**《数据库系统原理》课程设计报告**

**设计名称：** 网上书店管理系统

**学生学号：**

**学生姓名：**

**专业班级：**

**设计地点：**

**设计时间：** 2022.6.27-2022.7.15

|  |
| --- |
| 指导教师评价：  **成绩：**  1. 创新性 【好】【较好】【一般】【无】  2. 写作情况 【好】【较好】【一般】【差】  3. 需求分析 【好】【较好】【一般】【差】  4. 设计合理性【好】【较好】【一般】【差】  5. 系统实现 【好】【较好】【一般】【差】  6. 总结 【好】【较好】【一般】【差】  7. 演示情况 【好】【较好】【一般】【差】 |

**《数据库系统原理》课程设计**

**1　设计目的**

《数据库系统原理》课程设计是软件工程和计算机科学与技术专业集中实践性环节之一，旨在让同学们加深对数据库基础理论和基本知识的理解，掌握设计数据库管理系统的基本方法，锻炼运用知识解决实际问题的动手能力。

**2　任务与要求**

要求学生们从给定的设计题目中进行选择，进行需求分析，概念设计、逻辑设计，数据库的设计过程，表结构、表之间的关联，给出视图的定义、触发器的定义、索引，安全性的实现，用SQL语句等在SQL Server 2000/2008/2012, 或者OpenGauss等系统中实现数据库的数据输入，查询，更新和输出；给出实现效果截图及部分测试结果。

**3** **网上书店管理系统课程设计**

**3.1 引言**

本课程设计旨在开发一个网上书店管理系统，以满足现代书店的需求和客户的期望。随着互联网的普及，网上书店已成为图书销售的重要渠道之一。这个系统将帮助书店管理图书库存、查看客户订单、提供书籍信息、并确保数据的安全性。通过本设计，我们将解决以下问题：

提高图书销售效率：传统的书店可能需要大量人力来处理订单和提供客户服务。通过引入网上书店管理系统，可以更高效地处理这些任务，从而提高销售效率。

提供便捷的购书体验：通过在线平台，客户可以随时随地浏览和购买图书，无需前往实际书店。这提供了便捷的购书体验，吸引更多客户。

数据管理和安全性：系统将帮助书店管理图书信息、客户信息和订单数据。同时，它也需要确保数据的安全性，以防止未经授权的访问或数据泄露。

提供实时信息：通过该系统，书店可以提供实时的图书信息，包括库存状态、价格和作者等，以帮助客户做出购书决策。

订单处理：客户可以轻松下订单，系统将自动处理订单信息，包括支付信息，从而简化订单流程。

通过本课程设计，我们将建立一个网上书店管理系统，可以满足现代书店的需求，提高销售效率，提供更好的客户体验，同时确保数据的安全性和准确性。

**3.2 需求分析**

**3.2.1 系统需求调查**

用户访谈：我们与潜在用户、书店管理员和客户进行了面对面或在线访谈，以了解他们的需求和期望。

市场研究：我们进行了市场调查，分析现有的网上书店系统，以了解行业标准和最佳实践。

现有系统分析：如果书店已经使用过一些系统，我们会对其进行分析，以确定哪些功能需要保留，哪些需要改进。

需求文档：我们可以收集书店管理系统的需求文档，以确定基本要求。

系统主要任务： 网上书店管理系统的主要任务是提供一个在线平台，供客户浏览、搜索、购买图书，同时帮助书店管理图书、订单和客户信息。系统还需要确保数据的安全性和完整性。

