

# 合肥工业大学

## 计算机与信息学院

### 《数据库系统》课程设计报告

设计题目：书店书刊出租和零售管理系统  
项目组长：周布伟  
成 员：朱紫晨  
专业班级：计算机科学与技术 19-2 班  
指导老师：胡敏老师，沈明玉老师  
实验地点：计算机中心 304  
承担任务：客户管理、会员管理、图书借阅、零售及其查询

2022 年 06 月

# 目 录

1	系统设计概述 .....	1
1.1	项目背景 .....	1
1.2	设计目的及意义 .....	1
2	系统规划 .....	3
2.1	设计任务与要求 .....	3
2.2	软件环境与工具 .....	3
2.3	设计的分工 .....	3
3	需求分析 .....	4
3.1	用户需求说明 .....	4
3.1.1	数据需求 .....	4
3.1.2	事务需求 .....	6
3.2	系统需求说明 .....	7
3.3	数据流图 .....	8
3.4	数据字典 .....	9
4	数据库概念结构和逻辑结构设计 .....	11
4.1	概念结构设计：ER 图 .....	11
4.2	逻辑结构设计 .....	12
4.3	用户子模式设计 .....	15
5	数据库物理结构设计与系统的实施维护 .....	18
5.1	物理结构设计：索引 .....	18
5.2	实施维护：安全机制 .....	19
6	应用程序设计 .....	20
6.1	功能模块 .....	20
6.2	界面设计 .....	22
6.3	事务设计：触发器和存储过程 .....	23
7	测试和运行 .....	26
7.1	借阅管理模块的测试 .....	26
7.2	零售管理模块的测试 .....	29
7.3	会员管理模块的测试 .....	30
7.4	客户管理模块的测试 .....	31
7.5	其它模块的测试 .....	33
8	总结 .....	35
	附录 参考文献 .....	36

# 1 系统设计概述

本节将给出此次书店书刊出租和零售管理系统项目的背景和项目实施的目的与意义，重点介绍在计算机与互联网快速发展的背景下实现书店管理系统的目的、必要性和实现目标。

## 1.1 项目背景

随着计算机以及网络技术的不断发展，书店也进入了信息化的时代。书店，是文化的传播者，图书的收藏者，精神的守护者，是一座城市的文化地标。书店的好坏关系到一座城市的气质和内涵。书店的门类众多，不仅能够给读者提供安静舒适的场所，还能够带来知识的享受。在一些书店中，读者不但能够购买书刊，还能够出租借阅书刊。然而，当书店发展到一定的规模时，读者和书刊是每个书店管理者都需要考虑的因素，如何在能够管理好书刊和读者，又能带来一定的收益，是一个需要解决的关键问题。

## 1.2 设计目的及意义

计算机的快速发展使得数字化、信息化日益成熟，人们越来越感受到了计算机时代给人们日常生活带来的便利，随着伴随着计算机系统的普及，阅读需求的扩大越来越多的书店管理者利用计算机管理软件对书店进行经营管理，来借便于对书店进行管理，不断提高经营效益。尤其伴随着互联网在各行各业的渗透与应用，书店管理系统与互联网技术相辅相成，会员管理、余额管理、图书管理等功能也孕育而生。

书店在日常的经营管理过程中，需要实时了解书店中的书刊数量和种类、读者的个人信息和借阅情况，需要系统及时对读者和书籍产生的数据进行维护，才能确保书店的经营过程中不会出现经济的损失和形象的影响。同时，为了满足读者的不同阅读需求，书店的管理者应该根据读者的借阅和购买的情况调整经营的策略，动态更新书籍的种类和数量，以提高书店的经营效率和经济效益。

通过书店管理系统，书店管理员能够实时查看读者借阅和购买图书的情况，动态管理和维护书店中的书刊，并根据读者和时间做出及时的调整进货策略。书店管理员还能够根据书店的经营情况，针对读者的不同借阅需求，提供不同的会员优惠策略，以增加

书店的影响力和提高经济效益。

从长远的规划和经营来说，书店管理系统应该达到以下目标：

（1）目标的长期性：书店管理系统不仅能够满足当下的需要，还应该考虑更加长远的目标，应该把书店与时代发展相结合，不断改进与完善书店的经营策略，满足书店管理者和读者的需求，及时解决书店运营过程中出现的新问题。

（2）经济的效益性：书店管理系统与书店的经济效益密不可分，是需要考虑的重点因素。图书管理系统应该能够有效地维护客户的信息，尤其与余额相关的部分，涉及余额的充值与提现、图书的借阅与零售等；还应该能够针对不同的客户会员等级提供不同程度的优惠政策，对于损坏、逾期等行为应加入罚款机制。

（3）数据的可靠性：书店管理系统里保存的信息应该与书的真实信息相一致，如书刊的种类、数量，读者的姓名、联系方式、余额等内容。读者对书刊进行借阅、购买、预约等行为都应该记录在数据库中，并支持不同需求的查询功能，如根据时间、日期等信息进行查询。

## 2 系统规划

本节将介绍系统开发的规划，包括开发的任务要求、开发所使用的环境与工具，开发过程中的小组人员分工与安排，旨在明确设计的流程和开发环境，保证在规定的时间内满足设计的任务要求。

### 2.1 设计任务与要求

1. 设计任务：设计并实现书店书刊出租和零售管理系统
2. 设计要求

（1）信息要求：书店管理系统中应包括图书类别、期刊类别、出版社等基本信息管理，还应该管理图书信息、期刊信息、客户信息。

（2）功能要求：书店管理系统应该实现书刊的借阅、归还、预约、零售管理，包括信息的产生、查询等，针对读者的不同需求，设置不同等级的会员，提供不同的优惠策略。书店管理系统还应该实现余额管理，支持押金的收款、余额的充值和退款等功能，根据用户的借阅、购买、损坏等行为对余额进行扣除。

### 2.2 软件环境与工具

1. 操作系统环境：Windows 10
2. 开发环境：Tomcat8.5、Navicat Premium 12V、MySQL 5.5、IntelliJ IDEA2020.1 旗舰版、JDK8.0。

### 2.3 设计的分工

周布伟：参与需求分析，设计 ER 图，调整关系模式，设计存储过程，定义部分触发器。客户管理、会员类型管理、图书借阅、零售及其查询。

朱紫晨：参与需求分析，由 ER 图转换成关系模式，使用 MySQL 创建基本表和视图，定义部分触发器，完整性约束实现。图书管理、图书归还。

## 3 需求分析

第一节的系统设计和第二节系统规划指出了本项目的目标是旨在设计和实现一款支持书刊管理、借阅、归还、预约、零售管理、余额管理等功能的书店书刊出租和零售管理系统，并对设计的背景意义、开发环境、人员分工等方面进行了详细的介绍。本节将根据上文提出的设计任务与要求进行需求分析，然后设计数据字典和数据流图，其中主要包括用户需求和系统需求，着眼于了解用户的数据需求和事务需求，为后续的开发设计打下基础。

### 3.1 用户需求说明

用户需求主要包含数据需求和事务需求，数据需求描述了使用本书书店管理系统的用户所需要的数据，事务需求主要反映了使用者的实现某项事务的要求。

#### 3.1.1 数据需求

本系统主要涉及两种用户类型，分别是管理员和读者，其中，管理员主要管理书刊和读者的数据信息，读者主要提供个人信息，并参与书刊借阅和零售的全过程。针对用户类型的不同特点，分析其使用本系统时的数据需求。

##### 1. 管理员

管理员主要指书店的管理员，是系统中权限最大的用户，不但能够对书刊进行添加、修改、删除等管理操作外，还能够对读者客户进行添加、注销、修改信息等管理操作，包括用户名的变更、会员的升级与降级、余额的充值与提现等。对此，其数据需求主要包括书刊信息、读者客户信息、会员信息、借阅信息、零售信息等，以进行书店书刊和读者客户的一体化管理，提高经营效率和经济效益。

(1) 书刊信息，让管理员对书店中的书刊有直接的了解和记录，便于书刊的维护。包括书刊的编号、名称、出版社、价格、总量、在册数量、预约数量、作者、简介、书刊类别等信息。

(2) 读者客户信息，使管理员不仅记录经常光顾书店的读者信息，还能够帮助管理员实现书籍零售借阅的定位和追踪。包括读者的姓名、性别、手机号、会员等级、账

号状态、余额、已借阅数量等信息。

(3) 会员信息，让管理员能够自主设定借阅的额度、优惠措施和惩罚措施，扩大书店的影响力，增加书店的经济效益。包括押金、出租折扣、零售折扣、可借阅数量、可续借次数、预约保留天数、逾期扣费、损坏扣费、丢失扣费等。

