System.out.println("Success connect Mysql server!");
        } catch (Exception e) {
            System.out.print("get data error!");
            e.printStackTrace();
        }
        return connect;
    }
}

```

.....

```

package com.zhangmingshuai;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

/**
 * CreateDate: 2017-6-3 下午 08:04:24
 * Location: HIT
 * Author: Zhang Mingshuai
 * TODO
 * return
 */
public class Insert {

    private String isbn;
    private String title;
    private String author;
    private String pr;
    private String pd;
    private String price;

```

```

private String house;
private String count;
private String room;
private String floor;

private String snumber;
private String sname;
private String saddr;

private String Aid;
private String Aname;
private String Aage;
private String Acountry;

public Insert(String isbn,String title,String author,
              String pr,String pd,String price,String house,
              String count,String room,String floor){
    this.isbn = isbn;
    this.title = title;
    this.author = author;
    this.pr = pr;
    this.pd = pd;
    this.price = price;
    this.house = house;
    this.count = count;
    this.room = room;
    this.floor = floor;
}

public Insert(String snumber,String sname,String saddr){
    this.snumber = snumber;
    this.sname = sname;
    this.saddr = saddr;
}

public Insert(String Aid,String Aname,String Aage,String Acountry){
    this.Aid = Aid;
    this.Aname = Aname;
    this.Aage = Aage;
    this.Acountry = Acountry;
}

public String goBook(){
    Connection con = GetConnection.get();
    try {
        PreparedStatement stmt = con.prepareStatement("insert into book"+"
values(?, ?, ?, ?, ?, ?)");
        stmt.setString(1, isbn);
        stmt.setString(2, title);
        stmt.setString(3, author);
        stmt.setString(4, pr);

```

```

        stmt.setString(5, pd);
        stmt.setString(6, price);
        PreparedStatement stmt1 = con.prepareStatement("insert into house"+"
values(?, ?, ?)");
        stmt1.setString(1, house);
        stmt1.setString(2, isbn);
        stmt1.setString(3, count);
        PreparedStatement stmt2 = con.prepareStatement("insert into room"+"
values(?, ?, ?)");
        stmt2.setString(1, room);
        stmt2.setString(2, isbn);
        stmt2.setString(3, floor);
        stmt.executeUpdate();
        stmt1.executeUpdate();
        stmt2.executeUpdate();
        return "Success Insert!";
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "Fail";
    }
}

```

```

public String goStudent() {
    Connection con = GetConnection.get();
    try {
        PreparedStatement stmt = con.prepareStatement("insert into student"+"
values(?, ?, ?)");
        stmt.setString(1, snumber);
        stmt.setString(2, sname);
        stmt.setString(3, saddr);
        stmt.executeUpdate();
        return "Success Insert!";

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "Fail";
    }
}

```

```

public String goAuthor() {
    Connection con = GetConnection.get();
    try {
        PreparedStatement stmt = con.prepareStatement("insert into author"+"
values(?, ?, ?, ?)");
        stmt.setString(1, Aid);
        stmt.setString(2, Aname);
        stmt.setString(3, Aage);
        stmt.setString(4, Acountry);
        stmt.executeUpdate();
    }
}

```

```

        return "Success Insert!";

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "Fail";
    }
}
}

```

.....

```

package com.zhangminghsuai;

import java.awt.print.Book;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.table.DefaultTableModel;

import com.sun.org.apache.bcel.internal.generic.Select;
import com.sun.org.apache.xerces.internal.impl.dv.xs.DayDV;

/**
 * CreateDate: 2017-6-3 下午 11:22:37
 * Location: HIT
 * Author: Zhang Mingshuai
 * TODO
 * return
 */
public class SearchBorrow {
    private String snumber;
    private String sname;
    private String saddr;
    private String isbn;
    private String title;
    private String author;
    private String authorid;

    private String begindate;
    private String enddate;

    public SearchBorrow(String snumber) {
        this.snumber=snumber;
    }
    public String goSearch(DefaultTableModel tableModel) {
        Connection con = GetConnection.get();
        try {

```