数据库完整性要求：数据库必须维护图书信息的完整性，包括作者、书名、价格等，以及客户信息和订单信息。必须有有效的外键关系来维护数据一致性。

安全性要求：用户认证和授权机制，以防止未经授权的访问。数据加密，以保护敏感信息如用户密码。定期备份，以应对数据丢失或损坏情况。

性能要求：快速的响应时间，以提供良好的用户体验；数据库查询和更新操作需要高效执行。

一致性要求：数据库中的信息必须保持一致，例如订单状态必须准确反映实际情况。

数据库要求：数据库系统应该确保数据的一致性和完整性。同时，数据库应该能够容纳大量的图书信息、客户信息和订单信息。

**3.2.2 业务流程分析**

DFD示意图如下：

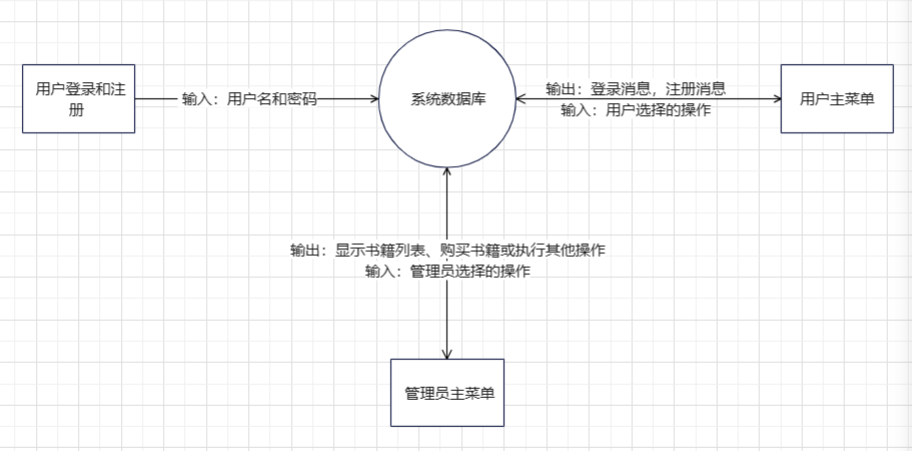


图 1 DED图

以下是完整的数据字典，以表格形式呈现，包括数据库中的表格、列名和列的描述。

用户表：

表 1 user\_info

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **描述** |
| id | INT (主键) | 用户ID |
| username | VARCHAR(255) | 用户名 |
| password | VARCHAR(255) | 密码 |

权限表：

表 2 user\_permissions

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **描述** |
| id | INT (主键) | 权限ID |
| user\_id | INT (外键，user\_info表) | 用户ID |
| is\_admin | TINYINT(1) 默认0 | 是否管理员 (0 或 1) |

书籍表：

表 3 books

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **描述** |
| id | INT (主键) | 书籍ID |
| title | VARCHAR(255) | 书籍标题 |
| author | VARCHAR(255) | 作者 |
| price | DECIMAL(10, 2) | 价格 |

订单表：

表 4 orders

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **描述** |
| id | INT (主键) | 订单ID |
| user\_id | INT (外键，user\_info 表) | 用户ID |
| book\_id | INT (外键，books 表) | 书籍ID |
| order\_date | TIMESTAMP 默认当前时间 | 订单时间 |

**3.2.3 系统功能描述**

（1）用户功能

注册和登录：用户可以通过注册新账户来创建自己的个人账户。注册流程通常包括提供必要的个人信息和设置登录密码。一旦注册成功，用户可以使用所创建的账户来登录系统，以便访问更多功能和个人化的体验。

浏览图书：用户可以通过浏览图书列表来获取关于图书的概览信息。这些信息通常包括图书的标题、作者、出版日期、价格等。用户可以浏览整个图书库或按照特定的分类进行筛选，以便找到感兴趣的图书。

搜索图书：为了方便用户快速找到特定的图书，系统提供了搜索功能。用户可以通过输入关键字或者特定的图书属性来搜索图书。搜索结果将显示与关键字或属性匹配的图书列表，帮助用户快速定位到所需的图书。

购买图书：用户可以在系统中选择购买一本或多本图书。他们可以将所选图书添加到购物车中，然后通过结算流程完成购买。结算流程通常包括选择支付方式和提供送货地址等步骤，以确保用户成功购买所需的图书。

查看已购图书：用户可以随时查看他们已经购买的图书列表。这样，他们可以方便地回顾自己的购买历史，查看已购图书的详细信息，并进行必要的管理操作。

修改密码：用户可以通过系统提供的功能来修改他们的登录密码。这样，他们可以保持账户的安全性，确保只有自己可以访问账户。

（2）管理员功能

添加图书：管理员可以通过系统界面添加新的图书信息。这包括输入图书的标题、作者、出版日期、价格等详细信息，并将其加入到图书库中。这样，管理员可以不断扩充图书库，以满足用户的需求。

