

成

绩



# 课程设计（论文）

课程名称 数据库课程设计

题目名称\_ 电脑书店信息管理系统

学生院系 人工智能与信息工程系

专业班级 21级计算机科学与技术7班

学 号\_\_\_ \_\_\_\_\_\_\_\_\_\_

学生姓名\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

指导教师

2023年 11月

目录

[1.绪论 2](#_Toc123844829)

[1.1 电脑书店信息管理系统 2](#_Toc123844830)

[1.2 发展现状 2](#_Toc123844831)

[1.3 可行性分析 2](#_Toc123844832)

[1.4 报告结构 3](#_Toc123844833)

[2. 需求分析 4](#_Toc123844835)

[2.1需求分析目标 4](#_Toc123844836)

[2.2数据字典 4](#_Toc123844837)

[2.3数据流图 5](#_Toc123844838)

[3. 概念结构设计 6](#_Toc123844839)

[4. 逻辑结构设计 7](#_Toc123844840)

[4.1 设计关系模型并对其进行优化分析 7](#_Toc123844841)

[4. 2系统功能模块图 8](#_Toc123844843)

[4. 3安全性设计 9](#_Toc123844845)

[4. 4完整性设计 9](#_Toc123844846)

[4. 4.1 主码，外码，其他约束 9](#_Toc123844847)

[4.4.2 触发器设计 10](#_Toc123844849)

[5.数据库物理设计 10](#_Toc123844850)

[5.1选择建立索引； 11](#_Toc123844851)

[6.数据库实施 11](#_Toc123844853)

[6.1创建数据库及数据库对象 11](#_Toc123844854)

[6.1.1 数据库创建 11](#_Toc123844855)

[6.1.2 基本表创建 12](#_Toc123844856)

[6.1.3应用程序 13](#_Toc123844862)

[7. 系统测试方案和测试报告 31](#_Toc123844863)

7 .1[系统测试 38](#_Toc123844864)

# 

# 1.绪论

## 1.1 电脑书店信息管理系统

信息技术的迅速发展加快了书店现代化与信息化的进程，书店的信息化管理已经成为一种发展趋势。传统的书店信息管理，一般依赖于一个非自动化以纸张文件为基础的系统来记录书籍的购买、进货，因此书店信息的管理的效率低下，所能管理的书店规模也很小，耗费人力资源。总的来说，传统书店信息管理存在以下这些问题：获取书籍库存信息速度慢，即在书店经营过程中，由于书籍种类多、数量大，查询各种书籍库存量等信息十分费时，而且易出错；书籍库存情况更新不准确或滞后，即书店管理员根据各种送货单、退货单以及的入库、出库搬运后，应当要随时修改库存信息，但由于书籍种类多、数量大，使得库存记录和实际库存常常不是严格一致的，这就导致了库存信息不正确或更新时间晚于实际库存变化情况；随着书店规模的不断扩大，书籍数量、种类的急剧增加，书店信息处理越来越复杂，信息数据量也更为庞大。面对庞大的书籍信息量，如何有效地管理书籍,对这些书店来说是非常重要的，所以这就需要由电脑书店信息管理系统来提高信息管理工作的效率，减少管理方面的工作量，同时对于调动员工的工作积极性，提高书店的经营效率，都具有十分重要的现实意义。针对电脑书店管理系统的问题，对电脑书店信息管理系统进行研究分析，然后开发设计出电脑书店管理系统系统以解决问题。

## 1.2 发展现状

由于传统书店信息管理存在的各种问题难于适应现代信息管理要求，通过查阅文献，所以现代书店信息管理系统要具有以下的特点：

1. 科学的库存管理流程

一般而言，书店信息管理包括书籍到货处理、书籍售卖和生成订单三个主要部分。通畅的业务流程是保障高效书店信息管理的基础。

1. 科学的组织结构

科学的组织结构确保了业务流程的高效执行，明确了书籍存储、存储种类、书籍编码之间的合理关系。

## 1.3 可行性分析

技术可行性：本电脑书店信息系统为书店信息管理人员开发，基于C#、

SQL server数据库，C#语句简单易用，学习成本低，既可以开发小工具，也可以开发小型管理系统，SQL server具有高性能，高可用，高安全，易运维等特点，开发者掌握C#编程，经过一学期的数据库系统学习掌握了如何设计、创建、查询和维护数据库，包括 SQL 语句、 存储过程、系统表和视图等基础知识，认识了SQL server的特性和优势，能够实现该电脑书店信息管理系统的开发。

经济可行性：计算机只要可以正常的使用进行代码的编写及页面设计就可行，PC机已经通过软件Visual Studio 2022安装部署SQL server实验环境，没有其他费用支出。

操作可行性：书店管理员可以通过登录界面进入系统，进而开始进行书籍信息管理，包括入库、出库的操作和各种信息查询等，通过界面导航菜单可以简单明了地操作功能模块，方便用户信息的操作需求和管理员管理数据信息，书店管理员只要会简单的电脑操作就完全可以使用，不需要专业人员协助，所以系统是可操作的。

## 1.4 报告结构

第1章是绪论，这部分介绍了什么是电脑书店信息管理系统，书店信息管理系统的发展现状，最后给出了本文的结构；

第2章是需求分析，需求分析结果转换为数据字典和数据流图；

第3章是概念结构设计，给出了该系统的ER图；

第4章是逻辑结构设计，给出了系统的关系模式；

第5章是数据库物理设计；

第6章是数据库实施，给出了数据库的具体创建sql代码；

第7章是系统测试。

# 

# 2、需求分析

## 2.1需求分析目标

本系统的目标主要是实现书店的信息管理，实现对书籍的出库、入库和书籍信息的查询、用户购买书籍、生成订单等功能，本系统的使用者为书店管理员、购买书籍的用户，通过使用电脑书店信息管理系统，书店管理员从原来的传统记录、查询方式中解脱出来，实现对书店信息的高效管理。用户更方便购买书籍，可以加入购物车，确认提交书籍订单后直接生成订单。

表1 系统目标

|  |  |
| --- | --- |
| 目标类型 | 子目标 |
| 书籍入库 | 添加书籍，更新库存 |
| 书籍出库 | 更新库存，生成订单 |
| 查询书籍信息 | 查询书籍种类、标题等，查询库存 |
| 修改信息 | 修改书籍、用户信息 |

## 2.2数据字典

数据字典是关于数据的信息的集合，是组成数据库很重要的组成部分，包含了数据项，数据结构等内容，只有认真分析相关的业务功能，进行全面细致的分析，才有可能得到全面的数据结构。数据运行才能够正常，不会出现缺少相应的信息等问题。数据项汇总表如表2所示，数据结构如表3所示。

表2 数据项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名 | 数据类型 | 长度 | 别名 | 取值范围 |
| Adid | INT |  | 管理员编号 |  |
| Adname | VARCHAR | 20 | 管理员姓名 |  |
| Adpass | VARCHAR | 20 | 管理员密码 |  |
| BId | INT |  | 书籍编号 | 大于100 |
| BTitle | NVARCHAR | 50 | 书籍名称 |  |
| BAuthor | NVARCHAR | 50 | 作者 |  |
| BCat | NVARCHAR | 50 | 书籍类型 |  |
| BQty | INT |  | 数量 | 大于0 |
| BPrice | INT |  | 价格 | 大于0 |
| UId | INT |  | 用户编号 | 大于1 |
| UName | NVARCHAR | 20 | 用户名 |  |
| UPhone | NVARCHAR | 50 | 电话号码 |  |
| UAdd | NVARCHAR | 50 | 地址 |  |
| UPassward | NVARCHAR | 50 | 密码 |  |
| BillId | INT |  | 订单号 | 大于1000 |
| Amount | INT |  | 金额 | 大于0 |

表3 数据结构

|  |  |  |
| --- | --- | --- |
| 数据结构名 | 含义说明 | 组成 |
| Admin | 管理员信息 | 编号，姓名，密码 |
| Books | 书籍信息 | 书籍编号，书籍名称，作者，书籍类型，数量，价格 |
| User | 用户信息 | 用户编号，用户名，电话号码，地址，密码 |
| Bills | 订单信息 | 订单号，书名，价格，数量，金额 |

## 2.3数据流图

数据流图描绘了信息流和数据从输入到输出的过程中所经受的变化，包含了对数据源点/终点、数据流、变换数据的处理、数据存储的描述。基于对书店信息管理系统的基本需求分析：书店管理员依据书店实际运营需求，将入库书籍、出库书籍记录到系统中，其中包含了书籍类型、价格等信息，录入之后，用户可以查询想要购买的书籍，选择想要购买的书籍加入购物车下单购买。