```

        PreparedStatement stmt1 = con.prepareStatement("select * from borrow
where snumber = ?");
        stmt1.setString(1, snumber);
        ResultSet rs1 = stmt1.executeQuery();
        while(rs1.next()) {
            isbn = rs1.getString(2);
            begindate = rs1.getString(4);
            enddate = rs1.getString(5);
            PreparedStatement stmt2 = con.prepareStatement("select * from
student where snumber = ?");
            stmt2.setString(1, snumber);
            ResultSet rs2 = stmt2.executeQuery();
            while(rs2.next()) {
                sname = rs2.getString(2);
                saddr = rs2.getString(3);
            }
            PreparedStatement stmt3 = con.prepareStatement("select
title,authorid from book where isbn = ?");
            stmt3.setString(1, isbn);
            ResultSet rs3 = stmt3.executeQuery();
            while(rs3.next()) {
                title = rs3.getString(1);
                authorid = rs3.getString(2);
                PreparedStatement stmt4 = con.prepareStatement("select name
from author where authorid = ?");
                stmt4.setString(1, authorid);
                ResultSet rs4 = stmt4.executeQuery();
                while(rs4.next()) {
                    author = rs4.getString(1);
                }
            }
            tableModel.addRow(new
String[] {snumber, sname, saddr, title, author, begindate, enddate});
        }
        return "Search Successfully!";
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "Search failed!";
    }
}
}
}

```

.....

```

package com.zhangminghsuai;

```

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```

import java.sql.SQLException;

import javax.swing.table.DefaultTableModel;

/**
 * CreateDate: 2017-6-4 上午 12:19:51
 * Location: HIT
 * Author: Zhang Mingshuai
 * TODO
 * return
 */
public class SearchISBN {

    private String isbn;
    private String title;
    private String author;
    private String pr;
    private String pd;
    private String price;
    private String house;
    private String count;
    private String room;
    private String floor;

    private String Aid;
    private String Aname;
    private String Age;
    private String Acountry;
    /**
     *
     */
    public SearchISBN(String isbn) {
        // TODO Auto-generated constructor stub
        this.isbn = isbn;
    }

    public String goSearch(DefaultTableModel table) {
        Connection con = GetConnection.get();
        try {
            PreparedStatement stmt = con.prepareStatement("select * from book
where isbn = ?");
            stmt.setString(1, isbn);
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
                title = rs.getString(2);
                author = Aid = rs.getString(3);
                pr =rs.getString(4);
                pd = rs.getString(5);
                price = rs.getString(6);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```

        PreparedStatement Stmt2 = con.prepareStatement("select * from
author where authorid = ?");
        Stmt2.setString(1, Aid);
        ResultSet rs2 = Stmt2.executeQuery();
        while(rs2.next()) {
            Aname = rs2.getString(2);
            Aage = rs2.getString(3);
            Acountry = rs2.getString(4);
        }

        PreparedStatement Stmt3 = con.prepareStatement("select * from
house where isbn = ?");
        Stmt3.setString(1, isbn);
        ResultSet rs3 = Stmt3.executeQuery();
        while(rs3.next()) {
            house = rs3.getString(1);
            count = rs3.getString(3);
        }

        PreparedStatement Stmt4 = con.prepareStatement("select * from room
where isbn = ?");
        Stmt4.setString(1, isbn);
        ResultSet rs4 = Stmt4.executeQuery();
        while(rs4.next()) {
            room = rs4.getString(1);
            floor = rs4.getString(3);
        }
        table.addRow(new
String[] {isbn, title, Aname, Aage, Acountry, pr, pd, price, house, count, room, floor});
    }
    return "Search Successfully";
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    return "Search failed!";
}

}
}

```

.....

```

package com.zhangminghsuai;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.table.DefaultTableModel;