修改图书信息：管理员可以对现有图书的信息进行修改。他们可以更新图书的标题、作者、出版日期、价格等信息，以确保图书库中的信息准确和最新。

删除图书：管理员可以从图书库中删除不再可用的图书。这可以包括已经下架、过时或其他不再提供的图书。通过删除不再需要的图书，管理员可以保持图书库的整洁和有序。

处理订单：管理员可以查看用户的订单信息，确认订单，并将图书进行发货。他们可以跟踪订单的状态，确保及时处理和交付用户的购买请求。

查看和修改用户信息：管理员可以查看所有用户的信息和权限。他们可以查看用户的个人资料、购买历史等信息，并可以进行必要的修改，如重置密码或更改权限设置。这样，管理员可以对用户账户进行管理和维护。

功能模块图：

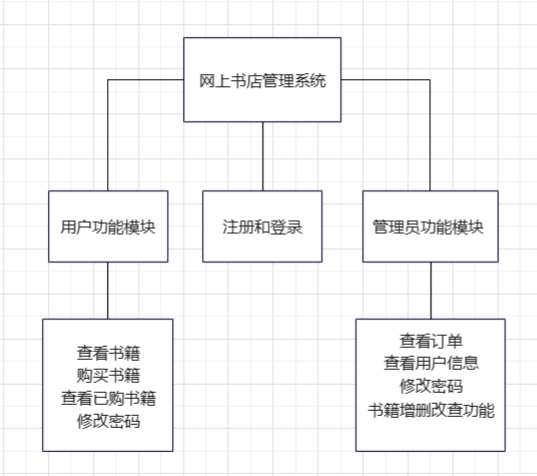


图 2 功能模块

**3.3 概念设计**

概念设计是数据库系统设计过程中的重要步骤，它的目标是识别并建立数据库中的实体、关系、属性以及它们之间的联系。在网上书店管理系统的概念设计阶段，我们将采用实体-关系图（简称 E-R 图）来表示数据库的概念模型。

**3.3.1 实体**

用户信息（user\_info）：用户ID（id，主键）、用户名（username）、密码（password）。每个用户是一个实体，具有唯一的用户ID。

用户权限（user\_permissions）：权限ID（id，主键）、用户ID（user\_id，外键参考user\_info表中的id）、是否为管理员（is\_admin）。每个用户在用户权限表中对应一条记录，标识其是否为管理员。

书籍信息（books）：书籍ID（id，主键）、标题（title）、作者（author）、价格（price）。每本书是一个实体，具有唯一的书籍ID。

订单信息（orders）：订单ID（id，主键）、用户ID（user\_id，外键参考user\_info表中的id）、书籍ID（book\_id，外键参考books表中的id）、订单日期（order\_date）。每个订单是一个实体，与用户和书籍相关联。