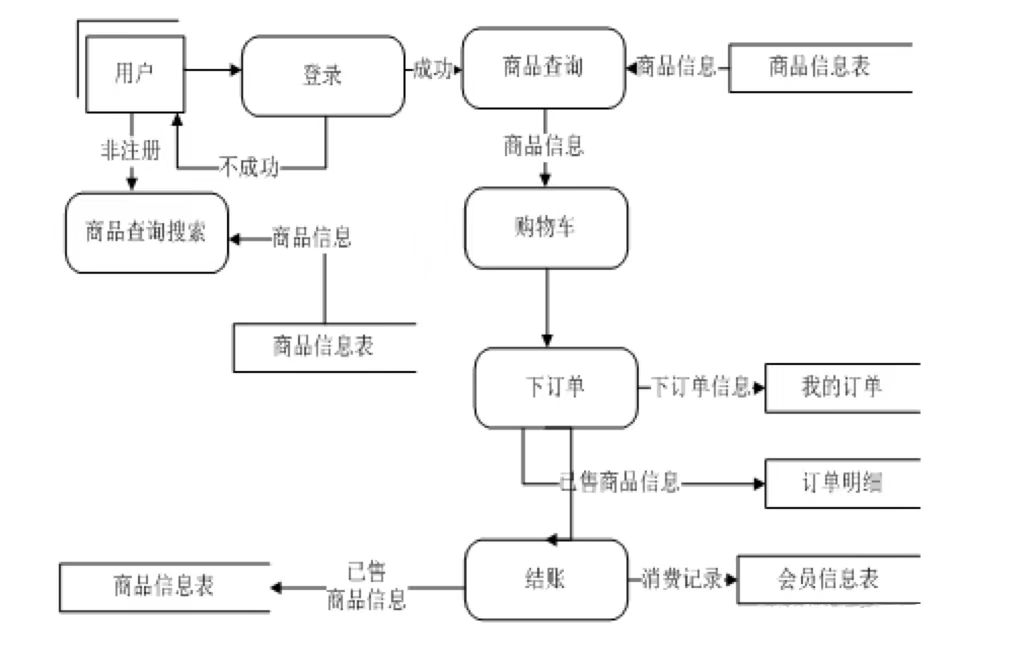


图1 数据流图

# 概念结构设计

概念设计是独立于数据库管理系统的设计，它的主要任务即时完成对现实事物，事物关系之间的转化，把抽象的事物转化成能够被人们易于理解的图形关系，更加直白的把现实的事物关系表达出来，从而为下一步的设计打下一个良好的基础。

识别书店信息管理信息系统中的实体，识别实体的属性，识别实体的关键字，识别实体间的联系，利用实体关系图（E—R图）来描述选题管理系统相关实体、属性及关系，从而为仓库管理信息系统建立良好的数据模型。根据前面的需求分析，包括以及相应的数据项，数据结构之间的关系，可以确定各包含不同实体的实体集。

每个实体定义的属性如下：

管理员（管理员编号，管理员姓名，密码）

书籍（书籍编号，书籍名称，作者，书籍类型，数量，价格）

用户（用户编号，用户名，电话号码，地址，密码）

订单（订单号，书名，价格，数量，金额）

其中带下划线的属性为该实体的主键。

概念设计的确定了基本实体以及相关的属性，并分别进行了主键的命名，就可以根据管理流程以及数据流图确定他们之间的关系，进行E-R图的设计，本书店信息管理系统的ER图如下图2。

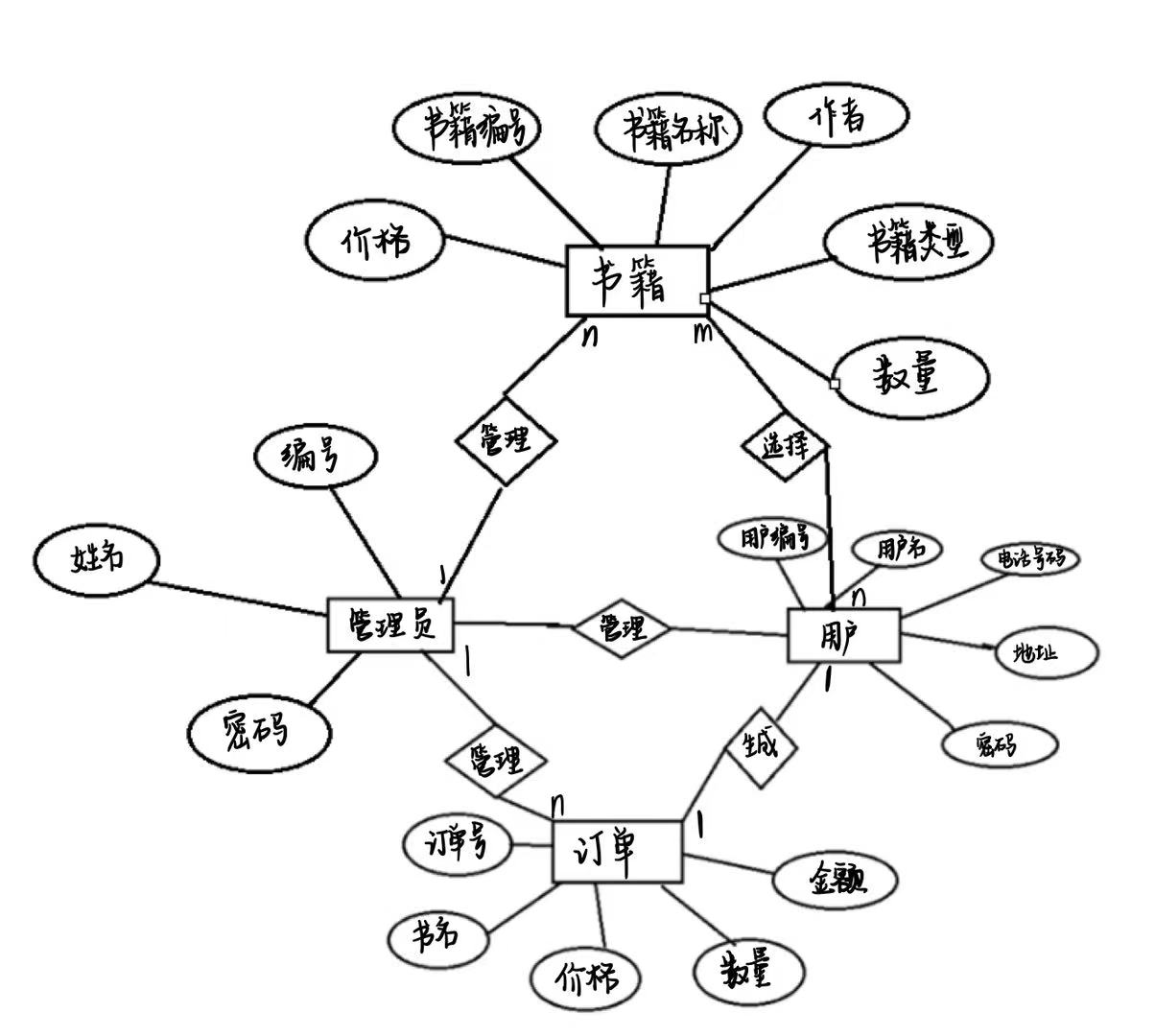


图2 网上书店信息管理系统ER图

4、逻辑结构设计

以上的概念设计阶段是独立于任何一种数据模型的，但是逻辑设计阶段就需要和选用的DBMS产品联系，逻辑结构设计的任务就是将概念设计阶段设计好的基本E-R图转换为选用DBMS产品所支持的数据模型相符合的逻辑结构。具体内容包括数据组织（将E-R图转换成关系模型、模型优化、数据库模式定义）、数据处理（画出系统功能模块图、系统功能描述）两大任务。

## 4.1 设计关系模型并对其进行优化分析

把ER模型转换成相应的关系表结构，每个关系模型的范式应最好满足第三范式，只有这样的关系模式才可能尽可能的减小冗余，达到较好的效果，各个关系模式和范式分析如下：

1.分析关系模式：管理员（管理员编号，管理员姓名，密码）

候选码：管理员编号

主属性：管理员编号

非主属性：管理员姓名，密码

假设姓名不唯一，那么所有函数依赖的决定因素均含有码，为BCNF。

2.分析关系模式：书籍（书籍编号，书籍名称，作者，书籍类型，数量，价格）

候选码：书籍编号

主属性：书籍编号

非主属性：书籍名称，作者，书籍类型，数量，价格

假设书籍名称不唯一，所有函数依赖的决定因素均含有码，为BCNF。

1. 分析关系模式：用户（用户编号，用户名，电话号码，地址，密码）

候选码：用户编号

主属性：用户编码

非主属性：用户名，电话号码，地址，密码

假设用户名可以重复，所有函数依赖的决定因素均含有码，为BCNF。

4.分析关系模式：订单（订单号，书名，价格，数量，金额）

候选码：订单号

主属性：订单号

非主属性：书名，价格，数量，金额

所有函数依赖的决定因素均含有码,为BCNF。

因此，以上4个关系模式均为BCNF，本文认为无需再优化，数据模型为选用DBMS产品所支持，可以直接使用。

## 4.2系统功能模块图

电脑书店信息管理系统的开发面向书店管理员、购书用户，主要功能分为管理员功能模块和用户功能模块，管理员功能中，书店管理员可以对书籍作出库和入库操作，进行操作的同时相应书籍的库存将更新，除此之外，书店管理员可以对书籍信息进行添加和修改。同理，也可以添加和修改购书用户信息；仓库管理员可以对出库、入库记录和客户信息进行查询，可以查询每种书籍当前的库存量。