(4) 借阅信息，为管理员能够快速查询和追踪读者与书籍的历史行为，进一步维护读者和书刊信息。包括客户编号、书刊编号、借阅时间、到期时间、续借次数、借阅状态、归还时间、扣费类型、扣费金额等。

(5) 零售信息，与借阅信息类似，同样能够为管理员提供读者与书刊历史行为的信息的追溯和管理，但由于零售后不存在归还的步骤，故其信息较借阅简单。包括客户编号、书刊编号、购买时间、购买数量等信息。

(6) 个人信息，主要便于管理员在系统上的登录与注册。包括管理员的用户名、密码和账号状态。

(7) 书刊类型信息，主要便于管理员在系统上管理书刊的类型，保证书刊的类型是统一的，不会出现同名异义或异名同义的情况，更便于规范化书店的图书分区和摆放位置。该数据需求在最初需求分析时忽略，经过验收时受老师启发后加入。

(8) 出版社信息，主要便于管理员了解和联系出版社，用于书刊的交易，该名称应该是统一的，不会出现同名异义或异名同义的情况。该数据需求在最初需求分析时忽略，经过验收时受老师启发后加入。

## 2. 读者（客户）

读者主要为本系统提供个人的信息，便于管理员进行读者的管理以及读者的书刊借阅与零售，必要时可以利用本系统进行个人信息、书刊信息和个人历史借阅信息的查询。因此，其相应的数据需求与管理员的数据需求有一定的重叠，但目的和所需的内容上又有区别。

(1) 书刊信息，让读者对书店中的书刊有直接的了解，便于书刊的借阅和购买。包括的信息与管理员的需求类似，重点关注书刊的编号、在册数量、名称、出版社、价格、总量、预约数量、作者、简介、书刊类别等信息。

(2) 会员信息，读者针对自己的借阅和购买的需求，通过会员办卡的方式成为书店的会员，并享有一定的权益，对此需要了解的信息包括会员的押金、出租折扣、零售折扣、可借阅数量、可续借次数、预约保留天数、逾期扣费、损坏扣费、丢失扣费等。

(3) 借阅信息，便于读者能够看到自己的历史行为，便于进一步的续借和归还决

策。包括读者自己所借阅的书刊的编号、借阅时间、到期时间、续借次数、借阅状态、归还时间、扣费类型、扣费金额等。

（4）零售信息，与借阅信息类似，同样能够为读者提供历史行为的信息的追溯和管理，包括读者自己购买的书刊的编号、购买时间、购买数量等信息。

（5）书刊类型信息，便于读者查询、借阅和购买期望的书刊，通过期刊类型查询能够快速限制书刊的范围和位置。

### 3.1.2 事务需求

#### 1. 管理员

管理员不仅要维护书刊、读者、会员类型等基本信息，还要记录读者的借阅、零售、预约等操作，包括书刊管理、读者管理、会员类型管理、借阅与零售管理等。

（1）书刊管理，图书、期刊等数量、名称、出版社、价格、总量、预约数量、作者、简介、书刊类别等信息管理；书刊基本信息，如名称、数量、作者、出版社、价格等的添加、删除、修改；查询书店中图书的信息等。

（2）读者管理，主要有读者的添加与注销，读者信息的修改，如姓名、性别、会员等级、账号状态、余额等信息。

（3）会员类型管理，支持不同等级的会员优惠和惩罚措施，涵盖借阅、零售，为客户提供有针对性的服务。

（4）借阅管理，实现读者对书刊的借阅，记录操作的时间、操作的业务对象等，借阅操作中还涉及归还、续借的管理与记录；

（5）零售管理，与借阅管理类似，需要完成读者书籍的零售操作，记录读者的零售的时间、书刊、数量等信息。

（6）预约管理，支持记录读者的预约请求，为读者保留预约的书刊，到达一定期限后自动撤销预约的请求，期限由会员等级决定，期限内管理员可以为读者提供对书刊的借阅或购买操作。

（7）修改密码，为了保证系统的安全性，管理员的密码要求在数据库中以密文的形式加密保存。

#### 2. 读者（客户）

读者的事务需求主要包括对书籍信息的查询、个人历史借阅与零售的操作和查询、余额的提现和充值等，因而主要涵盖了书刊查询、借阅操作、零售操作、预约操作等需



求。

(1) 书刊查询，使读者能够通过关键字查找目标书刊，对书刊的作者、出版社、数量、价格等信息有直接的了解。

(2) 借阅操作，让读者能够完成对书刊的借阅，支付一定的租金，在此期限内读者可以将书刊带走阅读，并指定的时间内归还，否则会产生逾期的费用，若归还时书籍损坏或丢失，这也会造成相应的费用。

(3) 零售操作，让读者能够完成对书刊的购买，通过支付书刊的费用，读者便可以为完全占有书刊的所有权，不同会员等级的读者也可享有不同的零售折扣。

(4) 预约操作，读者可以对不在书店中，已经借出的书刊进行预约操作，预约期限内，除该读者以外的其他读者无法对已经预约的书刊进行借阅或购买的操作，直至预约到期。

## 3.2 系统需求说明

书店书刊出租和零售管理系统不仅需要满足管理员和读者用户的功能需求，也应满足以下几个方面的性能需求。

### 1. 安全性

安全性主要包括系统应用安全、系统管理与维护安全、数据存储安全、数据的传输安全等。在书店书刊出租和零售管理系统中，需要引入认证机制、日志管理机制等维护系统应用安全，需要通过设定访问控制权限，规定用户能够进行的操作和不能进行的操作，实现系统管理与维护安全，确保无关人员和无权限的人员不能读取和更改系统内的模块和数据，以防止数据库中的数据被这些无关人员恶意破坏。某些敏感数据的存储，需要以密文的形式存储，防止数据库泄露对客户信息的破坏性影响，以确保数据存储的安全。确保系统长时间可靠、稳定的运行。要避免非法人员假冒，给书店带来经济损失。

### 2. 实用性

保证管理员和读者无需借助过多的教程，要有完善的系统运行保障机制，在系统的操作上要求简单明了，读者与管理人员都很方便的使用该系统。主要通过图形化的方式完成所需的业务操作，且操作的过程不会出现任何程序内部的错误，系统运行顺畅无卡顿，无闪退等严重 BUG。系统稳定性好，崩溃率低。系统响应速度快。系统对绝大多数功能的响应时间应该在人的明显感知时间以内。进行业务操作时的查询的查全率和查

准率为 100%，所在相应域中包含查询关键字的记录都能查到，所有在相应域中不包含查询关键字的记录都不能查到，能够为管理员和读者提供完全的技术支持。

3. 可扩展性与易维护性

书店书刊出租和零售管理系统应该能够不断顺应互联网和市场的不断变化，尽可能地根据业务变化以最小的消耗来及时扩展。系统在功能模块上面要预留相应的接口以方便系统不仅在今后的更新、管理、维护方面带来极大的便利还要保护书店现有的投资，弹性可伸缩，扩展的系统要具备较高的性价比。

4. 开放性

考虑到书店的长期利益，不同的产品供应商的机具必须兼容其它厂商的设备从而最大限度的保护书店的利益。系统在开发的过程当中必须结合国家颁布的相关的标准或者是工业标准。

3.3 数据流图

通过对用户和系统的需求分析，我们得到了详细的数据描述，对此可以通过数据流图定义数据项和数据结构，描绘数据信息在系统中流动和处理的情况和逻辑内容。

1. 系统顶层数据流图

书店书刊出租和零售管理系统的使用者主要是书店管理人员，服务对象是用户（读者）。读者来到书店主要进行图书的借阅、预约、归还、购买和零售等事务，书店管理人员使用该系统为宾客完成用户要求的业务，读者接受服务后，离开书店，书店管理员还可以管理书刊信息和会员信息。系统顶层数据流图如图 1 所示。