**3.3.2 联系**

用户与用户权限:一个用户可以对应于零个或一个用户权限记录。一个用户权限记录必须关联到一个用户。

用户与订单:一个用户可以对应于零个、一个或多个订单。一个订单必须关联到一个用户。

书籍与订单:一本书可以对应于零个、一个或多个订单。一个订单必须关联到一本书。

**3.3.3 转换规则:**

每个实体都映射到数据库中的一个表。每个属性都映射到表中的一个列。主键和外键用于定义实体之间的关系。在关联表中，外键是关联表的主键，它连接了不同表之间的关系。

E-R 图：

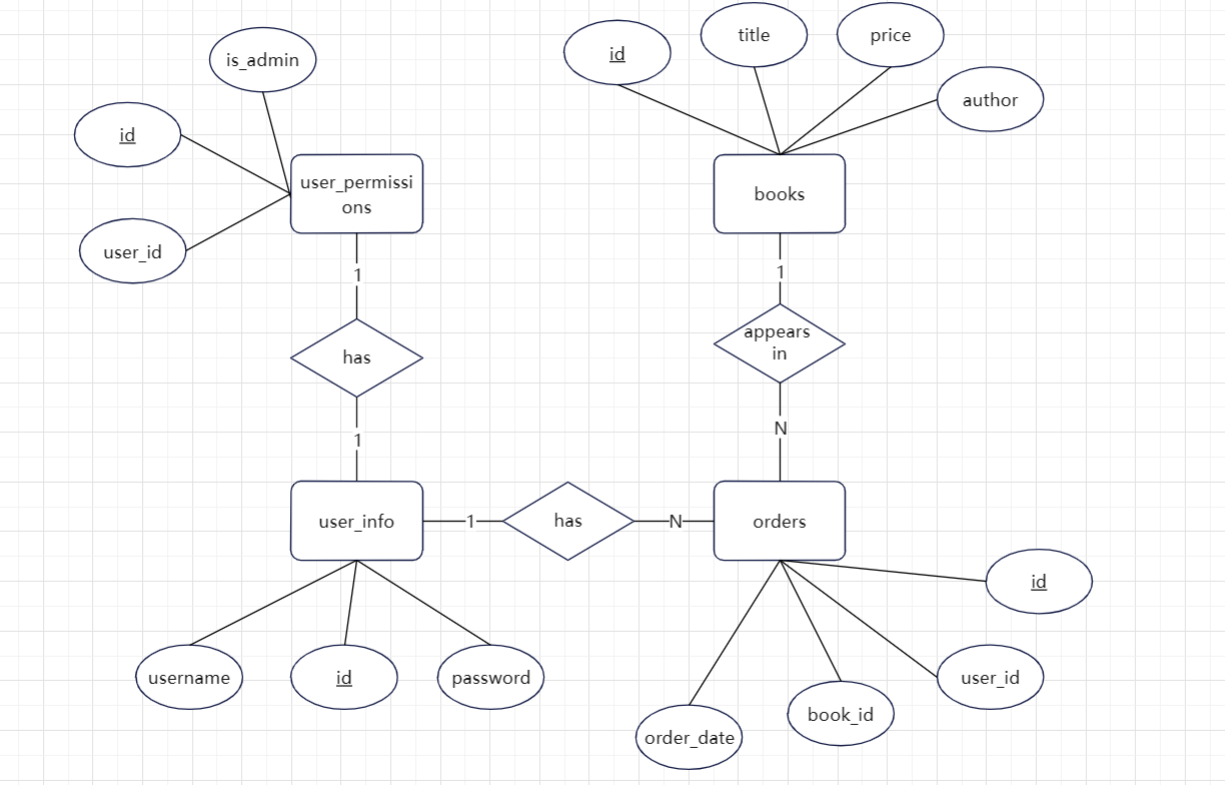


图 3 E-R图

**3.4逻辑设计**

逻辑设计是数据库系统设计过程中的关键步骤之一，其主要目标是将概念设计转化为具体的数据库关系模式（也就是数据库表结构）。在网上书店管理系统的逻辑设计阶段，将定义各个表格的结构，包括表格名称、字段名称、字段数据类型以及字段之间的关系。

E-R图转换为关系模型，如下所示：

user\_info (id、username、password) ；

user\_permissions (id、user\_id、is\_admin) ；

books (id、title、author、price)；

orders (id、user\_id、book\_id、order\_date)

数据库优化也是逻辑设计的一部分。在此阶段，我们可以考虑如何优化数据库查询性能，例如规范化表格以减少数据冗余等。这将有助于系统在面对大量数据时仍能快速响应查询和操作。

总之，逻辑设计是将概念设计转化为数据库结构的关键步骤，确保数据库能够满足系统的需求并保持良好性能。

**3.5物理结构设计**

物理结构设计是数据库设计过程中的重要一步，它决定了数据库在物理存储层面上的实际组织方式。

建表属性与sql代码：

books表：

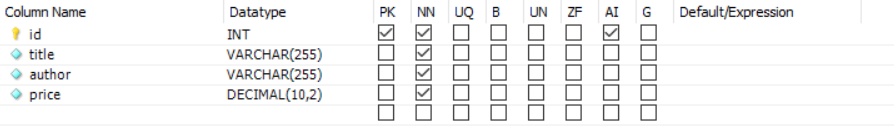


图 4 books

CREATE TABLE books (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

author VARCHAR(255) NOT NULL,

price DECIMAL(10, 2) NOT NULL

);

orders表：

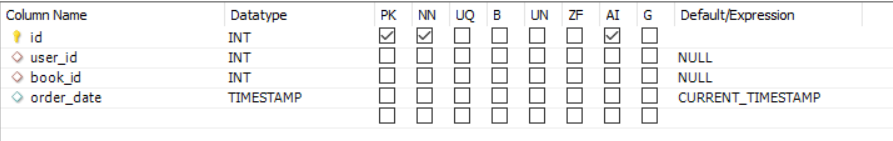


图 5 orders

CREATE TABLE orders (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

book\_id INT,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

-- TIMESTAMP：数据类型，表示这个列将存储日期和时间信息

FOREIGN KEY (user\_id) REFERENCES user\_info(id),

FOREIGN KEY (book\_id) REFERENCES books(id)