用户功能中，用户可以查看可购买的书籍、选择需要购买的书籍加入购物车最后生成购买书籍的订单和金额。管理员、用户模块图如下：

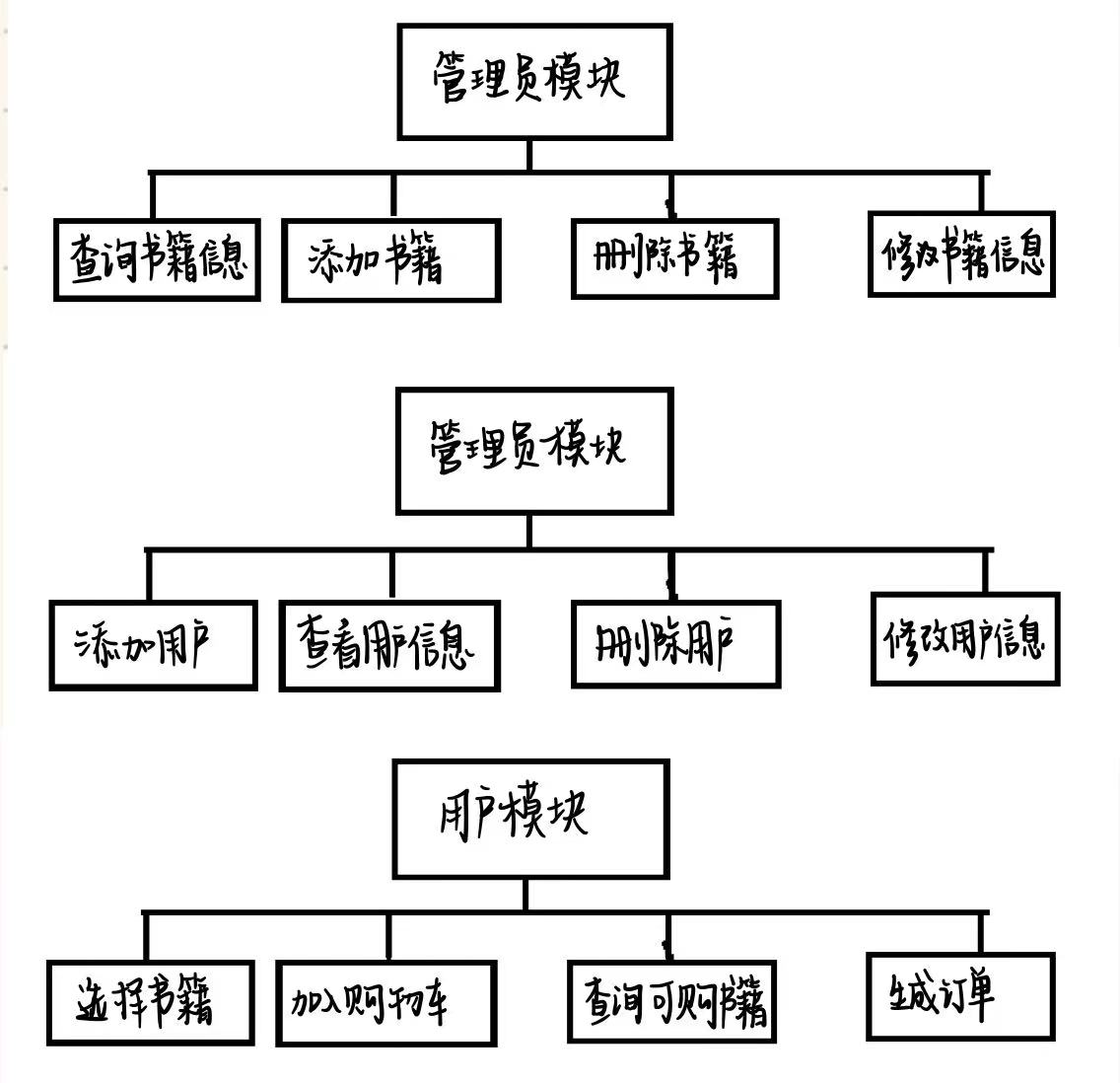


图3 电脑书店信息管理系统功能模块图

## 4.3 安全性设计

电脑书店信息管理系统的使用者书店管理员，拥有数据库对象级别权限（作用于指定的数据库对象上（表或者视图），他们的权限包括：向基本表添加、修改、查看和删除数据，但不允许创建、删除表或者修改表结构，不能创建、删除触发器，不能关闭数据库实例等。

（1）对数据库做定期备份（保存SQL文件，最好异地也备份，防止数据库崩溃），另外对配置二进制文件，数据库产生故障可通过二进制文件进行恢复。这样才能保障数据库一定的安全性。

## 4.4完整性设计

数据库完整性包括实体完整性、参照完整性和用户定义完整性，本节给出针对仓库管理数据库的完整性设计。

### 4.4.1 主码、其他约束

仓库管理数据库数据模型完整性约束如下表3。

表3 电脑信息管理系统完整性约束

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段含义 | 约束类型和描述 | 类型 |
| Admin.Adid | 管理员编号 | 主码，自增 | 实体完整性 |
| Admin.Adname | 管理员姓名 |  | 用户定义 |
| Books.BId | 书籍编号 | 主码，自增 | 实体完整性 |
| Books.BQty | 数量 | 大于0 | 用户定义 |
| Books.BPrice | 价格 | 大于0 | 用户定义 |
| User.UId | 用户编号 | 主码，自增 | 实体完整性 |
| User.UName | 用户名 |  | 用户定义 |
| User.UPhone | 用户电话 |  | 用户定义 |
| User.UAdd | 用户地址 |  | 用户定义 |
| Bills.BillId | 订单编号 | 主码 | 实体完整性 |

### 4.4.2触发器设计

触发器时定义在表上的一类由时间驱动的特殊过程，一旦定义，触发器就会保存在数据库服务器中，任何用户对表的增删改操作均由服务器自动激活相应的触发器，触发器能够在数据库管理系统核心层进行集中的完整性控制，可以实现更强的约束，实施更为复杂的检查和操作。基于对仓库管理数据库关系模式的分析，设计触发器如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 触发器名 | 涉及的表 | 类型 | 作用 |
| Books\_exceed | BookTb1, BQty | before | 书籍出库超过库存，出库数量改为库存数量 |
| Books\_update\_Pro | BookTb1, BQty | after | 书籍出库后修改书籍库存数量 |
| Books\_update\_Pro | BookTb1, BQty | after | 书籍入库时修改书籍库存数量 |
| Books | BookTb1 | before | 增加书籍信息，库存数量不等于0时修改为0 |

# 

# 5.数据库物理设计

数据的物理设计是为一个给定的逻辑数据模型选取一个适合的物理结构的过程，主要任务为确定存取方法和存储结构。下图为电脑书店信息管理数据库的类图。

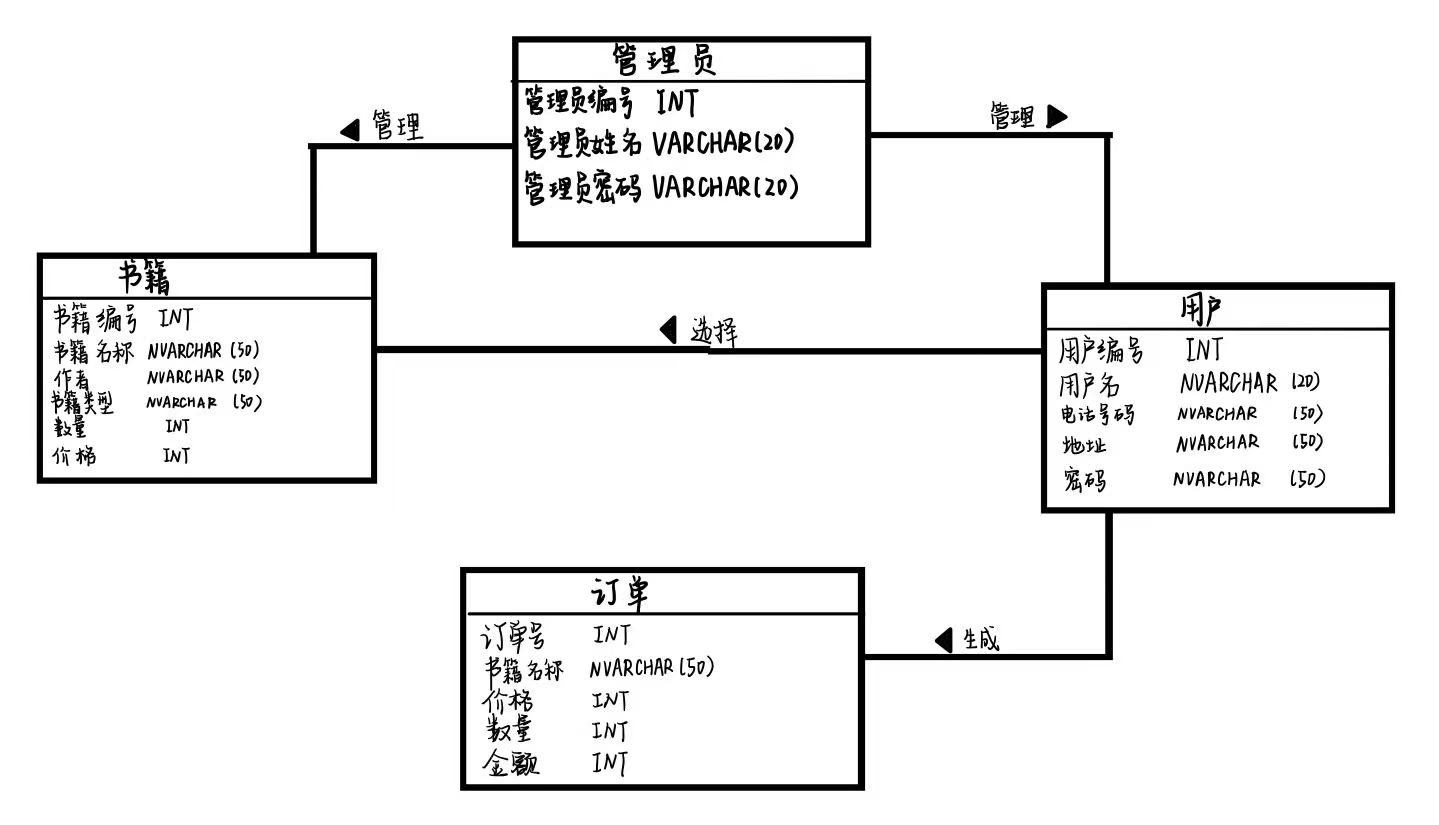


图4 电脑书店信息管理系统类图

## 5.1选择建立索引；

存取方法是快速存取数据库中的数据的技术，数据库管理系统一般提供多种存取方法，常用的方法有索引方法和聚簇方法，依据对仓库管理信息系统的分析，建立索引如下。

1. 在BookTb1表上建立BId字段（主键）建立索引

由于每一条出库/入库记录的插入都需要更新对应货物的BookTb1.BId（书籍编号），故考虑在该属性上建立索引。

|  |
| --- |
| alter table BookTb1 add index BookTb1\_INDEX (BOOKNO); |

1. 在BilTb1表上建立BillId字段（主键）建立索引

由于BilTb1.BillId在查询订单详细信息时需要建立订单表BilTb1和书籍表BookTb1的连接，故有必要在BilTb1.BillId属性上建立索引，以方便查询。