```

```

/**
 * CreateDate: 2017-6-3 下午 08:31:28
 * Location: HIT
 * Author: Zhang Mingshuai
 * TODO
 * return
 */
public class ShowAll {

    private String isbn;
    private String title;
    private String author;
    private String pr;
    private String pd;
    private String price;
    private String house;
    private String count;
    private String room;
    private String floor;

    private String Aid;
    private String Aname;
    private String Aage;
    private String Acountry;

    public void goShow(DefaultTableModel table) {
        Connection con = GetConnection.get();
        try {
            PreparedStatement stmt = con.prepareStatement("select * from book
order by isbn");
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
                isbn = rs.getString(1);
                title = rs.getString(2);
                author = Aid = rs.getString(3);
                pr =rs.getString(4);
                pd = rs.getString(5);
                price = rs.getString(6);

                PreparedStatement Stmt2 = con.prepareStatement("select * from
author where authorid = ?");
                Stmt2.setString(1, Aid);
                ResultSet rs2 = Stmt2.executeQuery();
                while(rs2.next()) {
                    Aname = rs2.getString(2);
                    Aage = rs2.getString(3);
                    Acountry = rs2.getString(4);
                }
            }
        }
    }
}

```



```

        PreparedStatement Stmt3 = con.prepareStatement("select * from
house where isbn = ?");
        Stmt3.setString(1, isbn);
        ResultSet rs3 = Stmt3.executeQuery();
        while(rs3.next()) {
            house = rs3.getString(1);
            count = rs3.getString(3);
        }

        PreparedStatement Stmt4 = con.prepareStatement("select * from room
where isbn = ?");
        Stmt4.setString(1, isbn);
        ResultSet rs4 = Stmt4.executeQuery();
        while(rs4.next()) {
            room = rs4.getString(1);
            floor = rs4.getString(3);
        }
        table.addRow(new
String[] {isbn, title, Aname, Age, Acountry, pr, pd, price, house, count, room, floor});
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}
}

```

```

.....

package com.zhangminghsuai;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

/**
 * CreateDate: 2017-6-3 下午 10:17:11
 * Location: HIT
 * Author: Zhang Mingshuai
 * TODO
 * return
 */
public class Update {
    private String isbn;
    private String title;
    private String author;
    private String pr;
    private String pd;
    private String price;

```

```

private String house;
private String count;
private String room;
private String floor;

private String snumber;
private String sname;
private String saddr;

private String Aid;
private String Aname;
private String Aage;
private String Acountry;

public Update(String isbn,String title,String author,
              String pr,String pd,String price,String house,
              String count,String room,String floor){
    this.isbn = isbn;
    this.title = title;
    this.author = author;
    this.pr = pr;
    this.pd = pd;
    this.price = price;
    this.house = house;
    this.count = count;
    this.room = room;
    this.floor = floor;
}

public Update(String snumber,String sname,String saddr){
    this.snumber = snumber;
    this.sname = sname;
    this.saddr = saddr;
}

public Update(String Aid,String Aname,String Aage,String Acountry){
    this.Aid = Aid;
    this.Aname = Aname;
    this.Aage = Aage;
    this.Acountry = Acountry;
}

public String goBook(){
    Connection con = GetConnection.get();
    try {
        PreparedStatement stmt = con.prepareStatement("update book set Title
= ?, AuthorID = ?, Publisher = ?, PublishDate = ?, Price = ?"+" where ISBN = ?");
        stmt.setString(6, isbn);
        stmt.setString(1, title);
        stmt.setString(2, author);
        stmt.setString(3, pr);
    }
}

```

```

        stmt.setString(4, pd);
        stmt.setString(5, price);
        PreparedStatement stmt1 = con.prepareStatement("update house set
hnumber = ?, count = ? where isbn = ?");
        stmt1.setString(1, house);
        stmt1.setString(3, isbn);
        stmt1.setString(2, count);
        PreparedStatement stmt2 = con.prepareStatement("update room set
rnumber = ?, floor = ? where isbn = ?");
        stmt2.setString(1, room);
        stmt2.setString(3, isbn);
        stmt2.setString(2, floor);
        stmt.executeUpdate();
        stmt1.executeUpdate();
        stmt2.executeUpdate();
        return "Success Update!";
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "Fail";
    }
}

public String goStudent() {
    Connection con = GetConnection.get();
    try {
        PreparedStatement stmt = con.prepareStatement("update student set
sname = ?, saddr = ? where snumber = ?");
        stmt.setString(3, snumber);
        stmt.setString(1, sname);
        stmt.setString(2, saddr);
        stmt.executeUpdate();
        return "Success Update!";

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "Fail";
    }
}

public String goAuthor() {
    Connection con = GetConnection.get();
    try {
        PreparedStatement stmt = con.prepareStatement("update author set name
= ?, age = ?, country = ? where authorid = ?");
        stmt.setString(4, Aid);
        stmt.setString(1, Aname);
        stmt.setString(2, Aage);
        stmt.setString(3, Acountry);
        stmt.executeUpdate();
    }
}

