夕拾市场使用说明书

大学生二手交易系统



夏雨柔 | 数据库 | 2017年5月25日

学号：201500800560

指导老师：衣振萍

# 夕拾市场起名来由

夕拾来源于鲁迅所写的唯一一部回忆[散文](http://iask.sina.com.cn/c/1098.html)集《朝花夕拾》，朝花夕拾，字面意思就是 早晨采得花（或者开的花），下午夕阳的时候重新拾起来。所以我希望二手书像朝花一样，能被别人夕拾、采摘，发挥它的剩余价值。

# 系统功能简要介绍

1. 搜索功能

* 搜索书籍
* 搜索订单（包括交换，买赠订单）

1. 收藏功能

* 收藏最喜欢的书籍（可撤销收藏）
* 把订单加入购物车收藏（可撤销收藏）

1. 买卖功能

* 根据搜索出的订单或在购物车的订单买入书籍
* 使用添加订单添加卖订单

1. 转赠功能

* 根据搜索出的订单或在购物车的订单夕拾书籍
* 用添加订单添加转赠订单

1. 交换功能

* 根据搜索出的订单或在购物车的订单交换书籍
* 用添加订单添加交换订单（一个用户只允许添加一个待交换订单）

1. 查看功能

* 查看买入订单（包括买入书籍和接受转赠）
* 查看卖出订单（包括已转赠和已卖）
* 查看已交换订单
* 查看发布订单（包括待售，待赠，待交换订单）
* 查看交易人信息
* 查看书籍信息
* 查看选定书籍的夕拾书单

1. 登录注册功能
2. 个人信息修改功能
3. 删除功能

* 删除发布的订单
* 从书架删除喜爱的书
* 从购物车删除收藏的订单

# 目录

主页介绍------------------------------------------------------------6

登录与注册 [6](#_登录与注册)

注册 [7](#_注册)

登录 11

搜索，购买，转赠，交换功能 11

搜索订单 11

购买书籍，转赠书籍 11

交换书籍 13

搜索书籍 [15](#_登录)

发布（添加订单）功能 [16](#注册与登录)

添加订单 [16](#_登录与注册)

查看功能（包括各页面操作） [20](#注册与登录)

查看图书信息 20

查看个人信息 [21](#_登录)

查看购物车 [22](#_登录)

查看收藏的书 [24](#_登录)

查看已发布的订单 25

查看已买入的订单（包含买入和转赠） [26](#_登录)

查看已卖出的订单（包含卖出和转赠） [28](#_登录)

查看已交换的订单 [29](#_登录)

收藏功能 [30](#注册与登录)

订单加入购物车 [30](#_登录)

收藏书籍 [31](#_登录)

其他功能 [31](#注册与登录)

修改本人信息/查看本人信息 [31](#_登录)

页面刷新 32

退出登录 32

关于夕拾 [32](#_登录)

系统详解 33

环境需求 33

设计与实现程度 33

项目展示 3[4](#_登录)

实现过程 3[7](#_登录)

总结与收获 37

参考资料 37

数据库设计 38

## 主页

夕拾市场图标

[书籍搜索](#_搜索功能)



[图书信息查看](#_查看书籍)

[退出登录](#_退出登录)

[关于夕拾](#_关于夕拾)

[个人信息](#_查看或修改个人信息)

[我的买入](#_查看已买入的订单（包含买入与转赠）)

[我的卖出](#_查看已卖出的订单（包含卖出与转赠）)

[我的交换](#_查看已交换的订单)

[我的发布](#_查看已发布订单)

[添加订单](#_添加订单)

[收藏的书](#_查看收藏的书)

[收藏的订单](#_查看购物车)

[登录注册](#_登录与注册)

热门收藏Top5

## 登录与注册

**1.点击主页上的此按钮**

**2．弹出登录与注册选择界面，如下图1.1**

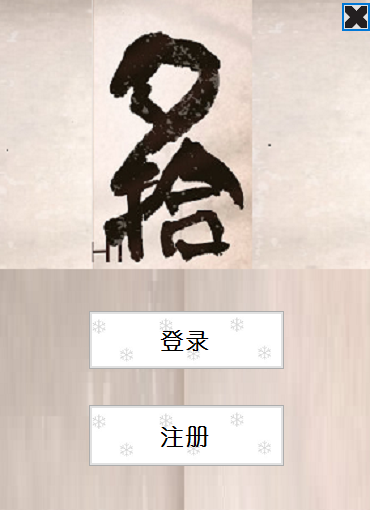
****

图1.1

**3.选择登录或注册**

### 注册

**3.1.1选择注册,点击注册按钮**

**注册页面如下图1.2**

****

图1.2

**3.1.1输入注册昵称与密码（这里以123456为例）**

****

**3.1.2 点击注册按钮**

****

**3.1.3 跳转填写个人信息，如图1.3所示**

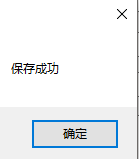
****

图1.3

**3.1.4 填写个人信息**

**3.1.5 填写完成后（手机号必填），点击保存按钮**

****

**弹出保存成功对话框，即为保存成功**

**若需再次修改请单击修改按钮**

****

**关闭此页面返回主页**

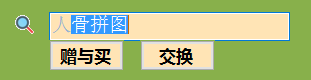
### 登录

进入登录界面如同上面的注册，具体请看[登录与注册](#_登录与注册)