|  |
| --- |
| alter table BilTb1 add index BilTb1\_INDEX (BILLNO); |

同样地，对订单详细信息进行查询时也需要建立订单表BilTb1和用户表UserTb1的连接故有必要在UserTb1.UId属性 上建立索引，以提高查询效率。

|  |
| --- |
| alter table UserTb1 add index UserTb1\_INDEX (UNO); |

# 

# 6.数据库实施

## 6.1创建数据库及数据库对象

### 6.1.1 数据库创建

创建数据库BookShopDb。

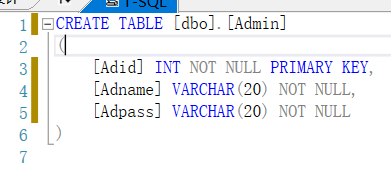


创建成功

### 6.1.2 基本表创建

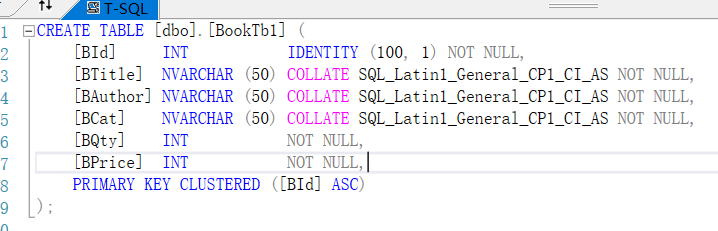
1. Table: Admin

创建表Admin



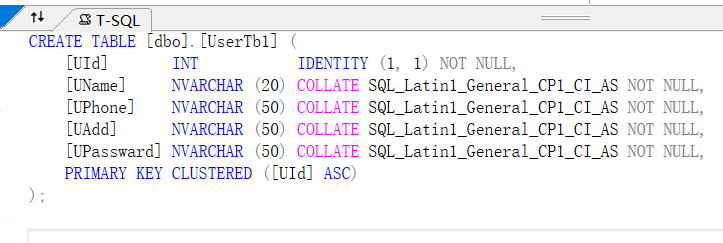
1. Table: BookTb1

创建表BookTb1。



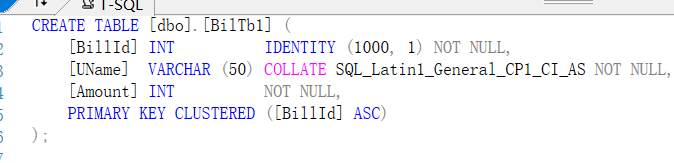
1. Table: UserTb1

创建表UserTb1



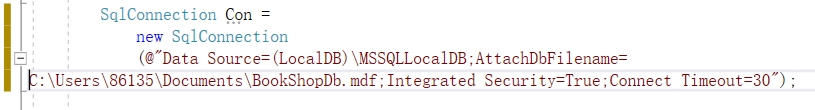
1. Table: BilTb1

创建表BilTb1。



## 6.1.3应用程序

1. 连接数据库



1. 加载界面类

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace bookstore\_management

{

public partial class bookstore : Form

{

public bookstore()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

int startpos = 0;

private void timer1\_Tick(object sender, EventArgs e)

{

startpos += 1;

Myprogress.Value = startpos;

PercentageLbI.Text = startpos+"%";

if(Myprogress.Value==100)

{

Myprogress.Value = 0;

timer1.Stop();

Login log =new Login();

log.Show();

this.Hide();

}

}

private void bookstore\_Load(object sender, EventArgs e)

{

timer1.Start();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

}

}

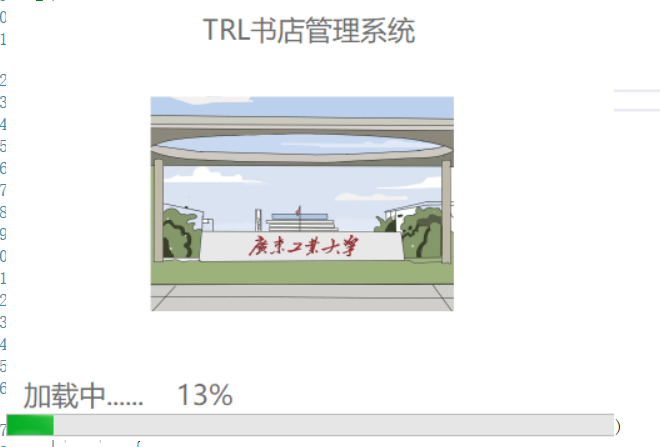


图5 电脑书店信息管理系统加载界面

1. 用户登陆界面类

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace bookstore\_management

{

public partial class Login : Form

{

public Login()

{

InitializeComponent();

}

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\86135\Documents\BookShopDb.mdf;Integrated Security=True;Connect Timeout=30");

private void label4\_Click(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

Application.Exit();//关闭窗口

}

public static string UserName = "";

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

int key = 0;

private void 登录\_Click(object sender, EventArgs e)

{

Con.Open();

SqlDataAdapter sda = new SqlDataAdapter("select count(\*) from UserTb1 where UName='"+UNameTb.Text+"'and UPassward = '"+UPassTb.Text+"'",Con);

DataTable dt = new DataTable();

sda.Fill(dt);

if(dt.Rows[0][0].ToString() == "1")

{

UserName = UNameTb.Text;

Billing obj = new Billing();

obj.Show();

this.Hide();

Con.Close();

}

else

{

MessageBox.Show("用户名和密码输入错误！！！");

}

Con.Close();

}

private void UPassTb\_TextChanged(object sender, EventArgs e)

{

}

private void Login\_Load(object sender, EventArgs e)

{

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

AdminLogin obj=new AdminLogin();

obj.Show();

this.Hide();

}

}

}

用户名：lrt 密码：123



图5 电脑书店信息管理系统用户登录界面

1. 管理员登录界面类

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace bookstore\_management

{

public partial class AdminLogin : Form

{

public AdminLogin()

{

InitializeComponent();

}

private void label7\_Click(object sender, EventArgs e)

{

Login obj = new Login();

obj.Show();

this.Hide();

}

private void 登录\_Click(object sender, EventArgs e)

{

if(UPassTb.Text =="123456")

{

books obj = new books();

obj.Show();

this.Hide();

}

else

{

MessageBox.Show("密码错误！！！");

}

}

private void label3\_Click(object sender, EventArgs e)

{

Login obj = new Login();

obj.Show();

this.Hide();

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

}

}

管理员登陆密码123456



图6 电脑书店信息管理系统管理员登录界面

1. 用户购书前端界面实现

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace bookstore\_management

{

public partial class Billing : Form

{

public Billing()

{

InitializeComponent();

populate();

}

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\86135\Documents\BookShopDb.mdf;Integrated Security=True;Connect Timeout=30");

private void populate()

{

Con.Open();

string query = "select \* from BookTb1";

SqlDataAdapter sda = new SqlDataAdapter(query, Con);

SqlCommandBuilder builder = new SqlCommandBuilder(sda);

var ds = new DataSet();

sda.Fill(ds);

BookDGV.DataSource = ds.Tables[0];

Con.Close();

}

private void UpdateBook()

{

int newQty=stock -Convert.ToInt32(QtyTb.Text);

try

{

Con.Open();

string query = "update BookTb1 set BQty=" + newQty + " where BId = " + key + " ";

SqlCommand cmd = new SqlCommand(query, Con);

cmd.ExecuteNonQuery();

MessageBox.Show("加入购物车成功！！！");

Con.Close();

populate();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

private void label12\_Click(object sender, EventArgs e)

{

}int n = 0,GraTotal=0;

private void AddtoBillBtn\_Click(object sender, EventArgs e)

{

if (QtyTb.Text == "" || Convert.ToInt32(QtyTb.Text) >stock)

{

MessageBox.Show("库存不足！！！");

}

else

{

int total=Convert.ToInt32(QtyTb.Text)\* Convert.ToInt32(PriceTb.Text);

DataGridViewRow newRow = new DataGridViewRow();

newRow.CreateCells(BillDGV);

newRow.Cells[0].Value = n+1;

newRow.Cells[1].Value = BTitleTb.Text;

newRow.Cells[2].Value = PriceTb.Text;

newRow.Cells[3].Value= QtyTb.Text;

newRow.Cells[4].Value =total;