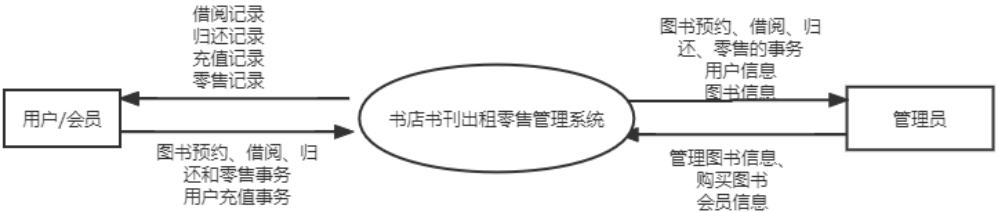


图 1 系统顶层数据流图

2. 系统 0 层数据流图

将系统顶层数据流图进一步细化，可以得到 0 层数据流图，书店书刊出租和零售管理系统的 0 层数据流图如图 2 所示，其中涵盖了借阅管理、零售管理、客户管理、书刊管理、会员管理等事务的数据流与数据存储表示，用户（读者）和管理员作为系统



余额记录信息	balance	记录每次金额变更的信息表格	余额记录信息=操作时间+客户 ID+操作金额+操作类型
书刊信息	book	记录每本书刊具体的信息表格	书刊信息=ISBN+书刊名称+出版社+价格+总量+在册数量+预约数量+单位+作者+简介+书刊类型+ 书刊类别
借阅信息记录	borrow	记录每次借阅发生的具体信息的表格	借阅信息=客户 ID + ISBN + 借阅时间+到期时间+续借次数+借阅状态+归还时间+扣费类型+扣费金额
零售信息记录	sell	记录客户购买图书记录信息的表格	零售信息记录=客户 ID+ISBN+购买时间+购买数量
书刊预约信息记录	reserve	记录客户预约图书记录信息的表格	书刊预约信息记录=预约时间+客户 ID+ISBN+预约到期时间

---

## 4 数据库概念结构和逻辑结构设计

通过第三节的需求分析,我们已经明确了用户和系统的需求,包括项目的使用对象,即管理员和客户的数据库需求和事务需求。本节将针对需求分析进行数据库的概念结构和逻辑结构的设计,旨在将需求分析中的用户需求抽象为概念模型,并将概念模型转换为数据库管理系统所支持的数据模型相符合的逻辑结构。

### 4.1 概念结构设计: ER 图

为了描述现实世界中的概念模型,进一步转换需求分析中的数据流图和数据字典,我们使用 E-R 图辅助概念结构的设计,如图 3 所示,其中包含了会员、客户、书刊、余额记录、管理员等实体,其相关的属性用椭圆形标识,并与实体连接,实体之间发生多种联系,部分联系包含了相关的属性。

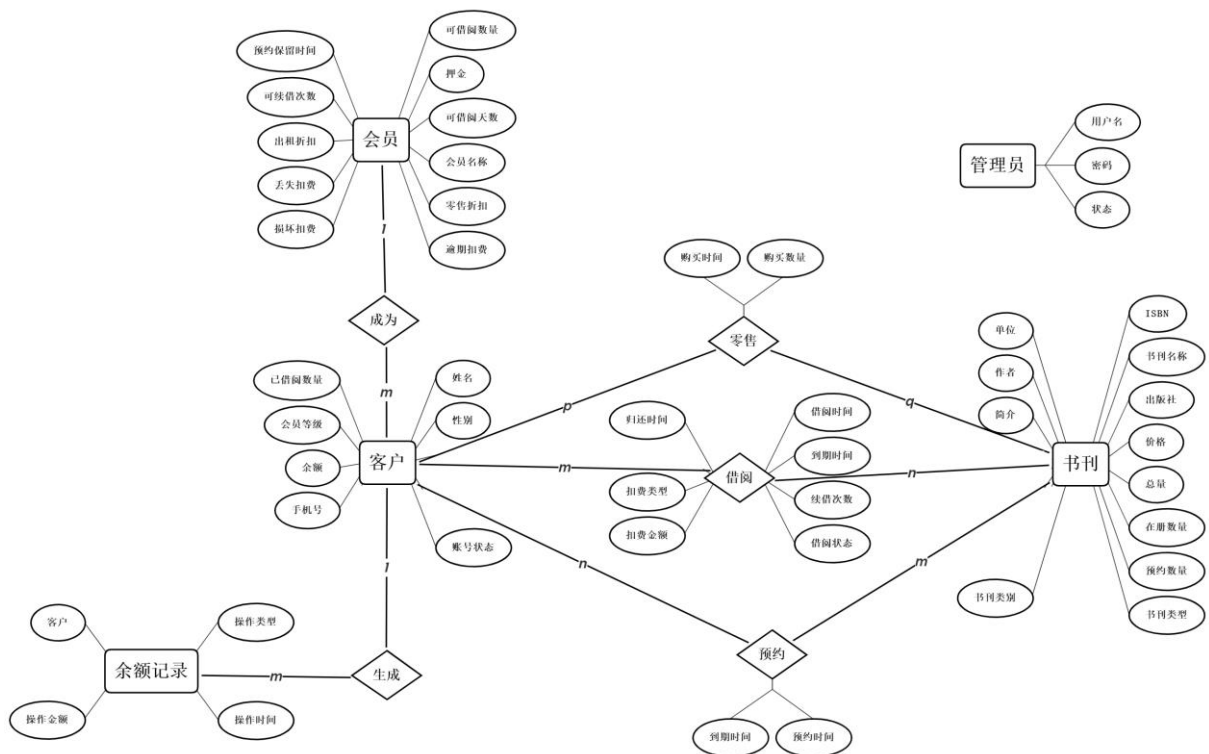


图 3 宾馆客房管理系统的 ER 图

由于管理员不直接参与整个书店的操作业务中,所以并未与其他实体发生联系,但是它的基本属性是之后设计数据库表和具体应用的前提,故在此将其单独列出。客户与书刊之间主要发生了借阅、零售、预约等联系,都是多对多的联系,即一位读者可以同

时借阅、零售购买、预约多种书，一种书可以被多个读者借阅、零售购买、预约，其中包含了发生联系之后增加的属性。值得注意的是，此处使用一种书作为基本的实体，而非一本书，故其中还包含了总量、在馆数量等信息。此外，在需求分析中还涉及到会员管理和余额管理等需求，在此专门定义了相关的实体用于后续的详细设计，其中会员包含了押金、可借阅数量、可借阅天数、出租折扣、零售折扣等基本属性，余额记录包括了操作类型、操作金额等属性，且是一对多的联系。

在后续的验收中发现，为了便于书店管理员进行书刊类型分区，与书店出版社进行交易，避免出现同名异义或异名同义的情况，E-R 图还应该加入书刊类型和出版社，并将其作为实体，而非仅仅是其他实体的属性。

## 4.2 逻辑结构设计

由于概念结构是独立于任何一种数据模型的信息结构，为了便于后期数据库设计和项目开发，我们需要进行逻辑结构的设计，即把概念设计阶段设计好的概念模型 E-R 图转换为关系模型，得到最初如下 8 个关系模式，关系的主码用下划线标出：

客户（手机号，姓名，性别，会员等级，余额，账号状态，已借阅数量）

管理员（用户名，密码，状态）

余额记录（操作时间，客户 ID，操作金额，操作类型）

会员（会员名称、押金、出租折扣、零售折扣、可借阅数量、可续借次数、可借阅天数、预约保留时间、逾期扣费、损坏扣费、丢失扣费）

书刊（ISBN、书刊名称、出版社、价格、总量、在册数量、预约数量、单位、作者、简介、书刊类型、书刊类别）

借阅（客户 ID、书刊 ID、借阅时间、到期时间、续借次数、借阅状态、归还时间、扣费类型、扣费金额）

零售（客户 ID、书刊 ID、购买时间、购买数量）

预约（预约时间、客户 ID、书刊 ID、预约到期时间）

在关系模式的转化过程中讨论发现，虽然使用手机号、ISBN 号能够在一定程度上保证元组的唯一与非空，为主键的选择提供了方便，但是从实际生活场景和数据库的设计实现角度出发，一方面，使用手机号、ISBN 作为主键等并不能保证顺序性。自增 ID 是顺序的，可以保证索引树上的数据比较紧凑，有更高的空间利用率以及减少数据页的

分裂合并等操作，提高数据库效率；另一方面，现实生活中可能出现手机号转让，书籍 ISBN 规范更改等情况，且书籍和期刊的编号规则有一定的区别，因此将引入自增 ID 作为部分关系模式的主码。改进后的 8 个关系模式如下：