**3.1.1 选择登录按钮**

## 搜索功能

1. **在主页上的搜索框内输入书名/作者名/出版社名/书籍标签名/书籍关键字名/**

****

1. **点击按钮选择需要进行买赠订单查询/交换订单查询**

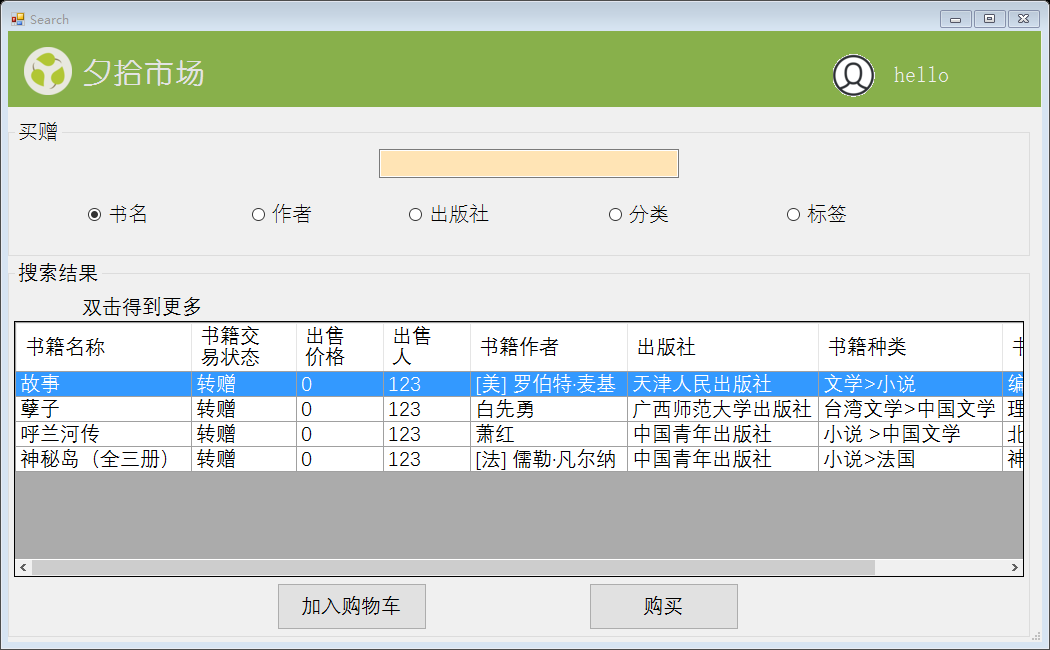
****

### 搜索买赠

1. **1选择买赠**

****

**3.1.1进入搜索页面，默认搜索书名为搜索框输入的文本**

****

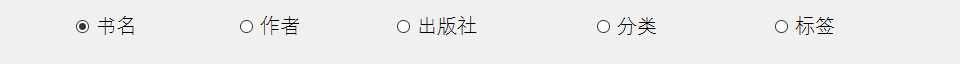
单击排序

选择查询条件

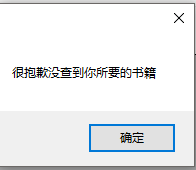
[双击查看选中图书信息](#_查看书籍)

默认为书名

**3.1.2若需搜索作者名/出版社名/书籍标签名/书籍关键字名，请选择搜索项(搜索书名不选)**

****

**3.1.3 查无订单提示**

****

**3.1.4 查到订单后选择操作**

**3.1.4 选择操作1：**[**查看书信息**](#_查看书籍)**（双击订单查看）**

**3.1.4 选择操作2：单击标题对订单排序**

****

* + 1. **选择操作3：加入购物车（此步需要**[**登录**](#_登录)**）**

****

**3.1.4 选择操作4：购买书籍（此步需要**[**登录**](#_登录)**）**

****

### 搜索交换

进入搜索交换如前面买赠，具体请看[搜索功能](#_搜索功能)

**3.2选择交换**

****

**3.2.1进入搜索页面，默认搜索书名为搜索框输入的文本**

选择查询条件

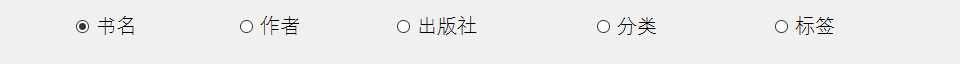
默认为书名

[双击查看选中图书信息](#_查看书籍)

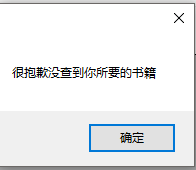
****

单击排序

**3.2.2若需搜索作者名/出版社名/书籍标签名/书籍关键字名，请选择搜索项(搜索书名不选)**

****

**3.2.3 查无订单提示**

****

**3.2.4 查到订单后选择操作**

**3.2.4 选择操作1：**[**查看书信息**](#_查看书籍)

**3.2.4 选择操作2：单击标题对订单排序**

****

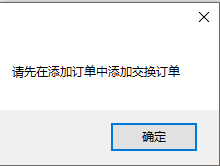
**3.2.4 选择操作3：加入购物车（此步需要**[**登录**](#_登录)**）**

****

**3.2.4 选择操作4：交换书籍（此步需要**[**登录**](#_登录)**）**

****

**Tips：用户必须有一个交换订单才能交换，不然会弹出**

****

### 搜索书籍

**退回主页点击收藏的书（此步需要**[**登录**](#_登录)**）**

****

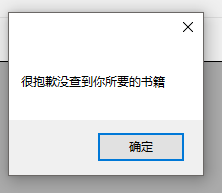
1. **在收藏的书上的搜索框内输入书名/作者名/出版社名/书籍标签名/书籍关键字名/**