```

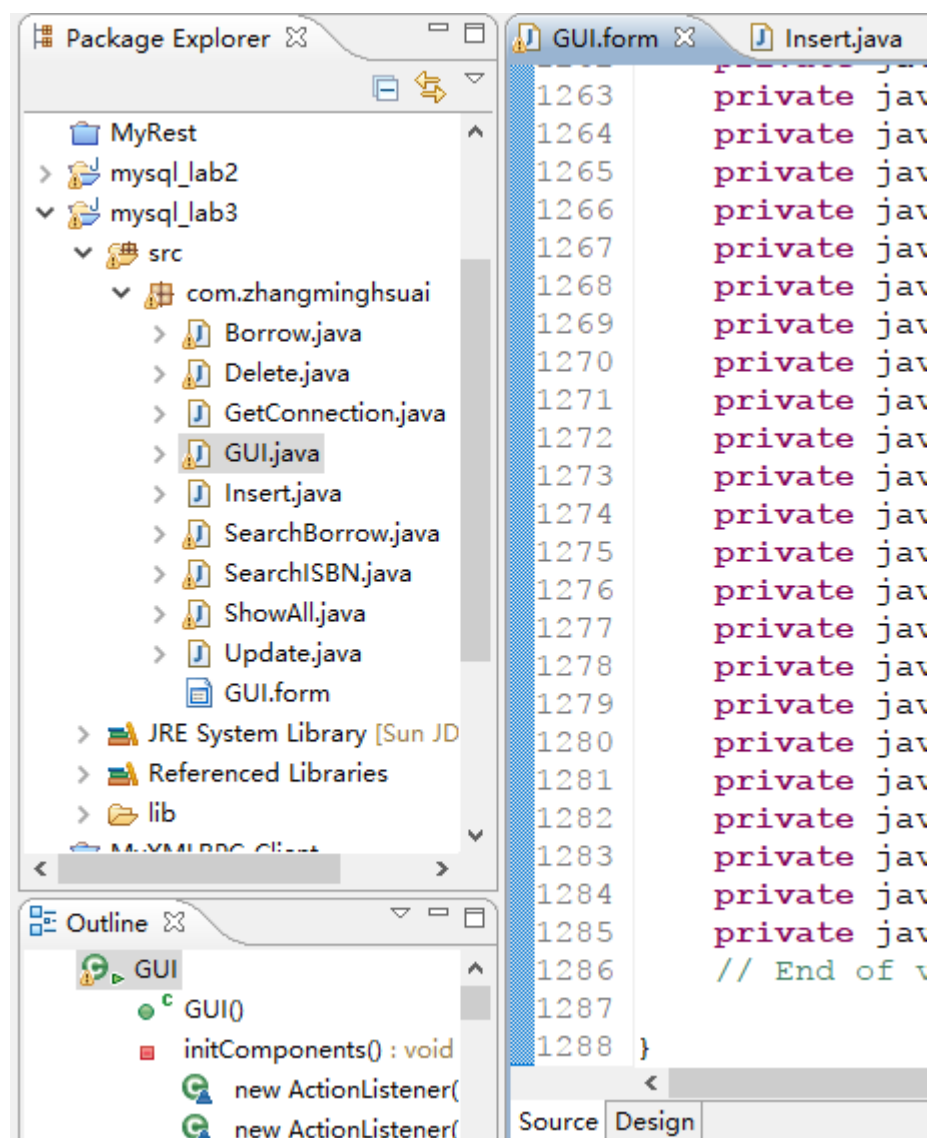
```

        return "Success Update!";

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return "Fail";
    }
}
}
}