客户（客户 ID，姓名，手机号，性别，会员等级，余额，账号状态，已借阅数量）

管理员（用户名，密码，状态）

余额记录（操作时间，客户 ID，操作金额，操作类型）

会员（会员名称，押金，出租折扣，零售折扣，可借阅数量，可续借次数，可借阅天数，预约保留时间，逾期扣费，损坏扣费，丢失扣费）

书刊（书刊 ID，书刊名称，书刊编号，出版社，价格，总量，在册数量，预约数量，单位，作者，简介，书刊类型，书刊类别）

借阅（客户 ID，书刊 ID，借阅时间，到期时间，续借次数，借阅状态，归还时间，扣费类型，扣费金额）

零售（客户 ID，书刊 ID，购买时间，购买数量）

预约（预约时间，客户 ID，书刊 ID，预约到期时间）

将上述各关系模式在 MySQL 数据库平台上实现，使用 CREATE 语句为其创建相应的数据库表，对应的 8 张基本表结构和满足的规范化程度如表 2~表 9 所示，可见所有的表都满足较高的规范化程度，几乎不存在数据的操作异常，且具有较低的冗余。

表 2 客户表 Client: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
客户 ID	clientID	char	11	主码	否	
手机号	clientPhone	char	11		否	
姓名	clientName	varchar	30		否	
性别	clientSex	char	1			check 男/女
会员等级	clientGrade	varchar	8	外码	否	
余额	clientBalance	decimal	10,2		否	
账号状态	clientStatus	varchar	4		否	正常、冻结、注销
已借阅数量	clientBorrowNum	smallint	2		否	

表 3 管理员表 Admin: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
用户名	adminName	varchar	30	主码	否	
密码	adminPassword	varchar	20		否	
状态	adminStatue	varchar	2		否	check 正常、注销

表 4 余额记录表 Balance: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
操作时间	opTime	date	6	主码	否	
客户 ID	clientID	varchar	11	主码, 外码	否	
操作金额	opAmount	decimal	10,2		否	
操作类型	opType	varchar	4		否	

表 5 会员表 VIP: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
会员名称	vipName	verchar	10	主码	否	
押金	vipDeposit	decimal	10,2		否	
出租折扣	vipRentDiscount	decimal	5,4		否	
零售折扣	vipSellDiscount	decimal	5,4		否	
可借阅数量	vipBorrowNum	smallint	2		否	
可续借次数	vipRenew	smallint	2		否	
可借阅天数	vipBorrowDate	smallint	4		否	
预约保留时间	vipTime	smallint	2		否	
逾期扣费	vipOverdue	decimal	4,2		否	
损坏扣费	vipDamage	decimal	4,2		否	

表 6 书刊表 Book: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
书刊 ID	bookID	char	13	主码	否	
书刊编号	bookNum	char	13		否	
书刊名称	bookName	varchar	30		否	
出版社	bookPublic	varchar	30		否	
价格	bookPrice	decimal	10,2		否	
总量	bookTotalNum	smallint	3		否	
在册数量	bookNowNum	smallint	3		否	
预约数量	bookReserveNum	smallint	3		否	
单位	bookUnit	varchar	3		否	check 本
作者	bookAuthor	varchar	30		否	
简介	bookIntroduction	varchar	500			
书刊类型	bookType	varchar	10		否	check 书籍、期刊
书刊类别	bookCategory	varchar	10		否	

表 7 借阅表 Borrow: 第三范式

属性名	字段名	类型	宽度	主/外码	空值	描述
客户 ID	clientID	char	13	主, 外	否	



ISBN	bookISBN	char	13	主，外	否	
借阅时间	borrowDate	date	6	主	否	
到期时间	expiryDate	date	6		否	
续借次数	renewNum	smallint	2		否	
借阅状态	borrowStatus	verchar	3		否	check 在借、归还
归还时间	returnDate	date	6			
扣费类型	deductionType	verchar	3			无、逾期、损坏、丢失
扣费金额	deductionAmount	decimal	10,2			借阅金额+扣费金额

表 8 零售表 Sell: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
客户 ID	clientID	char	13	主，外	否	
ISBN	bookISBN	char	13	主，外	否	
购买时间	buyDate	date	6	主	否	
购买数量	buyNum	smallint	3		否	

表 9 预约表 Reserve: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
预约时间	reserveDate	date	6	主	否	
客户 ID	clientID	char	13	主，外	否	
ISBN	bookISBN	char	13	主，外	否	
预约到期时间	reserveExpriyDate	date	6			归还时间+期限

### 4.3 用户子模式设计

通过逻辑结构设计，我们将概念模型转换为逻辑模型，定义了关系模型和基本表，然而，该逻辑模型是全局的，对应于数据库三级结构中的模式，还需要根据局部应用需求，结合具体的 MySQL 关系数据库管理系统设计用户的外模式，可以通过定义视图来实现。视图是从一个或几个基本表（或视图）导出的表。它是一个虚表。使用视图不仅能够简化用户对系统的使用，还能够确保数据库在一定程度上的安全和提高数据的独立性。

为了实际应用的需要，本书店管理系统中建立了 15 张视图，主要包括读者会员信息视图、读者借阅视图、借阅信息视图等，以下将列举本人参与设计和较为典型的视图进行介绍。

#### 1. 读者会员信息视图

由于在读者进行借阅、零售、归还等操作时需要对读者的会员信息进行查询，以获

得读者拥有的折扣优惠、罚款额度、借阅期限、借阅数量等信息，为了简化查询，通过以下语句将客户表和会员表进行连接构建视图，使用该视图，如图 4 所示，在完成读者业务操作执行查询语句时，无需使用复杂的语句，只需查询视图中的内容即可，类似的还有书刊借阅视图、书刊零售视图等，在此不再赘述。

```
select `client`.`clientGrade` AS `会员等级`,`client`.`clientPhone` AS `手机号`,`client`.`clientName` AS `用户名`,`client`.`clientBalance` AS `用户余额`,`client`.`clientStatus` AS `账号状态`,`client`.`clientBorrowNum` AS `已借阅数量`,`vip`.`vipRentDiscount` AS `出租折扣`,`vip`.`vipSellDiscount` AS `零售折扣`,`vip`.`vipDeposit` AS `押金`,`vip`.`vipBorrowNum` AS `可借阅数量`,`vip`.`vipRenew` AS `可续借次数`,`vip`.`vipBorrowDate` AS `可借阅天数`,`vip`.`vipTime` AS `预约保留时间`,`vip`.`vipOverdue` AS `逾期扣费`,`vip`.`vipDamage` AS `损坏扣费`,`vip`.`vipLose` AS `丢失扣费` from (`client` join `vip`) where (`client`.`clientGrade` = `vip`.`vipName`)
```

会员等级	手机号	用户名	用户余额	账号状态	已借阅数量	出租折扣
普通会员	13811199220	zbw	-37.80	注销	0	0.0500
普通会员	13999999955	zzzz	97.75	正常	1	0.0500
普通会员	14790409310	张一2	77.18	正常	2	0.0500
普通会员	18856316698	橙子	59.10	正常	2	0.0500
黄金会员	13616644404	super	-6.40	正常	1	0.0300
黄金会员	13999999333	ZZZ1	0.00	注销	0	0.0300

图 4 读者会员信息视图

## 2. 客户信息视图

在定义客户表时，我们给出了客户 ID，姓名，手机号，性别，会员等级，余额，账号状态，已借阅数量等属性，然而，在实际使用时，不同的应用场景所需要使用的属性不同，且多数情况下并不需要使用所有的属性，如进行用户信息的更改、余额的充值等情况，对此，我们定义行列子集视图，旨在限制对基本表的操作范围，与应用无关属性的出现，进而一定程度上保证安全性，定义的语句如下，结果如图 5 所示。

```
select `client`.`clientPhone` AS `clientPhone`,`client`.`clientName` AS `clientName`,`client`.`clientSex` AS `clientSex`,`client`.`clientGrade` AS `clientGrade`,`client`.`clientBalance` AS `clientBalance` from `client`
```

clientPhone	clientName	clientSex	clientGrade	clientBalance
13616644404	super	男	黄金会员	-6.40
13811199220	zbw	男	普通会员	-37.80
13999999333	ZZZ1	男	黄金会员	0.00
13999999900	ZZ1	男	黄金会员	62.20
13999999955	zzzz	男	普通会员	97.75