****

1. **点击搜索按钮，进入书籍搜索页面**

****

**3.1.1 查无书籍提示**

****

**3.1.2 查到书籍后选择操作**

**3.1.2 选择操作1：**[**查看书信息**](#_查看书籍)

**3.1.2 选择操作2：点击标题对书籍排序**

****

## 发布（添加订单）

### 添加订单

1. **在主页点击添加订单按钮**

****

1. **进入订单添加页面**

书号先输入

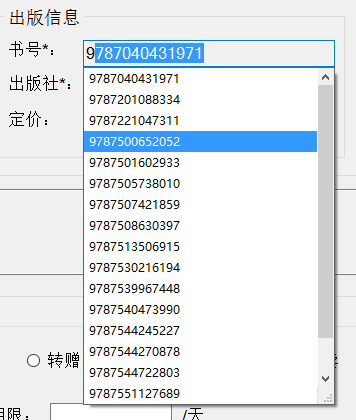
****

1. **先输入书号**

**3.1情况1：输入书号在系统已存在**

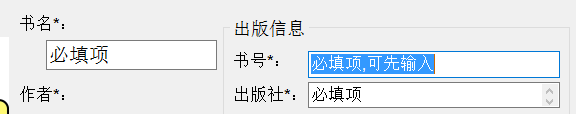
**3.1.1 选中系统中已有书号**

**3.1.2 系统自动填充信息（需等待）**



**3.2情况2：输入书号在系统不存在**

**3.1.1 填入必填项**



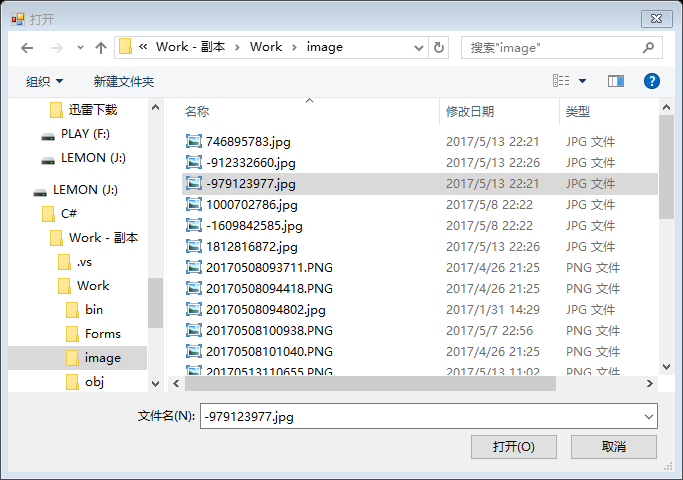
**3.1.2 选填其他非必填项**

**3.1.3 上传图书图片（此步可不上传）**

**3.1.3.1 双击添加图片**

****

* + - 1. **打开文件对话框选择合适书籍图片**

****

**3.1.3.3 点击确定图片就显示在添加订单页面**

**4.选择添加操作（交换/转赠/卖出）**

****

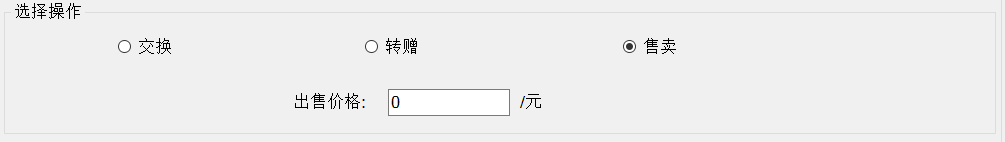
1. **若是交换填入交换日期，如图**

****

**若是转赠无需填写，如图**

****

**若是卖出则填入出售价格**

****

1. **点击确认添加按钮添加订单**

****

1. **跳转**[**我的发布页面**](#_查看已发布订单)

****

## 查看功能（包括各页面操作）

### 查看书籍信息

1. **查看书籍信息的方式**

**方式1：搜索页面的双击**

**方式2：书籍图片的双击**

1. **书籍页面介绍**

收藏按钮

****

买赠单

交换单

1. **书籍收藏方法**

**3.1书籍为未收藏，收藏按钮处于灰状**

****

**3.2收藏书籍，单击收藏按钮，按钮变为黄色**

****

**4.可在我收藏的书查看收藏的书，具体请看我**[**收藏的书**](#_查看收藏的书)

### 查看个人信息



1. **查看个人信息的方式（单击个人昵称）**

**方式1：在我的买入页面，查看售卖者以及转赠者信息**

**方式2：在我的卖出页面，查看买家以及夕拾者信息**

**方式3：在我的交换页面，查看夕拾者信息**

**Tips可通过手机号线下交易书籍**

### 查看购物车

1. **点击主页上的我的购物车按钮**

****

1. **打开购物车页面**

刷新

双击图片[查看图书信息](#_查看书籍)

****

此处根据选择操作进行变化

取消收藏按钮

选择操作

1. **首先根据想查看的订单类型选择需要查看的订单属于：**

**类型1：我收藏的交换**

**类型2：我收藏的转赠**

**类型3：我收藏的售卖**

1. **选择对订单的操作方式**

**操作1：取消收藏（点击取消收藏按钮）**

****

**操作2：** [**查看图书信息**](#_查看书籍)**（双击图片）**

**操作3：查看订单信息**

**操作4：可根据选择的操作类型选对收藏的订单进行进一步操作**

**类型1：交换（点击交换按钮）**

**Tips 完成后跳转**[**我的交换**](#_查看已交换的订单)**页面**

****

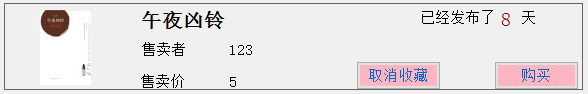
**类型2：转赠（点击转赠按钮）**

**Tips 完成后跳转**[**我的买入**](#_查看已买入的订单（包含买入与转赠）)**页面**

****

**类型3：购买（点击购买按钮）**

**Tips 完成后跳转**[**我的买入**](#_查看已买入的订单（包含买入与转赠）)**页面**

****

### 查看收藏的书

1. **点击主页上的收藏的书按钮**

****

1. **打开收藏的书页面**

刷新

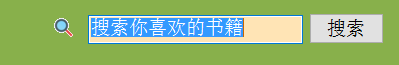
[搜索书籍](#_搜索书籍)

****

[双击查看图书信息](#_查看书籍)

1. **选择对书籍的操作方式**

**操作1：**[**搜索书籍**](#_搜索书籍)

****

**操作2：双击图书图片**[**查看图书信息**](#_查看书籍信息)

**Tips 如果对图书进行收藏或取消收藏，请刷新查看**

### 查看已发布订单

**1点击主页上的我的发布按钮**

****

**2打开我的发布页面**

刷新

[双击查看图书信息](#_查看书籍)

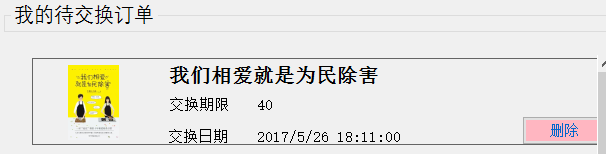
****

删除订单按钮

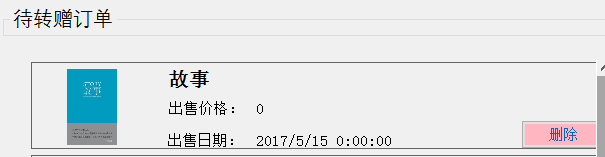
选择操作

**3首先根据想查看的订单类型选择需要查看的订单属于：**

**类型1：我的待交换**

****

**类型2：我的待转赠**

****

**类型3：我的待售卖**

****

**4选择对订单的操作方式**

**操作1：取消订单（点击删除按钮）**

****

**操作2：** [**查看图书信息**](#_查看书籍)**（双击图片）**

**操作3：查看订单信息**

### 查看已买入的订单（包含买入与转赠）

**1点击主页上的我买入的按钮**

****

1. **打开我买入的的页面**
2. 