BillDGV.Rows.Add(newRow);

n++;

UpdateBook();

GraTotal = GraTotal + total;

TotalLbl.Text = GraTotal+"元";

}

}

int key = 0,stock=0;

private void BookDGV\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

BTitleTb.Text = BookDGV.SelectedRows[0].Cells[1].Value.ToString();

//BautTb.Text = BookDGV.SelectedRows[0].Cells[2].Value.ToString();

//BCatCb.SelectedItem = BookDGV.SelectedRows[0].Cells[3].Value.ToString();

//QtyTb.Text = BookDGV.SelectedRows[0].Cells[4].Value.ToString();

PriceTb.Text = BookDGV.SelectedRows[0].Cells[5].Value.ToString();

QtyTb.Text = "";

if (BTitleTb.Text == "")

{

key = 0;

stock = 0;

}

else

{

key = Convert.ToInt32(BookDGV.SelectedRows[0].Cells[0].Value.ToString());

stock= Convert.ToInt32(BookDGV.SelectedRows[0].Cells[4].Value.ToString());

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void BautTb\_TextChanged(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void EditBtn\_Click(object sender, EventArgs e)

{

if (BillDGV.Rows[0].Cells[0].Value==null)

{

MessageBox.Show("您还没有选择书籍~~~");

}

else

{

try

{

Con.Open();

string query = "insert into BilTb1 values('" + UserNameLbl.Text + "'," + GraTotal+ ")";

SqlCommand cmd = new SqlCommand(query, Con);

cmd.ExecuteNonQuery();

MessageBox.Show("订单信息保存成功！！！");

Con.Close();

//populate();

//Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

printDocument1.DefaultPageSettings.PaperSize = new System.Drawing.Printing.PaperSize("pprnm", 285, 600);

if (printPreviewDialog1.ShowDialog() == DialogResult.OK)

{

printDocument1.Print();

}

}

}

int prodid, prodqty, prodprice, tottal, pos = 60;

private void label7\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void label11\_Click(object sender, EventArgs e)

{

Login obj = new Login();

obj.Show();

this.Hide();

}

string prodname;

private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawString("TRL书店", new Font("微软雅黑", 12, FontStyle.Bold), Brushes.Red, new Point(80));

e.Graphics.DrawString("编号 产品 价格 数量 总计", new Font("微软雅黑", 10, FontStyle.Bold), Brushes.Red, new Point(26, 40));

foreach(DataGridViewRow row in BillDGV.Rows)

{

prodid = Convert.ToInt32(row.Cells["Column7"].Value);

prodname=""+row.Cells["Column8"].Value;

prodprice=Convert.ToInt32(row.Cells["Column9"].Value);

prodqty = Convert.ToInt32(row.Cells["Column10"].Value);

tottal= Convert.ToInt32(row.Cells["Column11"].Value);

e.Graphics.DrawString("" + prodid, new Font("微软雅黑", 8, FontStyle.Bold), Brushes.Blue, new Point(26, pos));

e.Graphics.DrawString("" + prodname, new Font("微软雅黑", 8, FontStyle.Bold), Brushes.Blue, new Point(45, pos));

e.Graphics.DrawString("" + prodprice, new Font("微软雅黑", 8, FontStyle.Bold), Brushes.Blue, new Point(120, pos));

e.Graphics.DrawString("" + prodqty, new Font("微软雅黑", 8, FontStyle.Bold), Brushes.Blue, new Point(170, pos));

e.Graphics.DrawString("" + tottal, new Font("微软雅黑", 8, FontStyle.Bold), Brushes.Blue, new Point(235, pos));

pos = pos + 20;

}

e.Graphics.DrawString("订单总额：￥" + GraTotal, new Font("微软雅黑",12,FontStyle.Bold),Brushes.Crimson,new Point(60,pos+50));

e.Graphics.DrawString("\*\*\*\*\*\*TRL书店\*\*\*\*\*\*" , new Font("微软雅黑", 12, FontStyle.Bold), Brushes.Crimson, new Point(40, pos + 85));

BillDGV.Rows.Clear();//小票打印出来后，清空购物车

BillDGV.Refresh();

pos = 100;

GraTotal = 0;

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

UserNameLbl.Text = Login.UserName;

}

private void Reset()

{

BTitleTb.Text = "";

QtyTb.Text = "";

PriceTb.Text = "";

}

private void ResetBtn\_Click(object sender, EventArgs e)

{

Reset();

}

}

}



图7 用户购书界面

1. 管理员后台书籍管理界面实现：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace bookstore\_management

{

public partial class books : Form

{

public books()

{

InitializeComponent();

populate();

}

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\86135\Documents\BookShopDb.mdf;Integrated Security=True;Connect Timeout=30");

private void populate()

{

Con.Open();

string query = "select \* from BookTb1";

SqlDataAdapter sda = new SqlDataAdapter(query,Con);

SqlCommandBuilder builder = new SqlCommandBuilder(sda);

var ds = new DataSet();

sda.Fill(ds);

BookDGV.DataSource=ds.Tables[0];

Con.Close();

}

private void Filter()

{

Con.Open();

string query = "select \* from BookTb1 where BCat = '"+CatCbSearchCb.SelectedItem.ToString()+"' ";

SqlDataAdapter sda = new SqlDataAdapter(query, Con);

SqlCommandBuilder builder = new SqlCommandBuilder(sda);

var ds = new DataSet();

sda.Fill(ds);

BookDGV.DataSource = ds.Tables[0];

Con.Close();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void label9\_Click(object sender, EventArgs e)

{

account obj = new account();

obj.Show();

this.Hide();

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void books\_Load(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void SaveBtn\_Click(object sender, EventArgs e)

{

if(BTitleTb.Text==""||BautTb.Text==""||QtyTb.Text==""||PriceTb.Text==""||BCatCb.SelectedIndex==-1)

{

MessageBox.Show("信息缺失！！！");

}

else

{

try

{

Con.Open();

string query="insert into BookTb1 values('"+BTitleTb.Text+"','"+BautTb.Text+"','"+BCatCb.SelectedItem.ToString()+"',"+QtyTb.Text+","+PriceTb.Text+")";

SqlCommand cmd = new SqlCommand(query,Con);

cmd.ExecuteNonQuery();

MessageBox.Show("书籍信息保存成功！！！");

Con.Close();

populate();

Reset();

}catch(Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

int key = 0;

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

BTitleTb.Text=BookDGV.SelectedRows[0].Cells[1].Value.ToString();

BautTb.Text = BookDGV.SelectedRows[0].Cells[2].Value.ToString();

BCatCb.SelectedItem = BookDGV.SelectedRows[0].Cells[3].Value.ToString();

QtyTb.Text = BookDGV.SelectedRows[0].Cells[4].Value.ToString();

PriceTb.Text = BookDGV.SelectedRows[0].Cells[5].Value.ToString();

if(BTitleTb.Text =="")

{

key = 0;

}

else

{

key = Convert.ToInt32(BookDGV.SelectedRows[0].Cells[0].Value.ToString());

}

}

private void CatCbSearchCb\_SelectionChangeCommitted(object sender, EventArgs e)

{

Filter();

}

private void button4\_Click(object sender, EventArgs e)

{

populate();

CatCbSearchCb.SelectedIndex = -1;

}

private void Reset()

{

BTitleTb.Text = "";

BautTb.Text = "";

BCatCb.SelectedIndex = -1;

QtyTb.Text = "";

PriceTb.Text = "";

}

private void ResetBtn\_Click(object sender, EventArgs e)

{

Reset();

}

private void DeleteBtn\_Click(object sender, EventArgs e)

{

if (key==0)

{

MessageBox.Show("信息缺失！！！");

}

else

{

try

{

Con.Open();

string query = "delete from BookTb1 where BId = "+ key + "";

SqlCommand cmd = new SqlCommand(query, Con);

cmd.ExecuteNonQuery();

MessageBox.Show("书籍信息删除成功！！！");

Con.Close();

populate();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

private void EditBtn\_Click(object sender, EventArgs e)

{

if (BTitleTb.Text == "" || BautTb.Text == "" || QtyTb.Text == "" || PriceTb.Text == "" || BCatCb.SelectedIndex == -1)

{

MessageBox.Show("信息缺失！！！");

}

else

{

try

{

Con.Open();

string query = "update BookTb1 set BTitle='"+BTitleTb.Text+"',BAuthor='"+BautTb.Text+"',BCat='"+BCatCb.SelectedItem.ToString()+"',BQty="+QtyTb.Text+",BPrice="+PriceTb.Text+"where BId = "+key+" ";

SqlCommand cmd = new SqlCommand(query, Con);

cmd.ExecuteNonQuery();

MessageBox.Show("书籍信息更新完成！！！");

Con.Close();

populate();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

private void panel6\_Paint(object sender, PaintEventArgs e)

{

}

private void label11\_Click(object sender, EventArgs e)

{

Login obj = new Login();

obj.Show();

this.Hide();

}

private void panel5\_Paint(object sender, PaintEventArgs e)

{

}

private void panel4\_Paint(object sender, PaintEventArgs e)

{

}

private void label8\_Click(object sender, EventArgs e)

{

users obj = new users();

obj.Show();

this.Hide();

}

private void panel3\_Paint(object sender, PaintEventArgs e)

{

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void label10\_Click(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void CatCbSearchCb\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void label14\_Click(object sender, EventArgs e)