```

GUI.java(我的 GUI 界面通过 matisse 插件基于 swing 下的 JFrame 框架进行制作的, 只需在面板上进行控件拖动, 界面做好后保存自动生成代码, 在此只给出核心按钮控制相关代码, 不予以全部显示, 因为自动生成的代码达到 1288 行, 如下图, 果真为 1288 行)



核心控制代码如下：

```
private void
searchbyISBNActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    jTextField1.setText("");
    DefaultTableModel myModel = (DefaultTableModel)
jTable1.getModel();
    myModel.setRowCount(0);
    SearchISBN mys = new SearchISBN(Sisbn.getText());
    jTextField1.setText(mys.goSearch(myModel));
}

private void
searchbySActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jTextField1.setText("");
    DefaultTableModel tModel = (DefaultTableModel)
jTable2.getModel();
    tModel.setRowCount(0);
    SearchBorrow mySearch = new
SearchBorrow(Sid.getText());
    jTextField1.setText(mySearch.goSearch(tModel));
}

private void
borrowActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jTextField1.setText("");
    Borrow myBorrow = new Borrow(Bid.getText(),
Bisdn.getText(), Bday
        .getText(), Bbegin.getText(), Bend.getText());
    jTextField1.setText(myBorrow.goBorrow());
}

private void
ButtonDeleteActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    jTextField1.setText("");
    if (jCheckBox1.isSelected()) {
        Delete myDelete1 = new Delete();

        jTextField1.setText(myDelete1.goBook(isbn.getText()));
    }
    if (jCheckBox2.isSelected()) {
        Delete myDelete2 = new Delete();

        jTextField1.setText(myDelete2.goAuthor(Aid.getText()))
    }
};
```

```

    }
    if (jCheckBox3.isSelected()) {
        Delete myDelete3 = new Delete();

        jTextField1.setText(myDelete3.goStudent(Snumber.getText()
t()));
    }
}

private void
ButtonUpdateActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    jTextField1.setText("");
    if (jCheckBox1.isSelected()) {
        Update myUpdate1 = new Update(isbn.getText(),
title.getText(),
        author.getText(), pr.getText(), pd.getText(),
price
        .getText(), house.getText(),
countnumber.getText(),
        room.getText(), floornumber.getText());
        jTextField1.setText(myUpdate1.goBook());
    }
    if (jCheckBox2.isSelected()) {
        Update myUpdate2 = new Update(Aid.getText(),
Aname.getText(), Aage
        .getText(), Acountry.getText());
        jTextField1.setText(myUpdate2.goAuthor());
    }
    if (jCheckBox3.isSelected()) {
        Update myUpdate3 = new Update(Snumber.getText(),
Sname.getText(),
        Saddr.getText());
        jTextField1.setText(myUpdate3.goStudent());
    }
}

private void
ButtonInsertActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    jTextField1.setText("");
    if (jCheckBox1.isSelected()) {
        Insert myInsert1 = new Insert(isbn.getText(),
title.getText(),
        author.getText(), pr.getText(), pd.getText(),
price
        .getText(), house.getText(),
countnumber.getText(),
        room.getText(), floornumber.getText());

```

```

        jTextField1.setText(myInsert1.goBook());
    }
    if (jCheckBox2.isSelected()) {
        Insert myInsert2 = new Insert(Aid.getText(),
Aname.getText(), Aage
        .getText(), Acountry.getText());
        jTextField1.setText(myInsert2.goAuthor());
    }
    if (jCheckBox3.isSelected()) {
        Insert myInsert3 = new Insert(Snumber.getText(),
Sname.getText(),
        Saddr.getText());
        jTextField1.setText(myInsert3.goStudent());
    }
}

private void
searchAllActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel tableModel = (DefaultTableModel)
jTable1.getModel();
    tableModel.setRowCount(0);
    ShowAll myshow = new ShowAll();
    myshow.goShow(tableModel);
}

```