[查看交易方信息](#_查看个人信息)

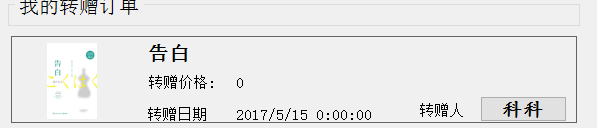
[双击查看图片信息](#_查看书籍)

操作选择

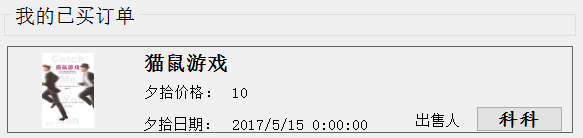
刷新

**3.首先根据想查看的订单类型选择需要查看的订单属于：**

**类型1：转赠订单**

****

**类型2：买入订单**

****

**4.选择对订单的操作方式**

**操作1：**[**查看交易方信息**](#_查看个人信息)**（双击交易方昵称）**

**操作2：** [**查看图书信息**](#_查看书籍)**（双击图片）**

**操作3：查看订单信息**

### 查看已卖出的订单（包含卖出与转赠）

**1点击主页上的我卖出的按钮**

****

**2.打开我卖出的的页面**

[查看交易方信息](#_查看个人信息)

[双击得到图片信息](#_查看书籍)

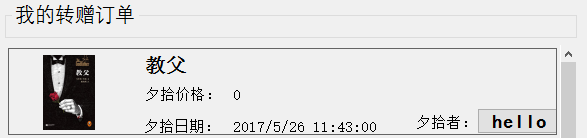
刷新

****

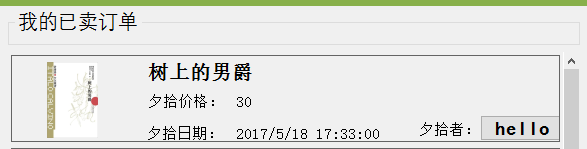
选择操作

**3.首先根据想查看的订单类型选择需要查看的订单属于：**

**类型1：已赠订单**

****

**类型2：已卖订单**

****

**4选择对订单的操作方式**

**操作1：**[**查看交易方信息**](#_查看个人信息)**（双击交易方昵称）**

**操作2：** [**查看图书信息**](#_查看书籍)**（双击图片）**

**操作3：查看订单信息**

### 查看已交换的订单

**1点击主页上的我的交换按钮**

****

1. **打开我的交换的页面**

单击得到[交换者信息](#_查看个人信息)

刷新

[双击得到图书信息](#_查看书籍)

****

[双击得到图书信息](#_查看书籍)

**3.选择对订单的操作方式**

**操作1：**[**查看交易方信息**](#_查看个人信息)**（双击交易方昵称）**

**操作2：** [**查看图书信息**](#_查看书籍)**（双击图片）**

**操作3：查看订单信息**

## 收藏功能

### 订单加入购物车

**1.通过选择搜索页面加入购物车（点击加入购物车按钮）**

[**搜索交换页面**](#_搜索交换)**：加入交换订单**

[**搜索买赠页面**](#_搜索交换)**：加入买赠订单**

### 收藏书籍

1. **书籍收藏方法**

**3.1书籍为未收藏，收藏按钮处于灰状**

**3.2收藏书籍，单击收藏按钮，按钮变为黄色**

**2.可在我收藏的书查看收藏的书，具体请看我**[**收藏的书**](#_查看收藏的书)

## 其他功能

### 查看或修改个人信息

**1点击主页上的个人信息按钮**

****

1. **打开个人信息的页面**

****

**3.选择对订单的操作方式**

**操作1：查看个人信息**

**操作2：**[**修改个人信息**](#_查看书籍)

**1点击修改按钮**

****

**2填入修改信息（手机号必填）**

**3点击保存按钮保存修改的信息**

****

### 页面刷新

1. **点击每个页面的刷新按钮**

****

### 退出登录

1. **点击主页的退出登录按钮**

****

### 关于夕拾

1. **点击主页的关于夕拾按钮**

****

1. **打开夕拾页面**

****

## 系统详解

### 运行环境

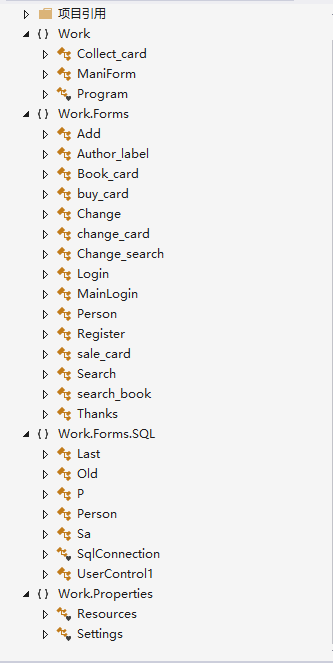
1需要在SQL server里面附加Change.mdf数据库名称为Change

2点开work.EXE文件即可运行

### 设计与实现程度

前期因为考虑个人的水平与时间暂定了三个功能（交换，转赠，买卖），后期在代码编写熟练的情况下，增加了收藏图书，加入购物车两个功能，总体来说，实现的与设计基本相符。不过有个尚未完成的遗憾，就是搜索页面没能设计成图片式的页面，希望自己在以后的时间能继续完成它。