图 5 客户信息视图

## 3. 数据统计视图

实际应用场景中有时需要统计客户的借阅或购买的历史信息，需要使用聚集函数并定义视图来满足相应需求，以客户借阅书刊数量统计为例，采用如下语句，其中通过聚集函数 COUNT 计数，用 GROUP BY 区分不同的客户，进而实现了对借阅书刊数量的统计，定义结果如图 6 所示。

```
select `v_c_b_bw`.`clientPhone` AS `clientPhone`,`v_c_b_bw`.`clientName` AS `clientName`,`v_c_b_bw`.`bookName` AS  
`bookName`,count(0) AS `borrowNum` from `v_c_b_bw` group by  
`v_c_b_bw`.`clientPhone`,`v_c_b_bw`.`clientName`,`v_c_b_bw`.`bookName`
```

clientPhone	clientName	bookName	buyNum
13616644404	super	数据挖掘	1
13999999900	ZZ1	活着	1
14790409310	张一2	数据挖掘	1
14790409310	张一2	活着	1
18856316698	橙子	数据挖掘	1

图 6 数据统计视图

## 5 数据库物理结构与系统的实施维护

数据库在物理设备上的存储结构和存取方法称为数据库的物理结构，它对应于三级结构中的内模式。本节将给逻辑数据模型选取合适的物理结构，主要考虑数据库的索引结构。然后从数据库和系统的安全性角度出发，对系统的实施维护进行设计。

### 5.1 物理结构设计：索引

为了加快查询的速度，我们在物理结构上进行优化，即定义若干索引，快速定位到需要查询的内容。索引主要给数据库管理系统使用，数据库管理员无法使用索引，只能定义索引，而索引是把“双刃剑”，它需要占用一定的存储空间，如果数据库表频繁地发生更新，那么索引的维护成本和负担就会增加。

本系统除了 MySQL 会默认在各表的主码上建立索引来方便进行实体完整性检验之外，还考虑建立 3 个索引。

#### 1. 会员名称索引

当读者进行借阅、零售、归还等业务操作时，系统往往需要查询该读者的会员名称，以确定借阅期限、借阅数量、出租折扣等会员信息，当用户量增加时，对会员名称的查询就有可能成为瓶颈，对此需要建立会员名称索引，以加快查询，而且，当读者开卡后，对读者基本表的操作相对较少，不会因为索引而造成更大的开销。

#### 2. 借阅和零售的日期索引

随着读者数量的增加和业务的不断发生，借阅、零售等基本表都会大规模的数据量，而实际应用场景中需要通过日期判断读者是否发生逾期，或统计某段时间内书店借阅和零售的数量，由此需要建立借阅和零售的日期索引以提高查询的速度，但是，这类数据表的数据更新频率高，在所有的维护上也需要一定的开销，故还需要视实际应用场景测试后再做决定。

#### 3. 客户名称和手机号索引

在实际的应用场景中很少直接通过客户的 ID 来查询信息，更多的时候是通过客户名称或手机号进行查找，为了提高查找速度，可以为其建立索引，且如上文所述，该类数据表中数据的维护并不复杂，不需要为索引造成更大的开销。

## 5.2 实施维护：安全机制

系统主要从数据库和系统本身来考虑安全性，通过数据库设计和系统开发，主要采用如下措施确保系统和数据库的安全性。

1. 采用数据备份与恢复策略，书店管理员可以随时备份和恢复数据，系统还能够定期保存数据，避免突发事件发生带来的毁灭性影响；
2. 使用 MD5 哈希函数以密文的形式对敏感信息进行保存，如管理员密码等，防止数据库泄露或非法用户入侵对管理员和客户信息造成的损害；
3. 基于 Session 机制，防止非法用户绕过登录功能直接进入系统，加入验证码机制，防止不法分子使用机器人登录账号；
4. 采用权限管理，建立健全规章制度，加强工作人员的安全防范意识，只有书店管理员才能进入系统，负责系统的客户和书籍信息；
5. 使用视图，不仅简化查询，提高数据独立性，还能保护其他数据，限制业务操作的作用范围。

## 6 应用程序设计

通过需求分析和数据库设计，本项目已经具有一定的雏形，基于 JavaWeb，便可以进行项目网页端的应用程序设计与开发。下面将介绍本项目的功能模块、界面设计和事务设计，并重点本系统中我所实现的部分。

### 6.1 功能模块

本项目面向书店管理员进行设计，主要设计开发管理员端和客户端的应用，其中管理员端包括书刊管理、借阅管理、零售管理、会员管理、客户管理等功能模块，客户端包括借阅管理、零售管理、客户管理。二者的模块内部的实现存在一定的差异。

在需求分析中发现目前大部分中小型的书店都设有专门的收银台进行书刊的借阅与零售操作，而读者自助操作的实现需要依托还书机器、图书磁性条码感应等硬件设备，无法单独依托于手机或电脑网页实现借阅的操作，故在客户端并没有加入自助借阅和零售的功能，这些功能的实现需要读者自行到收银台完成借阅或零售的操作，系统为读者保留了续借、预约和查询的功能，支持个人信息的修改。

#### 1. 管理员端

管理员端支持系统中的大部分功能，功能框图如图 7 所示，主要功能包括书刊及其类型的添加、下架、修改和查询，实现会员和客户的管理，并支持书刊的借阅与零售。

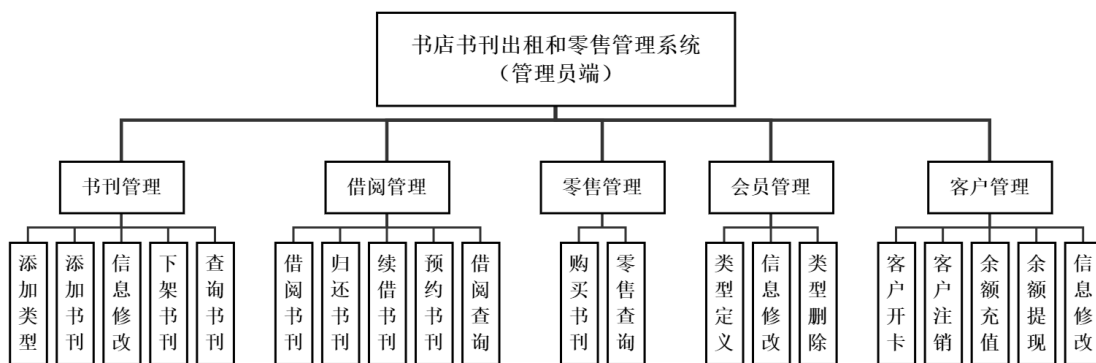


图 7 管理员端功能框图

书刊管理中，管理员能够为书刊添加相应的类型，如文学类、计算机类等，能够添加书刊信息，并支持信息的修改和查询操作，考虑到书刊在书店借阅和购买的过程中在客户和管理员之间产生流动，形成记录，若直接删除书刊的信息，这些记录也有可能随

之消失，无法在必要时进行审查，故在此并未做真正的删除操作，而是将书刊的在馆数量清零，无法进行借阅和购买的操作，因而完成了下架的处理。

借阅管理中，读者能够在管理员端通过管理员人工完成借阅、归还、续借和预约的操作，借阅过程中能够查询读者和书刊的信息，操作过程中将依据客户的会员信息产生一定的费用，其中形成的记录可以查询。

零售管理与借阅管理类似，需要读者通过管理员人工购买，购买完后会进行费用的结算，其中形成的记录可以查询。

会员管理定义了会员的借阅期限、借阅数量、出租折扣、零售折扣、逾期罚款额度等信息，书店管理员可以依据书店的经营情况进行适当的调整，由于会员信息与余额相联系，所以当管理员配置后系统会自动检查是否会影响客户，如缩短借阅时长后导致部分客户出现逾期的情况等。会员信息的删除并不是真正的删除，会员的记录将会保留，只是不会参与客户的事务中。

客户管理主要面向客户，支持客户的添加和信息修改，支持客户余额的充值和提现，确保完整性的约束，由于书店的客户会形成借阅和购买的记录，不能对其进行删除的操作，在此用注销代替，注销后的客户不再参与事务中。

2. 客户端

结合特定的应用场景，客户端所实现的功能是管理员端的子集，在具体功能细节上存在一定的差异，如图 8 所示，其中主要包括书刊的续借、预约和查询，零售的查询，对个人信息的修改和余额的充值提现等业务功能。