);

user\_info表：

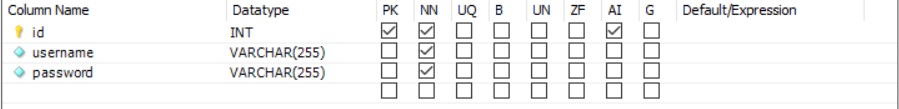


图 6 user\_info

CREATE TABLE user\_info (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL

);

user\_permissions表：

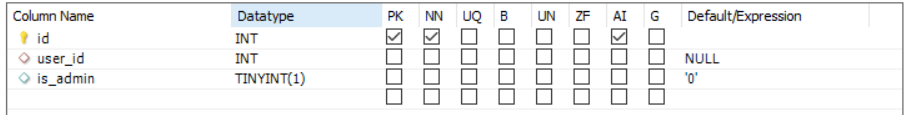


图 7 user\_permissions

CREATE TABLE user\_permissions (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

is\_admin TINYINT(1) DEFAULT 0,

-- 该列存储名为TINYINT的小整数（通常用于表示布尔值）

-- DEFAULT 0指示如果没有指定值，将使用默认值0

FOREIGN KEY (user\_id) REFERENCES user\_info(id)

-- 确保只能插入已经存在于user\_info表格中的用户的权限信息

);

在 MySQL 中，常见的存储引擎有 InnoDB 和 MyISAM。一个在线书店系统更倾向于选择 InnoDB，因为它支持事务处理和外键约束。这对于确保数据完整性和支持复杂的查询非常有用。