### 项目展示

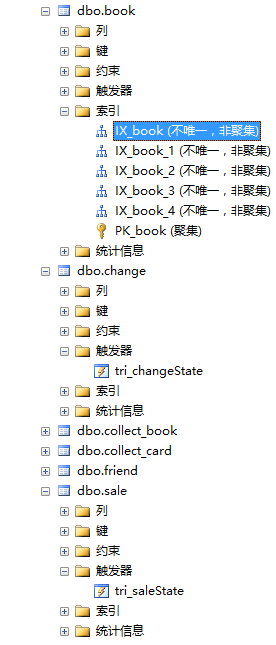


**类图**

总共25个类

数据库图

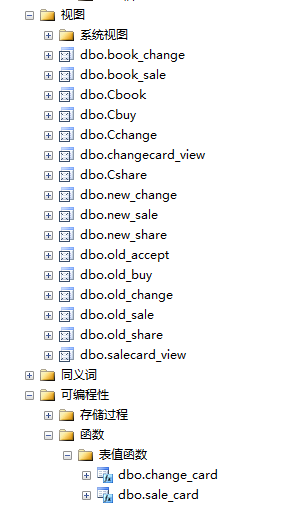
**表图**



总共2个触发器

总共6个表

总共6个索引



总共2个函数

总共16个视图

### 实现过程

### 总结与收获

首先很开心能完成自己写的第一个系统，从刚开始枯燥的数据库设计到最后的代码实现，一步一步脚踏实地，终于完成了它。

不足的是没有实现在一个窗口跳转页面，这是写该系统时最大的遗憾。

在完成系统中最大的感悟就是要把每次的程序，数据库小问题记录下来做成word文档，

多看几遍，减少自己的出错率。能在数据库课程设计中收获错误与知识，更加坚定了我从事软件专业的心。

### 参考资料

图片存取方法参考百度

数据库连接方式来自于王为老师的代码解释

自定义用户控件方法参考百度

## [数据库设计](J:\\C#\\夕拾市场数据库设计文档.docx)

### 需求分析

**市场需求**：

* 近年来校园二手书无法及时回收利用，本人买的书看完之后成闲置，又不想出售，自己想看的书在别人那成了闲置，提高二手书利用率迫在眉睫，网上现有的二手书需加上邮费，不符合二手书本身的价值
* **特色**：
* 增加对二手书的售卖交易
* 对于自己的成为闲置的书可采取转赠或与他人交换自己所需看的书
* 增加对书友专业的分类，便于快速找书
* 增加对书籍的标签管理
* **功能:**
* **书友管理**
* 书友注册
* 书友交换记录
* 书友信息记录，包括专业与联系方式
* 书友的增查删改
* **书籍管理**
* 书籍的书号，书名，类别的管理
* 书籍的状态管理（在售，转赠，交换，待购）
* 书籍的增查删改
* **书籍销售管理**
* 书籍的类别或模糊查询
* 书籍的上架与交换
* 对书籍的销售价格管理（交换需要交付押金，赠书无需押金，买书按卖家意愿决定）
* 书籍的标签管理，便于更快找书

**数据字典**

**书友管理子系统**

**属性：书友ID号 手机号 书友姓名 密码 专业 年级 性别**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **数据项** | **含义说明** | **别名** | **类型** | **长度** | **取值范围** | **取值含义** | **与其他数据项的关系** |
| **ID号** | **书友ID号** | **书友ID号** | **字符型** | **7** | **1111111**  **~9999999** | **书友的登录认证** | **主键** |
| **手机号** | **书友手机号** | **无** | **字符型** | **11** | **无** | **书友找回id凭证** | **与ID号一一对应**  **书友交易联系方式** |
| **姓名** | **书友姓名** | **无** | **字符型** | **20** | **无** | | |
| **性别** | **书友身份识别** | **无** | **字符型** | **2** | **男或女** |  | **可不填** |
| **密码** | **书友识别** | **无** | **字符型** | **18** | **无** | **权限授予** | **查询交换记录，销售等** |
| **专业** | **书友的专业** | **无** | **字符型** | **10** | **无** | | **便于其它书友找专业书** |
| **年级** | **书友的年级** | **无** | **整型** | **无** |

**书籍管理子系统**

**属性：书名，书号，类别，关键词，状态，原价，作者，出版社**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **数据项** | **含义说明** | **别名** | **类型** | **长度** | **取值范围** | **取值含义** | **与其他数据项的关系** |
| **书籍名称** | **标识书籍** | **无** | **字符型** | **20** | **无** | | **决定书籍的唯一标识**  **与书籍编号一一对应** |
| **书籍编号** | **标识书籍** | **书刊号** | **字符型** | **20** | **无** | **主键** | **与书籍名称一一对应**  **便于找书** |
| **价格** | **书籍原价** | **无** | **整型** | **无** | | | **便于书籍销售与交换押金确定** |
| **书籍状态** | **书籍状态** | **无** | **字符型** | **4** | **在售，转赠，已交换，待交换** | **确定书籍状态** | **与销售，交换子系统关联对应** |
| **书籍类别** | **书籍类别** | **无** | **字符型** | **无** | | **专业分类**  **常用分类** | **依赖于书友的专业**  **依赖于书友自定义的标签** |
| **书籍关键词** | **书籍关键词** | **无** | **字符型** | **20** |  | **可添加多个标签** | **便于书友找书** |
| **作者** | **书籍作者** | **无** | **字符型** | **20** | **不能为null** | **主属性** | **一本书籍一个作者** |
| **出版社** | **书籍出版社** | **无** | **字符型** | **30** |  | **主属性** | **一本书籍一个出版社** |

**销售管理子系统**

**属性：书籍编号 ID号 订单号 日期 销售价**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **数据项** | **含义说明** | | **别名** | | **类型** | | **长度** | | **取值范围** | | | **取值含义** | | **与其他数据项的关系** |
| **订单号** | **销售订单号** | | **无** | | **字符型** | | **9** | | **无** | | |  | | **主键** |
| **书籍编号** | | **标识书籍** | | **书刊号** | | **字符型** | | **20** | | **无** |  | | **同书籍管理的书籍编号** | |
| **ID号** | | **书籍所有者** | | **书友ID号** | | **字符型** | | **7** | | **1111111**  **~9999999** | **书友的销售认证** | | **同书友管理系统的书友ID号** | |
| **日期** | **书籍销售日期** | | **销售日期** | | **日期型** | | **无** | | **yyyy：mm：ss** | | | **记录书籍销售记录** | |  |
| **销售价** | **书籍的成交** | | **销售价格** | | **整型** | | **无** | |  | | | **区别于原价** | | **《=原价，转赠为0元** |