{

}

private void label13\_Click(object sender, EventArgs e)

{

}

private void PriceTb\_TextChanged(object sender, EventArgs e)

{

}

private void label12\_Click(object sender, EventArgs e)

{

}

private void QtyTb\_TextChanged(object sender, EventArgs e)

{

}

private void BCatCb\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void BautTb\_TextChanged(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void pictureBox6\_Click(object sender, EventArgs e)

{

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

}

}

}

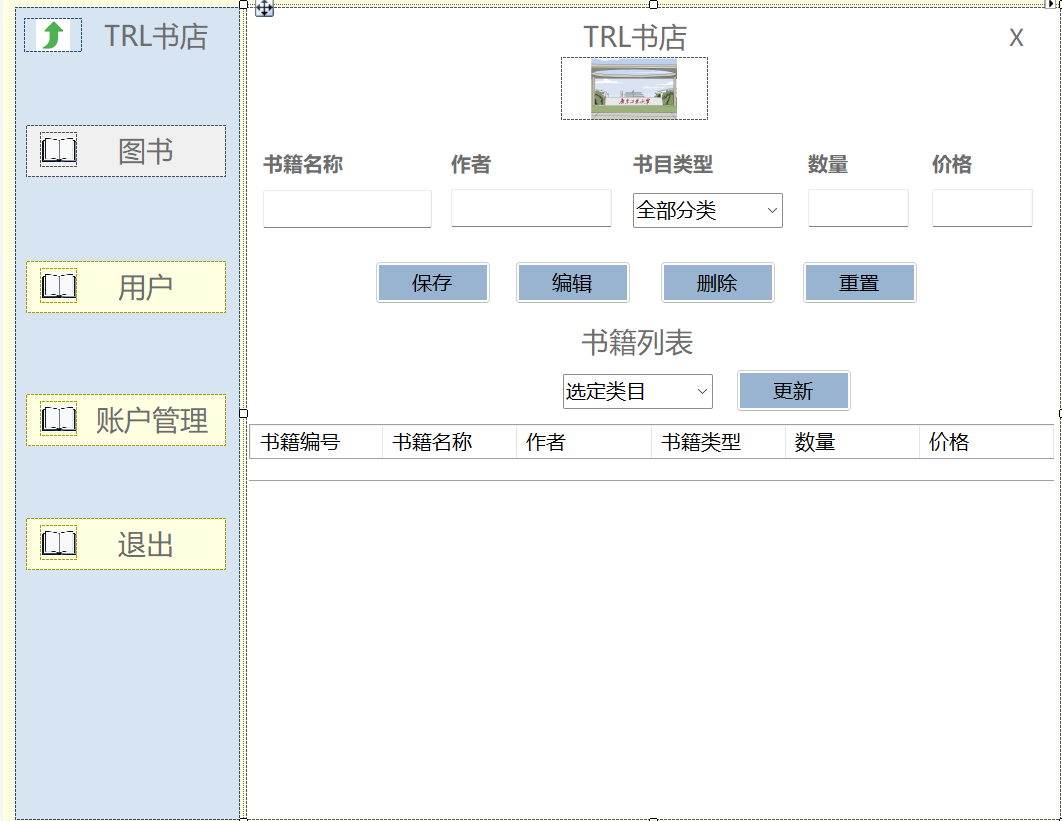


图8 管理员书籍管理界面

1. 管理员后台用户管理界面实现：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace bookstore\_management

{

public partial class users : Form

{

public users()

{

InitializeComponent();

populate();

}

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\86135\Documents\BookShopDb.mdf;Integrated Security=True;Connect Timeout=30");

private void label14\_Click(object sender, EventArgs e)

{

}

private void populate()

{

Con.Open();

string query = "select \* from UserTb1";

SqlDataAdapter sda = new SqlDataAdapter(query, Con);

SqlCommandBuilder builder = new SqlCommandBuilder(sda);

var ds = new DataSet();

sda.Fill(ds);

UserDGV.DataSource = ds.Tables[0];

Con.Close();

}

private void SaveBtn\_Click(object sender, EventArgs e)

{

if (UnameTb.Text == "" || PhoneTb.Text == "" || AddTb.Text == "" || PassTb.Text == "")

{

MessageBox.Show("信息缺失！！！");

}

else

{

try

{

Con.Open();

string query = "insert into UserTb1 values('" + UnameTb.Text + "','" + PhoneTb.Text + "','" + AddTb.Text + "'," + PassTb.Text + ")";

SqlCommand cmd = new SqlCommand(query, Con);

cmd.ExecuteNonQuery();

MessageBox.Show("用户信息保存成功！！！");

Con.Close();

populate();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

int key = 0;

private void UserDVG\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

UnameTb.Text = UserDGV.SelectedRows[0].Cells[1].Value.ToString();

PhoneTb.Text = UserDGV.SelectedRows[0].Cells[2].Value.ToString();

AddTb.Text = UserDGV.SelectedRows[0].Cells[3].Value.ToString();

PassTb.Text = UserDGV.SelectedRows[0].Cells[4].Value.ToString();

if (UnameTb.Text == "")

{

key = 0;

}

else

{

key = Convert.ToInt32(UserDGV.SelectedRows[0].Cells[0].Value.ToString());

}

}

private void PassTb\_TextChanged(object sender, EventArgs e)

{

}

private void Reset()

{

UnameTb.Text = "";

PhoneTb.Text = "";

AddTb.Text = "";

PassTb.Text = "";

}

private void ResetBtn\_Click(object sender, EventArgs e)

{

Reset();

}

private void DeleteBtn\_Click(object sender, EventArgs e)

{

if (key == 0)

{

MessageBox.Show("信息缺失！！！");

}

else

{

try

{

Con.Open();

string query = "delete from UserTb1 where UId = " + key + "";

SqlCommand cmd = new SqlCommand(query, Con);

cmd.ExecuteNonQuery();

MessageBox.Show("用户信息删除成功！！！");

Con.Close();

populate();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

private void EditBtn\_Click(object sender, EventArgs e)

{

if (UnameTb.Text == "" || PhoneTb.Text == "" || AddTb.Text == "" || PassTb.Text == "")

{

MessageBox.Show("信息缺失！！！");

}

else

{

try

{

Con.Open();

string query = "update userTb1 set UName='" + UnameTb.Text + "',UPhone='" + PhoneTb.Text + "',UAdd='" + AddTb.Text + "',UPass=" + PassTb.Text + ",BPrice=" + PassTb.Text + "where UId = " + key + " ";

SqlCommand cmd = new SqlCommand(query, Con);

cmd.ExecuteNonQuery();

MessageBox.Show("用户信息更新完成！！！");

Con.Close();

populate();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

private void label11\_Click(object sender, EventArgs e)

{

Login obj = new Login();

obj.Show();

this.Hide();

}

private void label6\_Click(object sender, EventArgs e)

{

books obj = new books();

obj.Show();

this.Hide();

}

private void label8\_Click(object sender, EventArgs e)

{

}

private void label9\_Click(object sender, EventArgs e)

{

account obj = new account();

obj.Show();

this.Hide();

}

private void label7\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

}

}

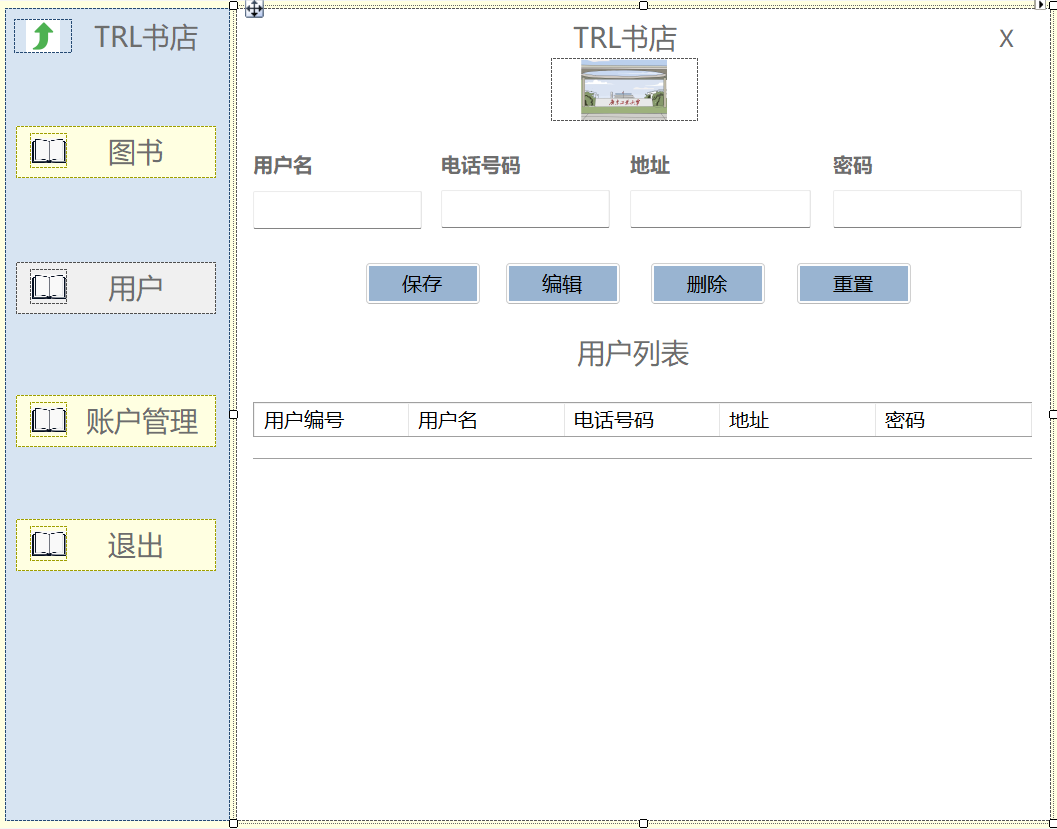


图9 管理员用户管理界面

1. 管理员后台账户管理界面实现：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace bookstore\_management