总之，物理结构设计是数据库设计的重要组成部分，它确保了数据在物理存储层面上的高效组织和访问。

**3.6数据库实施**

数据库实施是将数据库设计从理论到实际应用的过程，包括创建数据库、定义表结构、存储过程等。创建了一个名为 "bookstore" 的数据库包含以下表格：

user\_info: 存储用户信息；

user\_permissions: 存储用户权限；

books: 存储书籍信息；

orders: 存储订单信息日期。

用户功能模块：

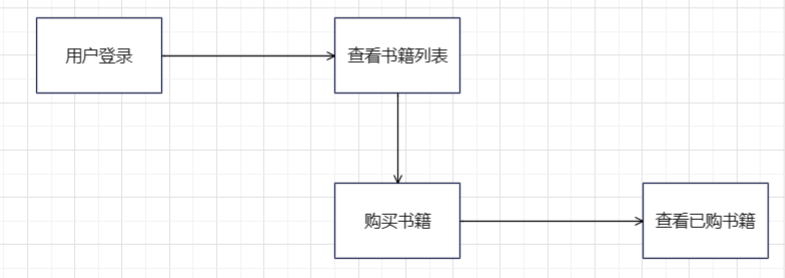


图 8 用户功能

**3.7系统调试和测试**

在系统实施完成后，我们需要对系统进行调试和测试，以确保系统的功能正确性和性能满足需求分析的要求。

登录和注册功能：

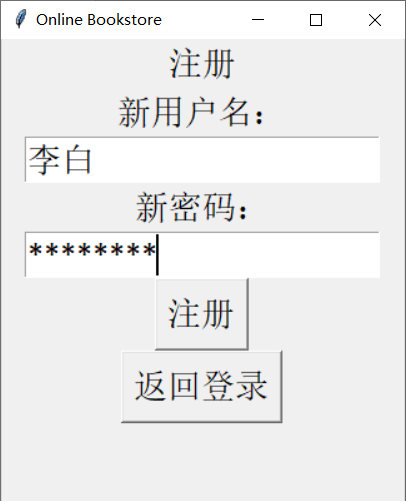


图 9 登录和注册

重复注册响应：

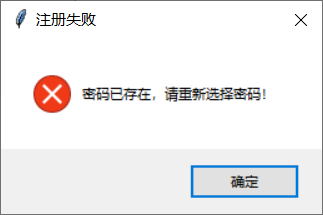


图 10 重复报错

用户主菜单：

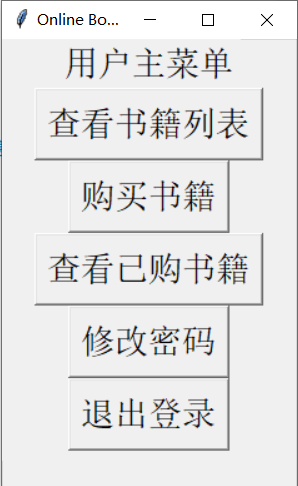


图 11 用户主菜单

管理员主菜单：

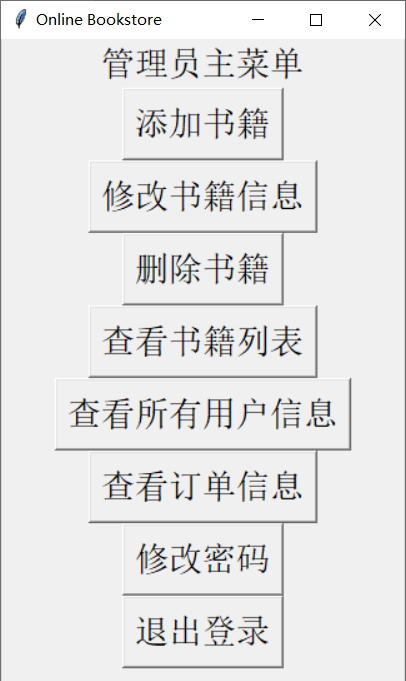


图 12 管理员主菜单

管理员查看订单：

****

图 13 订单

管理员查看用户信息：



图 14 用户信息

管理员修改密码：

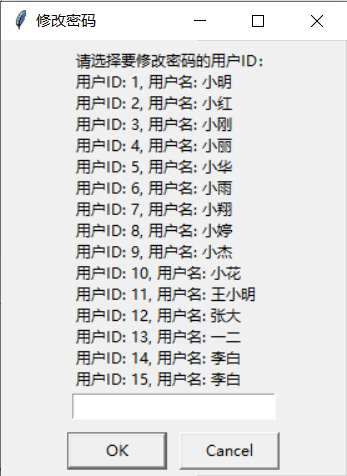


图 15 修改密码

管理员添加书籍：



图 16 添加书籍

管理员修改书籍：

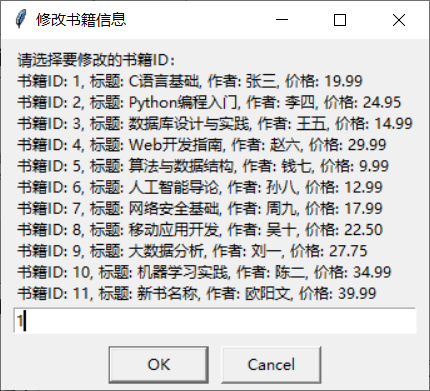


图 17 选择修改ID

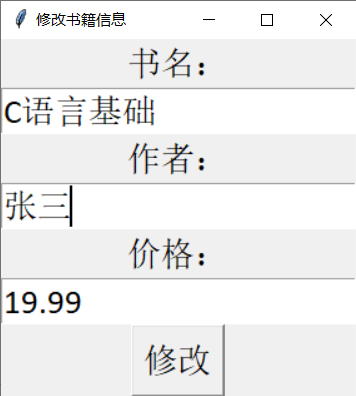


图 18 修改书籍信息

管理员删除书籍：

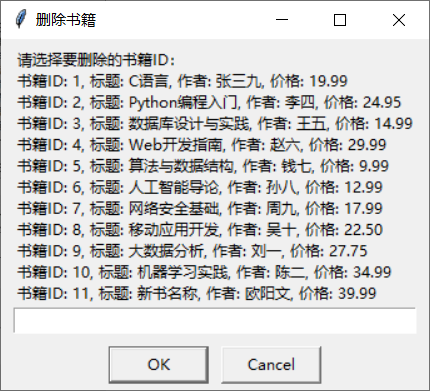


图19删除书籍

管理员查看书籍列表：



图 20 查看书籍

用户购买书籍功能：

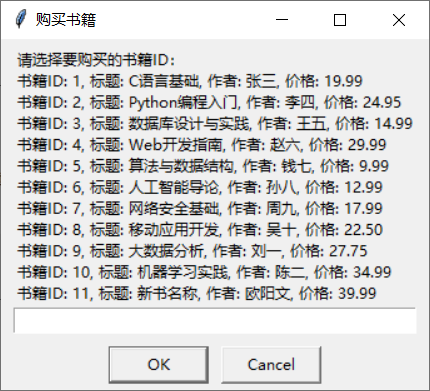
****

图 21 购买书籍

用户查看已购书籍功能：

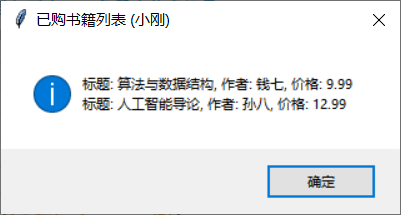
****

图 22 查看已购书籍

用户修改密码：

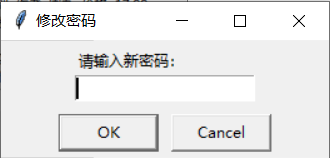


图 23 用户修改密码

通过调试和测试，我们验证了网上书店管理系统的功能正确性和性能，确保每个功能都按照预期运行。

**4 系统评价**

**4.1 系统特色**

用户认证和权限管理： 系统支持用户的注册、登录，并实现了基本的用户认证和权限管理。用户可以根据其角色（普通用户或管理员）执行不同的操作。