**书籍交换子系统**

**属性：订单号 书籍编号，iD号，日期，交换截止日期**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **数据项** | **含义说明** | | **别名** | | **类型** | | **长度** | **取值范围** | | **取值含义** | **与其他数据项的关系** |
| **订单号** | **交换订单号** | | **无** | | **字符型** | | **9** | **无** | |  | **主键** |
| **书籍编号** | **标识书籍** | | **书刊号** | | **字符型** | | **20** | **无** | | **主键** | **同书籍管理的书籍编号（此项有两方）** |
| **ID号** | | **书籍所有者** | | **书友ID号** | | **字符型** | **7** | | **1111111**  **~9999999** | **书友的交换认证** | **同书友管理系统的书友ID号（此项有两方）** |
| **日期** | **书籍交换日期** | | **销售日期** | | **日期型** | | **无** | **yyyy：mm：ss** | | **记录书籍交换记录** |  |
| **交换截止日期** | **书籍交换截止日期** | | **截止日期** | | **日期型** | | **无** | **yyyy：mm：ss** | | **计算交换日期** | **与日期对应** |

* **数据流图**

0层图



* 此0层图展示了本系统的主要功能：

赠书 换书 出售书 买书 交换书

* 为了帮助理解特设定了两个不同的书友：

书友1 与 书友2

1层图:对系统的阐释



* 此1层图中书友1是卖书，交换书，赠书的代表，而书友2代表买书，交换书的代表
* 此1层图是对本系统内部交易的描述

 2.1层图，书友管理子系统

* 本层图罗列了用户注册，登录，管理员管理非法用户

 2.2层图 书籍出售子系统

此层图展示图书的售卖系统

此层图包括赠与，买书，卖书三个事件

1.3层图 书籍交换子系统



此层图的书友1，书友2分别表示开始有交换意愿的书友，有待交换书籍的书友

此层图展示的是图书交换的数据流图

交易情况包括失败和成功信息返回给书友1

### 概念设计

* 分E-R图

实体1：书友

**属性**：**书**友ID号 手机号 书友姓名 密码 专业 年级 性别



实体2：图书

属性：书名，书号，类别，关键词，原价，作者，状态，出版社



实体3：销售订单

属性：书籍编号 ID号 订单号 日期 销售价



实体4：交换订单

属性：订单号 书籍编号，iD号，日期，交换截止日期



* 总E-R图



### 逻辑设计

书友信息表（书友ID号，昵称，手机号，书友姓名，密码，专业，年级，性别）

书籍信息表（书号，作者，出版社，书名，类别，关键词，原价，详细信息）

售卖信息表（订单号 书籍编号 ID号 日期 销售价，状态）

交换信息表（订单号 书籍编号，ID号，日期，交换截止日期，状态）

系统管理记录表（违规ID号，违规日期，违规类型）

收藏的书表（昵称，书号）

收藏的订单表（昵称，订单号，订单状态）

### 物理设计

书友信息建表

use Change;

CREATE TABLE friend

(

friend\_ID char(7) PRIMARY　KEY check(friend\_ID between 1111111 and 9999999 ),

friend\_phone char(11) NOT NULL,

friend\_name varchar(20),

friend\_sex char(2) check(friend\_sex in('男','女')),

friend\_password varchar(18) NOT NULL,

friend\_speciality Varchar(10),

friend\_grade int,

);

书籍信息建表

use Change;

CREATE TABLE book

(

book\_ID varchar(20),

book\_name varchar(20) NOT NULL,

book\_price smallint,

book\_author varchar(20),

book\_type varchar(18) ,

book\_publish Varchar(30),

book\_key varchar(20),

PRIMARY　KEY(book\_ID,book\_author,book\_publish),

);

售卖信息表建表

use Change;

CREATE TABLE sale

(

sale\_ID char(9) ,

book\_ID varchar(20) NOT NULL,

friend\_ID char(7),

sale\_price smallint,

sale\_date Date NOT NULL,

sale\_state varchar(4) check(sale\_state in('在卖','已卖','转赠','已赠')),

PRIMARY　KEY(sale\_ID),

FOREIGN KEY(friend\_ID ) REFERENCES friend(friend\_ID),

);

交换信息表建表

use Change;

CREATE TABLE change

(

friend\_ID char(7) ,

book\_ID varchar(20) NOT NULL,

change\_ID char(9),

change\_datedif smallint,

change\_date Date NOT NULL,

change\_state varchar(4) check(change\_state in('待交换',’已交换')),

PRIMARY　KEY(change\_ID),

FOREIGN KEY(friend\_ID ) REFERENCES friend(friend\_ID),

);

书籍查询建视图

1.交换视图

SELECT dbo.change.change\_ID, dbo.book.book\_name AS 书籍名称, dbo.book.book\_price AS 书籍原价,

dbo.change.change\_state AS 书籍交易状态, dbo.friend.friend\_nickname AS 出售人, dbo.book.book\_author AS 书籍作者,

dbo.book.book\_publish AS 出版社, dbo.book.book\_type AS 书籍种类, dbo.book.book\_key AS 书籍标签

FROM dbo.friend INNER JOIN

dbo.change ON dbo.friend.friend\_ID = dbo.change.friend\_ID INNER JOIN

dbo.book ON dbo.change.book\_ID = dbo.book.book\_ID

WHERE (dbo.change.change\_state IN ('待交换'))