{

public partial class account : Form

{

public account()

{

InitializeComponent();

}

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\86135\Documents\BookShopDb.mdf;Integrated Security=True;Connect Timeout=30");

private void pictureBox7\_Click(object sender, EventArgs e)

{

}

private void panel7\_Paint(object sender, PaintEventArgs e)

{

}

private void label11\_Click(object sender, EventArgs e)

{

Login obj = new Login();

obj.Show();

this.Hide();

}

private void label8\_Click(object sender, EventArgs e)

{

users obj = new users();

obj.Show();

this.Hide();

}

private void label6\_Click(object sender, EventArgs e)

{

books obj = new books();

obj.Show();

this.Hide();

}

private void label9\_Click(object sender, EventArgs e)

{

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

private void account\_Load(object sender, EventArgs e)

{

Con.Open();

SqlDataAdapter sda = new SqlDataAdapter("select sum(BQty) from BookTb1", Con);

DataTable dt = new DataTable();

sda.Fill(dt);

BookStockLbI.Text = dt.Rows[0][0].ToString();

SqlDataAdapter sda1 = new SqlDataAdapter("select sum(Amount) from BilTb1", Con);

DataTable dt1 = new DataTable();

sda1.Fill(dt1);

AmountLbI.Text = dt1.Rows[0][0].ToString();

SqlDataAdapter sda2 = new SqlDataAdapter("select Count(\*) from UserTb1", Con);

DataTable dt2 = new DataTable();

sda2.Fill(dt2);

UserTotalLbI.Text = dt2.Rows[0][0].ToString();

Con.Close();

}

private void AmountLbI\_Click(object sender, EventArgs e)

{

}

private void UserTotalLbI\_Click(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

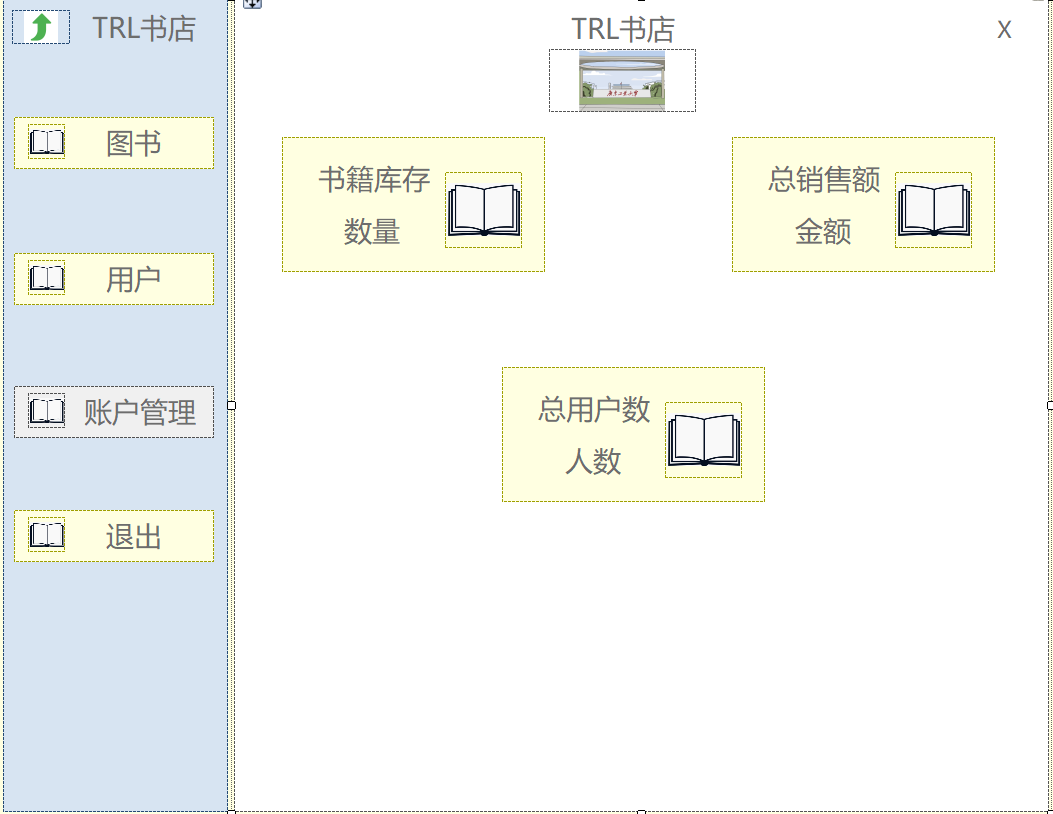


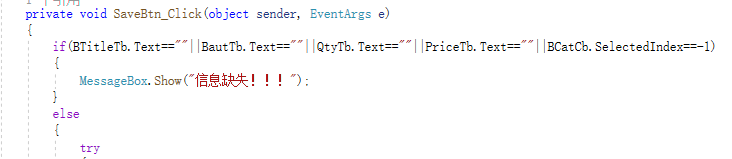
图10 管理员账户管理界面

# 7.系统测试方案和测试报告

## 7.1系统测试

书籍信息表：BookTb1

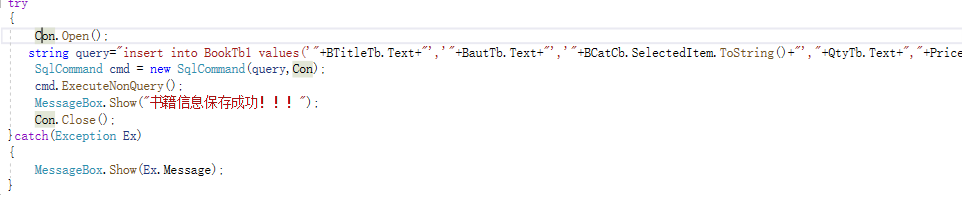
（1）插入书籍信息缺少以上书籍任一信息时显示：信息缺失！！！



测试：

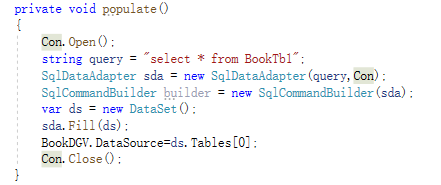


（2）往BookTb1表中插入书籍的所有信息，并显示书籍信息保存成功！！！



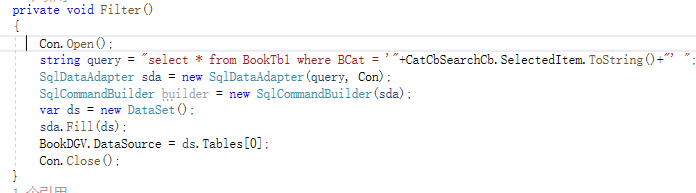


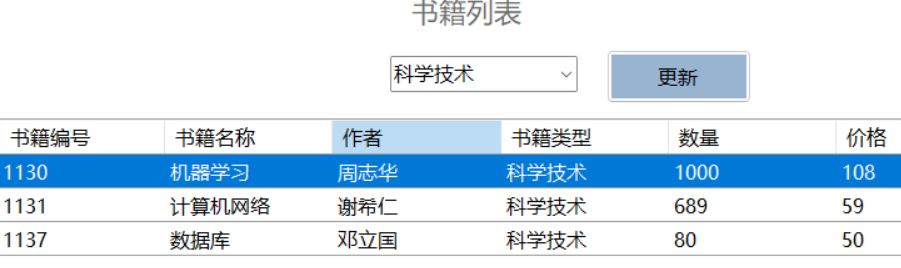
（3）在界面显示BookTb1表，并查询所有书籍的信息



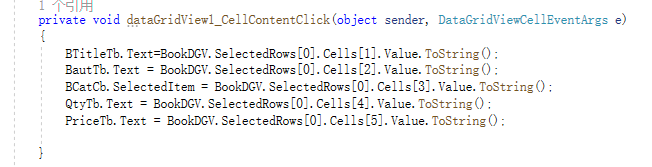


（4）添加查询书籍类目功能





(5)添加“点击书籍列表时，搜索框可以显示书籍信息”的功能

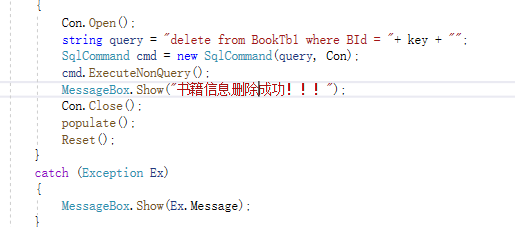




(6)对系统添加删除书籍功能

选定要删除的书籍进行书籍删除

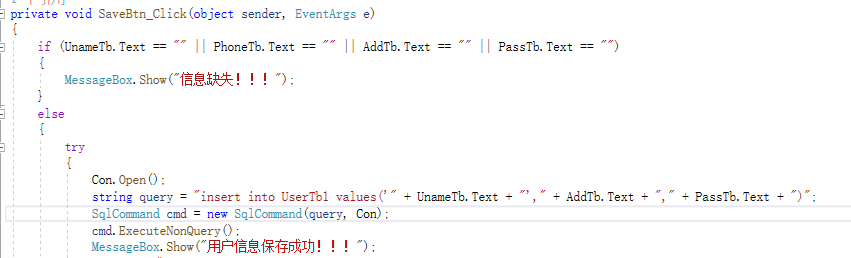
删除成功时，系统界面显示“书籍信息删除成功！！！”