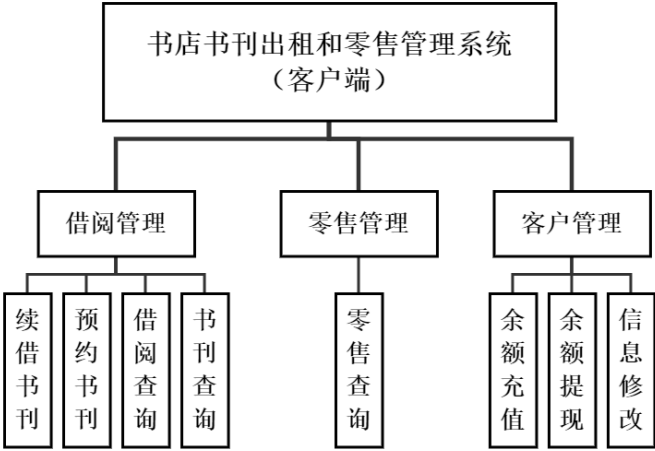


图 8 客户端功能框图

借阅管理中，客户可以对已经借阅的书刊进行续借操作，延长借阅的期限，还可以

对未借阅的书刊进行预约，预约期限内其他读者无法进行借阅或购买的操作，只有该读者能够进行借阅。客户还可以查询书刊信息和个人的借阅信息，此处查询的内容与管理员端有所区别，客户只能查询到与自己相关的信息。

零售管理中，客户可以查看自己的购买书刊所形成的记录。

客户管理中，客户可以修改自己的个人信息，对自己的余额进行充值和提现，由于处在设计与开发阶段，该功能拟定支持微信和支付宝的方式，通过线上的方式即可完成充值和提现操作。

## 6.2 界面设计

依据系统的需求设计，本系统设计主要设计了登录界面和主界面，基于此可以满足管理员和客户的相关需求。

### 1. 登录界面

书店书刊出租和零售管理系统的登录界面如图 9 所示，主要包含封面图、版权信息、账号密码提示框，用户根据界面中的提示输入用户名、密码和验证码，即可登录系统，完成相关的业务操作，若用户的用户名、密码或验证码输入错误，系统将提示错误信息，并让用户返回重新登录。



图 9 登录界面

### 2. 用户主界面

书店书刊出租和零售管理系统的用户主界面如图 10 所示，系统的主界面整体结构为左右结构，左侧为选项栏，展示了当前用户的所有功能模块，右侧区域主体部分为内容区，用于显示选项栏的项目内容，界面右上角显示了当前用户的用户名称，用户可点



击“安全退出”退出系统。后续的功能模块将基于该网页框架进行设计和开发。

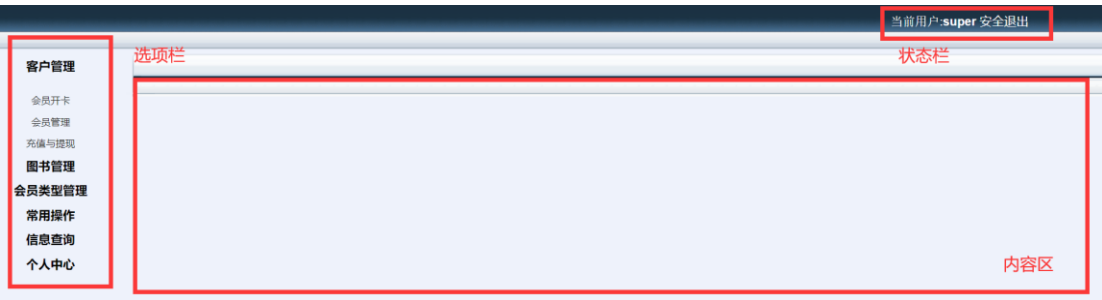


图 10 用户主界面

### 6.3 事务设计：触发器和存储过程

本系统提供了书刊管理、借阅管理、零售管理、会员管理、客户管理等事务处理的支持，主要通过 JavaWeb MVC 框架和嵌入式 SQL 编程开发实现，还结合触发器和存储过程，不仅能够实现完成事务的流程，还能给为系统的完整性和安全性提供保障，本节将介绍系统中我所设计运用的触发器和存储过程。

#### 1. 触发器

触发器能够在任何用户对表的增、删、改操作时由服务器自动激活，可以实施更为复杂的检查和操作，保证系统的完整性约束。在本系统中，有单位限制、在馆数量修改（借阅、零售、归还）、余额交易记录等多个触发器，其中，我实现的是余额交易记录的触发器。

该触发器定义在客户表 Client 上，当读者进行余额的充值和提现操作时，触发器就会记录当前操作的客户信息、操作时间和操作的金额，并根据操作的金额判断操作的类型是充值还是提现，其具体的定义实现如下，主要判断新的余额值和旧的余额值之间的大小关系，如果新的值大于旧的值，则为充值，反之为提现，并将结果插入到余额表 Balance 中。

```
IF new.clientBalance > old.clientBalance THEN
INSERT INTO `bookdb`.`balance`(`opTime`, `clientPhone`, `opAmount`, `opType`) VALUES (NOW(), new.clientPhone,
new.clientBalance - old.clientBalance, '充值');
ELSEIF new.clientBalance < old.clientBalance THEN
INSERT INTO `bookdb`.`balance`(`opTime`, `clientPhone`, `opAmount`, `opType`) VALUES (CURRENT_DATE,
new.clientPhone, old.clientBalance - new.clientBalance, '提现');
END IF
```

#### 2. 存储过程

系统在进行借阅、销售、信息查询的过程中往往到多张数据表的查询和修改操作，

后端往往需要编译并执行多条 SQL 语句，与数据库进行多轮的交互才能完成既定的事务，在这过程中往往需要多次连接数据库且容易出现问題，对此本系统定义了借阅、销售、指定时间段查询数量、指定客户查询数量等多个存储过程简化这一过程，存储过程将复杂操作作用存储过程封装起来与数据库提供的事务处理结合一起使用，不仅能够提高数据库执行速度，还能够重复使用，减少数据库开发人员的工作量，进一步保障数据库的安全性和稳定性。在本系统中，我设计并实现了借阅、销售、指定时间段查询数量、指定客户查询数量等多个存储过程。

### （1）存储过程 insertBorrow

该存储过程实现读者的借阅业务，完成借阅操作和读者费用的扣除，通过传入读者的 ID 和书刊的 ID，利用视图查询读者的会员信息，得到可借阅天数和出租折扣，利用书刊的 ID 查询书刊表 Book，得到书刊的价格，计算可得此次借阅过程的费用，然后向借阅表 Borrow 中插入借阅信息，其中根据客户的可借阅天数设定到期时间，通过 DATE\_ADD(NOW(),INTERVAL @time DAY)进行计算，利用 CURRENT\_DATE 获取当前时间，插入信息的同时，与之相应的触发器会启动，更改在馆的图书数量。最后更新客户的余额。实现过程如下。

```
CREATE DEFINER='root'@'localhost' PROCEDURE `insertBorrow`(IN `clientID` char(11),IN `bookID` char(13))
BEGIN
    #Routine body goes here...
    set @time = 0;
    set @ratio = 0.00;
    set @price = 0.00;

    select `可借阅天数`,`出租折扣` into @time, @ratio from v_c_vip where `手机号`=`clientID`;
    select bookPrice into @price from booknew where bookISBN = `bookID`;
    set @cost = @ratio * @price;

    INSERT INTO `bookdb`.`borrow`(`clientPhone`, `bookISBN`, `borrowDate`, `expiryDate`, `renewNum`,
`borrowStatus`, `returnDate`, `deductionType`, `deductionAmount`) VALUES (`clientID`, `bookID`, CURRENT_DATE,
DATE_ADD(NOW(),INTERVAL @time DAY), 0, '在借', NULL, NULL, NULL);
    UPDATE v_c_info SET clientBalance = clientBalance - @cost;
END
```

### （2）存储过程 insertSell

该存储过程实现读者的零售业务，完成零售的业务和读者费用的扣除，实现逻辑与 insertBorrow 类似，需要通过读者 ID 和书刊 ID 查询读者的会员信息和书刊信息并计算产生的费用，然后添加记录，并更新读者的余额。实现过程如下。

```
CREATE DEFINER='root'@'localhost' PROCEDURE `insertSell`(IN `clientID` char(11),IN `bookID` char(13))
BEGIN
```

```

#Routine body goes here...
set @ratio = 0.00;
set @price = 0.00;
select `零售折扣` into @ratio from v_c_vip where `手机号`=`clientID`;
select bookPrice into @price from booknew where bookISBN = `bookID`;
set @cost = @ratio * @price;
INSERT INTO `bookdb`.`sell`(`clientPhone`, `bookISBN`, `buyDate`, `buyNum`) VALUES (`clientID`, `bookID`,
NOW(), 1);
UPDATE v_c_info SET clientBalance = clientBalance - @cost;
END