图形用户界面（GUI）： 使用 Tkinter 构建了直观的图形用户界面，使用户能够通过点击按钮和填写表单等简单的方式与系统进行交互。

订单管理： 实现了订单管理功能，用户可以查看他们的购买记录，管理员可以查看所有用户的订单信息。

密码安全性： 在用户注册和密码修改过程中，系统实现了一些安全性措施，例如检查密码是否已存在、密码是否与数据库中的其他密码重复等。

用户友好性： 界面使用了大字体，按钮和标签的布局使得用户界面更加友好和易于理解。

可扩展性： 提供了添加新书籍和新用户的功能，使系统具有一定的可扩展性，可以根据需求进行进一步的扩展和定制。

**4.2 系统不足及改进**

安全性加强： 当前系统在密码存储上需要更强的安全性。可以考虑使用哈希函数和加盐等技术来存储密码，以提高用户密码的安全性。例如，使用安全的密码哈希算法，如bcrypt，以替代简单的明文存储密码。

用户体验的改进： 尽管系统使用了 Tkinter 构建 GUI，但可以考虑通过改进界面设计和布局来提高用户体验。更直观的图形元素、更友好的错误提示等都可以改进用户体验。考虑使用吸引人的主题和颜色，以提升整体用户体验。

异常处理的细化： 对于一些特殊情况或异常情况，系统需要更细致的异常处理。这可以包括更具体的错误消息，以帮助用户和管理员更好地理解问题的根本原因。

日志记录： 引入日志记录机制可以帮助跟踪系统的操作和发现潜在问题。这对于系统管理员来说是非常有用的，可以帮助他们诊断和解决问题。后续可以记录关键操作和事件，包括用户登录、关键事务、错误等，以便管理员追踪系统；记录敏感操作和权限变更，以实现更严格的审计和安全监控；制定日志保留策略，确保日志信息可以在需要时用于审计或调查。

**5 设计心得**

在进行《数据库系统原理》课程设计的过程中，我遇到了一些问题，但通过不断努力和探索，最终都得到了解决。

例如，在进行需求分析时，我发现有时难以确定系统的具体需求。为了解决这个问题，我采用了多种需求调查方法，包括访谈、问卷调查和文献研究等，最终确定了系统的主要任务和具体要求。

在进行数据库设计时，我也遇到了一些困难。为了解决这些困难，我认真学习了数据库设计的基本原理和方法，并结合实际情况设计，设计了用户信息、权限、书籍和订单等表格，并使用外键建立它们之间的关系，最终建立了一个高效、可靠、易用的数据库。

使用 Tkinter 构建图形用户界面是一个学习过程。我学到了如何创建各种 GUI 元素，如标签、按钮和文本框，并将它们组织成一个用户友好的界面。在开发过程中，我通过手动测试确保图形用户界面的各个部分都按预期工作。这有助于发现和修复潜在的用户体验问题。

此外，在进行系统实施和调试时，我也遇到了一些问题。为了解决这些问题，我认真阅读了相关文档，并结合实际情况进行了调试和测试，最终确保系统的功能正确性和性能满足需求分析的要求。

总之，在进行《数据库系统原理》课程设计的过程中，我不仅学到了很多知识，而且锻炼了自己解决问题的能力。这对我今后的学习和工作都将产生积极的影响。