2. 买赠视图

SELECT dbo.sale.sale\_ID, dbo.book.book\_name AS 书籍名称, dbo.book.book\_price AS 书籍原价,

dbo.sale.sale\_state AS 书籍交易状态, dbo.sale.sale\_price AS 出售价格, dbo.friend.friend\_nickname AS 出售人,

dbo.book.book\_author AS 书籍作者, dbo.book.book\_publish AS 出版社, dbo.book.book\_type AS 书籍种类,

dbo.book.book\_key AS 书籍标签

FROM dbo.friend INNER JOIN

dbo.sale ON dbo.friend.friend\_ID = dbo.sale.friend\_ID INNER JOIN

dbo.book ON dbo.sale.book\_ID = dbo.book.book\_ID

收藏的书的视图

SELECT dbo.book.book\_name AS name1, dbo.book.image\_path AS path1, dbo.collect\_book.nickname

FROM dbo.book INNER JOIN

dbo.collect\_book ON dbo.book.book\_ID = dbo.collect\_book.book\_ID

收藏的订单视图

售卖视图

SELECT a.nickname, dbo.friend.friend\_nickname AS n, a.card\_ID AS ID, b.sale\_price AS d, b.sale\_date AS date1,

c.book\_name AS b, c.image\_path AS p

FROM dbo.collect\_card AS a INNER JOIN

dbo.sale AS b ON a.card\_ID = b.sale\_ID INNER JOIN

dbo.book AS c ON b.book\_ID = c.book\_ID INNER JOIN

dbo.friend ON b.friend\_ID = dbo.friend.friend\_ID

WHERE (a.card\_state = '在卖')

转赠视图

SELECT a.nickname, dbo.friend.friend\_nickname AS n, a.card\_ID AS ID, b.sale\_price AS d, b.sale\_date AS date1,

c.book\_name AS b, c.image\_path AS p

FROM dbo.collect\_card AS a INNER JOIN

dbo.sale AS b ON a.card\_ID = b.sale\_ID INNER JOIN

dbo.book AS c ON b.book\_ID = c.book\_ID INNER JOIN

dbo.friend ON b.friend\_ID = dbo.friend.friend\_ID

WHERE (a.card\_state = '转赠')

交换视图

SELECT a.nickname, dbo.friend.friend\_nickname AS n, a.card\_ID AS ID, b.change\_datedif AS d, b.change\_date AS date1,

c.book\_name AS b, c.image\_path AS p

FROM dbo.collect\_card AS a INNER JOIN

dbo.change AS b ON a.card\_ID = b.change\_ID INNER JOIN

dbo.book AS c ON b.book\_ID = c.book\_ID INNER JOIN

dbo.friend ON b.friend\_ID = dbo.friend.friend\_ID

WHERE (a.card\_state = '待交换')

我的发布视图

售卖视图

SELECT dbo.friend.friend\_nickname, dbo.sale.sale\_ID AS 订单号, dbo.book.book\_name AS name1,

dbo.sale.sale\_price AS price1, dbo.sale.sale\_date AS date1, dbo.book.image\_path AS path1

FROM dbo.sale INNER JOIN

dbo.book ON dbo.sale.book\_ID = dbo.book.book\_ID INNER JOIN

dbo.friend ON dbo.sale.friend\_ID = dbo.friend.friend\_ID

WHERE (dbo.sale.sale\_state = '在卖')

转赠视图

SELECT dbo.friend.friend\_nickname, dbo.sale.sale\_ID AS 订单号, dbo.book.book\_name AS name1, dbo.sale.sale\_date AS date1,

dbo.book.image\_path AS path1

FROM dbo.sale INNER JOIN

dbo.book ON dbo.sale.book\_ID = dbo.book.book\_ID INNER JOIN

dbo.friend ON dbo.sale.friend\_ID = dbo.friend.friend\_ID

WHERE (dbo.sale.sale\_state = '转赠')

交换视图

SELECT dbo.friend.friend\_nickname, dbo.change.change\_ID AS 订单号, dbo.book.book\_name AS name1,

dbo.change.change\_datedif AS price1, dbo.change.change\_date AS date1, dbo.book.image\_path AS path1

FROM dbo.change INNER JOIN

dbo.book ON dbo.change.book\_ID = dbo.book.book\_ID INNER JOIN

dbo.friend ON dbo.change.friend\_ID = dbo.friend.friend\_ID

WHERE (dbo.change.change\_state = '待交换')

我的买入视图

已买视图

SELECT dbo.sale.sale\_ID, dbo.sale.friend\_ID1 AS friend\_ID, dbo.book.book\_name AS name1, dbo.sale.sale\_price AS price1,

dbo.sale.sale\_date AS date1, dbo.friend.friend\_nickname AS nickname, dbo.book.image\_path AS path1

FROM dbo.book INNER JOIN

dbo.sale ON dbo.book.book\_ID = dbo.sale.book\_ID INNER JOIN

dbo.friend ON dbo.sale.friend\_ID = dbo.friend.friend\_ID

WHERE (dbo.sale.friend\_ID <> dbo.sale.friend\_ID1) AND (dbo.sale.sale\_state = '已卖')

转赠视图

SELECT dbo.sale.sale\_ID, dbo.sale.friend\_ID1 AS friend\_ID, dbo.book.book\_name AS name1, dbo.sale.sale\_price AS price1,

dbo.sale.sale\_date AS date1, dbo.friend.friend\_nickname AS nickname, dbo.book.image\_path AS path1

FROM dbo.book INNER JOIN

dbo.sale ON dbo.book.book\_ID = dbo.sale.book\_ID INNER JOIN

dbo.friend ON dbo.sale.friend\_ID = dbo.friend.friend\_ID

WHERE (dbo.sale.friend\_ID <> dbo.sale.friend\_ID1) AND (dbo.sale.sale\_state = '已赠')