用户信息UserTb1表

（7）添加用户信息，若添加的用户信息为空，则添加失败，显示“信息缺失”，反之显示“用户信息保存成功”



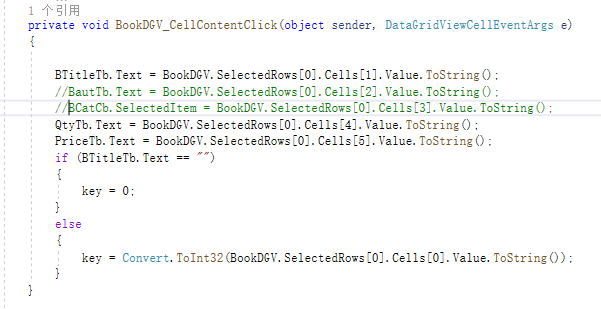




用户书籍购买功能实现：

（8）添加书籍商品选定功能

选择可购买的书籍，显示在搜索框中

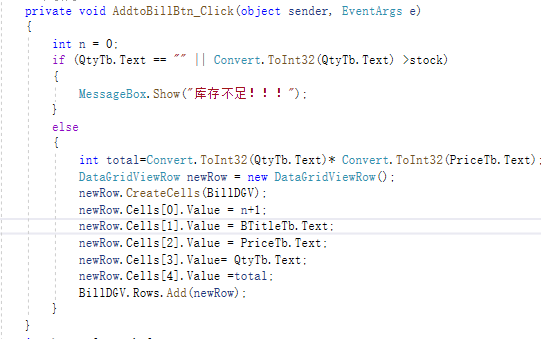




（9）用户添加书籍加入购物车功能实现：

用户选择书籍、数量加入购物车

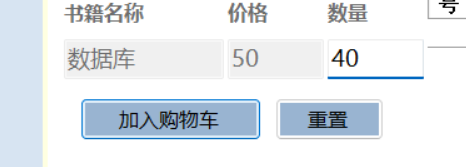
当用户没有选择购买数量或用户选择的购买数量超出库存时，显示“库存不足”



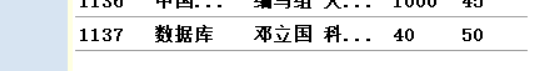


（10）添加更新书籍库存功能





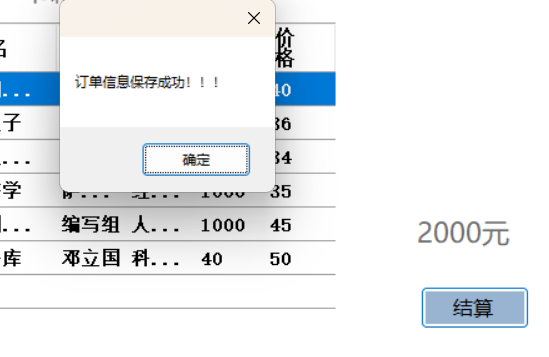
用户购买40本书



库存由80变为40

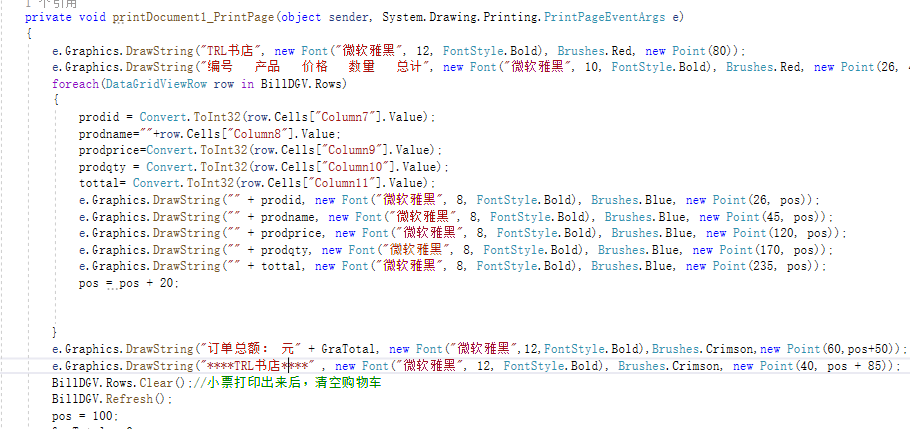
(11)添加显示当前订单消费总额功能





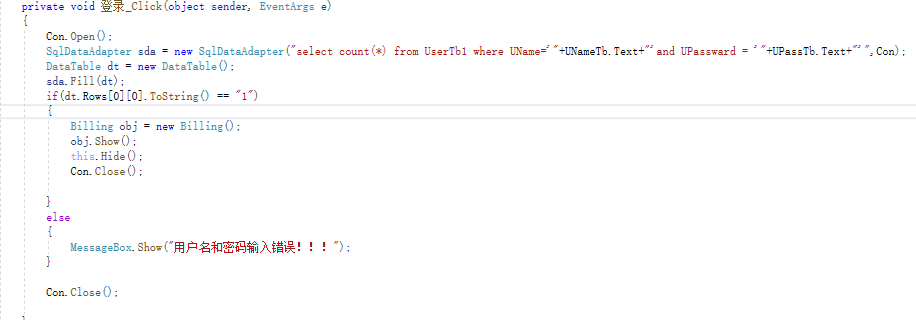
(12)添加订单结算功能

打印订单功能，打印订单后清空购物车





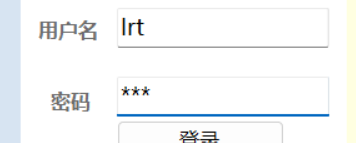
(13)用户登录界面功能实现：



用户名和密码输入错误时：



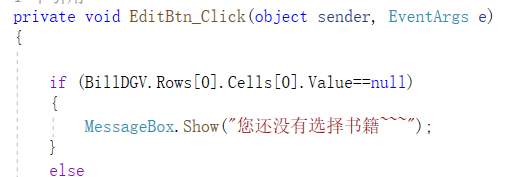
输入正确用户名、密码时登陆成功

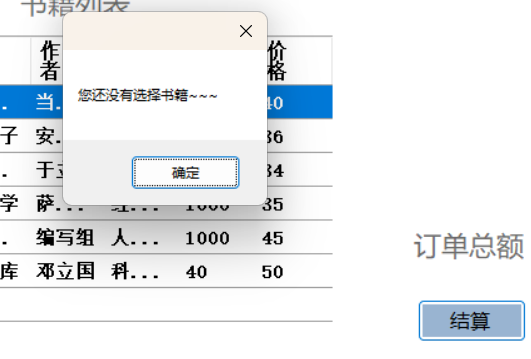




（14）订单结算信功能实现：

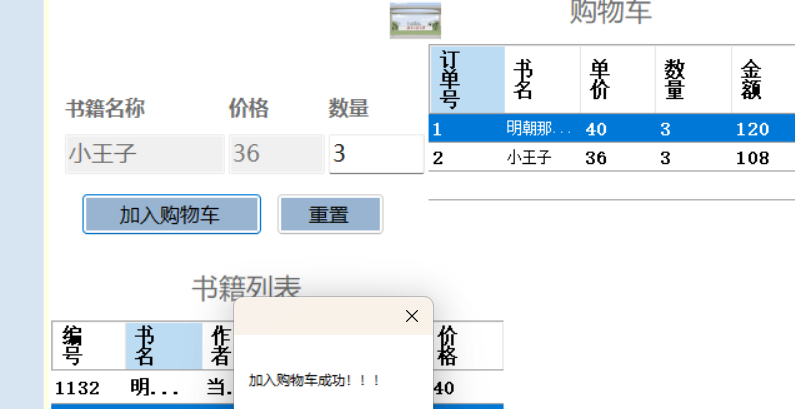
当用户购物车为空时，无法购买，界面显示“您还没有选择书籍”





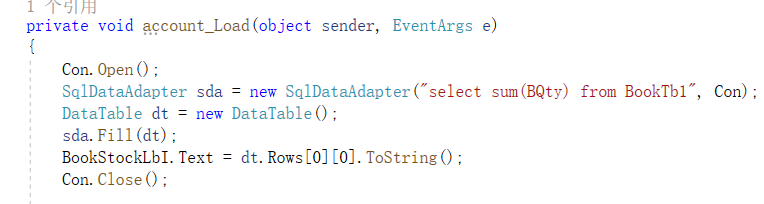
（15）用户选择要购买的书的标题、种类、数目加入购物车





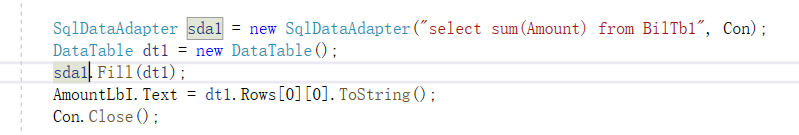
（16）用户账户管理界面的实现

一、账户管理界面实现显示书籍库存数量





账户管理界面实现显示书店总销售额





账户管理界面实现显示用户总数量