```

### (3) 存储过程 getBookBNum & getBookSNum

本存储过程分别对应查找规定日期范围内借阅和购买书刊的数量，该存储过程的实现相对较为简单，只需要根据输入的读者 ID、起始日期和截止日期进行查询即可，当输入的起始日期晚于截止日期，结果将返回空值。实现过程如下。

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `getBookBNum`(IN `begin` date,IN `end` date)
BEGIN
    #Routine body goes here...
    SELECT `借阅日期` borrowDate, `图书名称` bookName, count(*) borrowNum from v_b_bw_date where `借阅日期` >=begin and `借阅日期` <= end group by `图书编号`,`图书名称`;
END

CREATE DEFINER=`root`@`localhost` PROCEDURE `getBookSNum`(IN `begin` date,IN `end` date)
BEGIN
    #Routine body goes here...
    SELECT `购买日期` buyDate, `图书名称` bookName, count(*) buyNum from v_b_sell_date where `购买日期` >=begin and `购买日期` <= end group by `图书编号`,`图书名称`;
END

```

### (4) 存储过程 getClientBNum & getClientSNum

本存储过程分别对应查找指定读者借阅和购买书刊的情况，该存储过程的实现设计到了多表的查询，不仅要查询读者表 Client，还需要查询借阅表 Borrow 或零售表 Sell，为了简化查询操作，系统为此定义了视图，将读者表与其他两个表进行连接，在执行存储过程时只需执行一次对视图的查询即可，实现过程如下。

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `getClientBNum`(IN `clientID` char(11))
BEGIN
    #Routine body goes here...
    select * from v_c_b_bw_count where v_c_b_bw_count.clientPhone=`clientID`;
END

CREATE DEFINER=`root`@`localhost` PROCEDURE `getClientSNum`(IN `clientID` char(11))
BEGIN
    #Routine body goes here...
    select * from v_c_b_sell_count where v_c_b_sell_count.clientPhone=`clientID`;
END

```

## 7 测试和运行

经过前期的需求分析、数据库设计和应用程序设计和开发，本系统实现了设计阶段的功能模块，满足了需求分析阶段读者客户和管理员的使用需求。下面将介绍本系统的测试和运行过程，以此来展示本系统的功能模块的实现情况，着重体现我在项目开发过程中设计和实现的部分。

### 7.1 借阅管理模块的测试

本模块主要划分为借阅书刊、归还书刊、续借书刊、预约书刊和借阅查询，我主要负责设计和实现借阅书刊和借阅查询。

#### 1. 借阅书刊子模块的测试

通过左侧选项栏点击“常用操作”，如图 11 所示，选择图书借阅即可进入借阅书刊的模块中。该模块主要分为上下两部分，上方用于查询读者信息，下方用于查询书刊信息。

图 11 借阅书刊子模块

在查询会员中输入客户的手机号即可查到客户的信息，执行查询后会通过嵌入式 SQL 语句查询客户表 Client，若不存在则会如图 12 所示，进行提示。

图 12 查询不存在的客户信息

当查询到了客户的会员信息，则会在下方展示，且上方的确认按钮锁定，下方的按钮解锁，如图 13 所示。

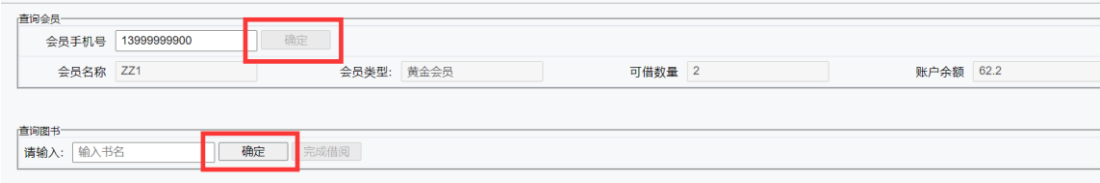
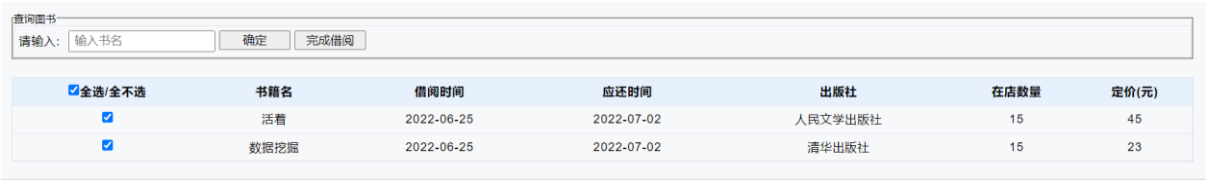


图 13 完成会员查询

此时在下方可以进行书刊的查询，类似于客户信息的查询，若书刊不存在也会给出提示，如图 14 所示，当输入正确的书刊名称，下方便会显示书刊的信息，如图 15 所示，其中包括借阅的时间、应还的时间、在馆的数量等信息，可以进行选择借阅，最后点击“完成借阅”即可实现借阅的操作，系统会给出借阅成功的提示。



图 14 查询不存在的书刊



<input checked="" type="checkbox"/> 全选/全不选	书籍名	借阅时间	应还时间	出版社	在店数量	定价(元)
<input checked="" type="checkbox"/>	活着	2022-06-25	2022-07-02	人民文学出版社	15	45
<input checked="" type="checkbox"/>	数据挖掘	2022-06-25	2022-07-02	清华大学出版社	15	23

图 15 显示书刊的信息

## 2. 借阅查询子模块的测试

通过左侧选项栏点击“信息查询”，选择借阅历史记录即可进入借阅查询的模块中。该部分支持多种条件的查询，且能实现关键字模糊查询，如图 16 所示。

若不输入关键字，默认查询条件选择“全部”，点击“搜索”后即可获取所有的客户借阅信息，包括联系方式、会员名称、书刊名称、借阅时间、归还时间、到期时间等信息，若图书尚未归还，则归还时间将为空。

查询条件: <input checked="" type="radio"/> 全部 <input type="radio"/> 已归还 <input type="radio"/> 未归还 <input type="radio"/> 最近一周需归还           请输入关键字: <input type="text"/> <input type="button" value="搜索"/>					
联系方式	会员名称	书籍名称	借阅时间	归还时间	到期时间
14790409310	张一2	数据挖掘	2022-06-16	2022-06-16	2022-07-16
18856316698	橙子	数据挖掘	2022-06-16	2022-06-17	2022-07-16
13616644404	super	活着	2022-06-17		2022-08-16
13999999900	ZZ1	活着	2022-06-17	2022-06-17	2022-08-16
13999999955	zzzz	活着	2022-06-17		2022-07-17
14790409310	张一2	活着	2022-06-16	2022-06-16	2022-07-16
18856316698	橙子	活着	2022-06-16	2022-06-17	2022-07-16

2019217192:周布伟, 2019218139:朱紫晨

图 16 借阅查询子模块

切换查询条件，不输入任何关键字，点击“搜索”，结果如图 17 所示，展示了所有已经归还书刊的借阅信息，类似的，可以更换条件，查询未归还、最近一周需归还的信息。该功能本质上是在执行 SQL 语句时增加约束条件，即可实现预期的功能。

<input type="radio"/> 全部 <input checked="" type="radio"/> 已归还 <input type="radio"/> 未归还 <input type="radio"/> 最近一周需归还           请输入关键字: <input type="text"/> <input type="button" value="搜索"/>					
联系方式	会员名称	书籍名称	借阅时间	归还时间	到期时间
14790409310	张一2	数据挖掘	2022-06-16	2022-06-16	2022-07-16
18856316698	橙子	数据挖掘	2022-06-16	2022-06-17	2022-07-16
13999999900	ZZ1	活着	2022-06-17	2022-06-17	2022-08-16
14790409310	张一2	活着	2022-06-16	2022-06-16	2022-07-16
18856316698	橙子	活着	2022-06-16	2022-06-17	2022-07-16