我的已交换视图

SELECT a2.friend\_nickname AS n, b2.friend\_nickname AS n1, a1.book\_name AS b, b1.book\_name AS b1, a1.image\_path AS p,

b1.image\_path AS p1, a.change\_datedif AS d, b.change\_datedif AS d1, a.change\_date AS date1, a.change\_ID AS id,

b.change\_ID AS id1

FROM dbo.change AS a INNER JOIN

dbo.change AS b ON a.change\_date = b.change\_date AND a.friend\_ID = b.friend\_ID1 AND

a.friend\_ID1 = b.friend\_ID INNER JOIN

dbo.book AS a1 ON a.book\_ID = a1.book\_ID INNER JOIN

dbo.book AS b1 ON b.book\_ID = b1.book\_ID INNER JOIN

dbo.friend AS a2 ON a.friend\_ID = a2.friend\_ID INNER JOIN

dbo.friend AS b2 ON b.friend\_ID = b2.friend\_ID

我的卖出视图

已售视图

SELECT dbo.sale.sale\_ID, dbo.sale.friend\_ID, dbo.book.book\_name AS name1, dbo.sale.sale\_price AS price1,

dbo.sale.sale\_date AS date1, dbo.friend.friend\_nickname AS nickname, dbo.book.image\_path AS path1

FROM dbo.book INNER JOIN

dbo.sale ON dbo.book.book\_ID = dbo.sale.book\_ID INNER JOIN

dbo.friend ON dbo.sale.friend\_ID1 = dbo.friend.friend\_ID

WHERE (dbo.sale.friend\_ID <> dbo.sale.friend\_ID1) AND (dbo.sale.sale\_state = '已卖')

已转赠视图

SELECT dbo.sale.sale\_ID, dbo.sale.friend\_ID, dbo.book.book\_name AS name1, dbo.sale.sale\_price AS price1,

dbo.sale.sale\_date AS date1, dbo.friend.friend\_nickname AS nickname, dbo.book.image\_path AS path1

FROM dbo.book INNER JOIN

dbo.sale ON dbo.book.book\_ID = dbo.sale.book\_ID INNER JOIN

dbo.friend ON dbo.sale.friend\_ID1 = dbo.friend.friend\_ID

WHERE (dbo.sale.friend\_ID <> dbo.sale.friend\_ID1) AND (dbo.sale.sale\_state = '已赠')

交换单视图

SELECT dbo.change.change\_ID, dbo.book.book\_name, dbo.friend.friend\_nickname, dbo.change.change\_date

FROM dbo.change INNER JOIN

dbo.friend ON dbo.change.friend\_ID1 = dbo.friend.friend\_ID INNER JOIN

dbo.book ON dbo.change.book\_ID = dbo.book.book\_ID

买赠单视图

SELECT dbo.sale.sale\_ID, dbo.book.book\_name, dbo.friend.friend\_nickname, dbo.sale.sale\_date

FROM dbo.sale INNER JOIN

dbo.book ON dbo.sale.book\_ID = dbo.book.book\_ID INNER JOIN

dbo.friend ON dbo.sale.friend\_ID1 = dbo.friend.friend\_ID

买卖书籍查询索引建立

书籍名称·索引

use Change;

CREATE CLUSTERED

INDEX book\_name ON book\_sale

Book\_name ASC

书籍类别·索引

CREATE

INDEX book\_type ON book\_sale

Book\_name ASC

书籍作者·索引

CREATE

INDEX book\_author ON book\_sale

Book\_name ASC

书籍标签·索引

CREATE CLUSTERED

INDEX book\_key ON book\_sale

Book\_name ASC

交换书籍查询索引建立

书籍名称·索引

use Change;

CREATE CLUSTERED

INDEX book\_name ON book\_change

Book\_name ASC

书籍类别·索引

CREATE

INDEX book\_type ON book\_change

Book\_name ASC

书籍作者·索引

CREATE

INDEX book\_author ON book\_change

Book\_name ASC

书籍标签·索引

CREATE CLUSTERED

INDEX book\_key ON book\_change

Book\_name ASC

对出售书籍状态改变

CREATE Trigger tri\_saleState

ON sale INSTEAD OF UPDATE

AS

Declare @state varchar(4)

SELECT @state=sale\_state FROM deleted

IF(@state='在卖')

BEGIN

UPDATE sale SET sale.sale\_state='已卖'FROM deleted,sale WHERE sale.sale\_ID= deleted.sale\_ID

UPDATE sale SET sale.sale\_date=GetDATE() FROM deleted,sale WHERE sale.sale\_ID= deleted.sale\_ID

END

IF(@state='转赠')

BEGIN

UPDATE sale SET sale.sale\_state='已赠'FROM deleted,sale WHERE sale.sale\_ID= deleted.sale\_ID

UPDATE sale SET sale.sale\_date=GetDATE() FROM deleted,sale WHERE sale.sale\_ID= deleted.sale\_ID

END

IF(@state='已卖')

BEGIN

print'该书籍已出售'

ROLLBACK

END

IF(@state='转赠')

BEGIN

print'该书籍已转赠'

ROLLBACK

END

对交换书籍状态改变

CREATE Trigger tri\_changeState

ON change INSTEAD OF UPDATE

AS

Declare @state varchar(4)

SELECT @state=change\_state FROM INSERTED

IF(@state='待交换') UPDATE change SET change.change\_state='已交换' FROM sale,inserted WHERE change.change\_ID=inserted.change\_ID

IF(@state='已交换') print'该书籍已交换'

创建买卖书单函数

CREATE FUNCTION sale\_card(@name varchar(20))

RETURNS TABLE

AS

RETURN SELECT top 20 book\_name as '书的名字',friend.friend\_nickname as '赠予者',friend.friend\_nickname as '夕拾者',sale.sale\_date as '夕拾时间'

FROM sale,book,friend

WHERE book\_name=@name AND book.book\_ID=sale.book\_ID AND friend.friend\_ID=sale.friend\_ID AND friend.friend\_ID=sale.friend\_ID1

ORDER BY sale.sale\_date DESC

创建交换单函数

CREATE FUNCTION change\_card(@name varchar(20))

RETURNS TABLE

AS

RETURN SELECT top 20 book\_name as '书的名字',friend.friend\_nickname as '交换者',friend.friend\_nickname as '夕拾者',change.change\_date as '夕拾时间'

FROM change,book,friend

WHERE book\_name=@name AND book.book\_ID=change.book\_ID AND friend.friend\_ID=change.friend\_ID AND friend.friend\_ID=change.friend\_ID1

ORDER BY change.change\_date DESC