图 17 查询已归还的借阅信息

在关键字输入框中输入读者的手机号或名称，可以通过模糊查询的方式找到用户的借阅信息，如图 19 所示，若信息不存在也会给出提示。

<input checked="" type="radio"/> 全部 <input type="radio"/> 已归还 <input type="radio"/> 未归还 <input type="radio"/> 最近一周需归还           请输入关键字: <input type="text" value="14"/> <input type="button" value="搜索"/>					
联系方式	会员名称	书籍名称	借阅时间	归还时间	
14790409310	张一2	数据挖掘	2022-06-16	2022-06-16	
14790409310	张一2	活着	2022-06-16	2022-06-16	

<input checked="" type="radio"/> 全部 <input type="radio"/> 已归还 <input type="radio"/> 未归还 <input type="radio"/> 最近一周需归还           请输入关键字: <input type="text" value="子"/> <input type="button" value="搜索"/>					
联系方式	会员名称	书籍名称	借阅时间	归还时间	
18856316698	橙子	数据挖掘	2022-06-16	2022-06-17	
18856316698	橙子	活着	2022-06-16	2022-06-17	

图 18 通过手机号或会员名称模糊查询借阅信息



图 19 查询的信息不存在

此外，还可以在通过左侧选择借阅时间段信息，输入日期，即可查询到该时间范围内的借阅信息。

请输入起始时间: 2022-06-13 请输入终止时间: 2022-06-17 搜索		
借阅时间	书籍名称	借阅数量
2022-06-16	数据挖掘	2
2022-06-17	活着	5

图 20 查询指定时间范围内的借阅信息

## 7.2 零售管理模块的测试

本模块主要划分为购买数控和零售查询两个子模块。

### 1. 购买书刊子模块的测试

本子模块的设计实现思路与借阅模块类似，受限于篇幅，在此不再赘述，与借阅模块的主要区别在于前端展示的信息不同，业务逻辑实现也不同，经过测试，该模块能够完成零售的功能，且并未出现错误。

### 2. 零售查询子模块的测试

该子模块的实现和测试过程与借阅查询模块类似，同样支持关键字模糊查询，但并未加入如同借阅的条件约束，如图 21 所示展示了客户零售的信息。

请输入关键字: 搜索			
联系方式	会员名称	书籍名称	销售时间
14790409310	张一2	数据挖掘	2022-06-16
18856316698	橙子	数据挖掘	2022-06-16
14790409310	张一2	活着	2022-06-16
13616644404	super	数据挖掘	2022-06-17
13999999900	ZZ1	活着	2022-06-17

图 21 借阅查询子模块

### 7.3 会员管理模块的测试

本模块分为类型定义、信息修改和类型删除。

#### 1. 类型定义模块的测试

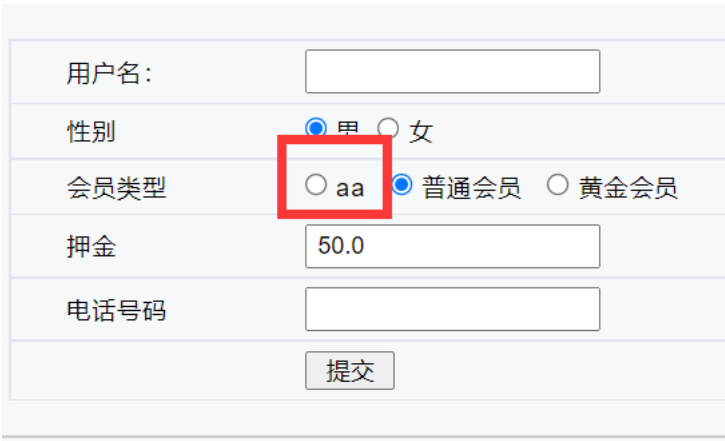
在左侧选项栏中选择“会员类型管理”，点击“添加类型”即可进入会员类型的添加页面，输入相应的信息即可完成会员类型的添加，如图 22 所示，点击提交后，若其中的内容出现出现负数或其他违反完整性约束的情况，则会激活触发器，导致会员类型的添加失败。



会员名称	aa
押金	50.0
借阅折扣	0.05
销售折扣	1.0
借阅数量	3
续借数量	2
借阅日期	30
预约时长	7
逾期罚款额度	0.01
损坏罚款额度	1.0
丢失罚款额度	1.0
<input type="button" value="提交"/>	

图 22 会员类型的添加

若会员添加成功，如图 23 所示，在客户开卡和信息修改处也会加入该会员名称。



用户名:	
性别	<input checked="" type="radio"/> 男 <input type="radio"/> 女
会员类型	<input type="radio"/> aa <input checked="" type="radio"/> 普通会员 <input type="radio"/> 黄金会员
押金	50.0
电话号码	
<input type="button" value="提交"/>	

图 23 新增定义的会员类型



## 2. 信息修改模块的测试

在会员列表中点击修改，系统会确认是否执行修改操作，点击确定后即可进入修改页面，修改后进行提交，若不违反完整性约束，则修改成功，结果如图 24 所示。

会员名称	押金	出租折扣	销售折扣	可借阅数量	可续借次数	可借阅天数	预约保留时间	逾期扣费额度	损坏扣费额度	丢失扣费额度	
普通会员	50.0	0.05	1.0	3	2	30	7	0.01	1.0	1.0	修改 删除
aa	100.0	0.59	0.88	7	2	30	10	1.05	1.05	1.05	修改 删除
黄金会员	200.0	0.03	0.76	3	2	60	7	0.01	1.0	1.0	修改 删除

图 24 修改会员信息

## 3. 类型删除模块的测试

由于会员信息与其他用户牵连，直接删除会导致用户表的外码为空，影响正常的业务实现，故当客户中有该会员类型，如图 25 所示，则不能删除，反之，会进行假删除，会员类型的状态改变，前端不予以显示该会员类型。



图 25 含客户信息，无法删除会员类型

# 7.4 客户管理模块的测试

本模块主要分为客户开卡、客户注销、余额充值、余额提现和信息修改等子模块。

## 1. 客户开卡模块的测试

在本子模块中输入要求的信息，如图 26 所示，选择会员类型，不同的会员类型会展示不同的押金要求，且押金字段锁定，无法输入，只会随着会员类型的选择而变化，点击提交即可开卡成功，由于在一些查询中往往使用用户的手机号作为查询条件，故在此限制手机号唯一，如重复则创建失败，新添加的客户余额为 0。

用户名:	<input type="text" value="aaa"/>
性别	<input type="radio"/> 男 <input checked="" type="radio"/> 女
会员类型	<input type="radio"/> 普通会员 <input checked="" type="radio"/> 黄金会员
押金	<input type="text" value="200.0"/>
电话号码	<input type="text" value="13616644400"/>
	<input type="button" value="提交"/>

图 26 客户开卡

## 2. 客户注销模块的测试

由于客户的信息与借阅、零售等记录牵连，故无法实现真正的删除操作，改变客户的状态，限制客户的所有操作。若客户含有借阅记录，如图 28 所示，则注销失败。

用户名	联系方式	性别	会员等级	余额	状态	借阅数量	操作
super	13616644404	男	黄金会员	-6.4	正常	1	修改 注销
zbw	13811199220	男	普通会员	-37.8	注销	0	修改 注销
ZZZZ1	13999999333	男	黄金会员	0.0	注销	0	修改 注销
ZZ1	13999999900	男	黄金会员	62.2	正常	1	修改 注销
zzzz	13999999955	男	普通会员	97.75	正常	1	修改 注销
张一2	14790409310	女	普通会员	77.18	正常	2	修改 注销
橙子	18856316698	女	普通会员	59.1	正常	2	修改 注销

图 27 客户列表



图 28 含有借阅记录，注销失败

## 3. 余额充值与提现模块的测试

本子模块只需要填写客户的手机号和金额即可完成充值和提现，如果金额为正数则为充值，为负数即为提现，操作的过程和结果如图 29 所示。

13999999900	男	黄金会员	62.2
13999999955	男	普通会员	97.75
14790409310	女	普通会员	77.18
18856316698	女	普通会员	59.1

手机号码:

14790409310

金额:

100

提交

13999999900	男	黄金会员	62.2
13999999955	男	普通会员	97.75
14790409310	女	普通会员	177.18
18856316698	女	普通会员	59.1

图 29 充值的实现过程

当提现的金额超过客户的余额时，从实际应用场景来看，应该出错，但是在测试时能够正常提现，故需要重新修改，通过添加触发器限制提现的金额不能超过余额，回归

测试后结果如图 30 所示，该操作将不被执行。



图 30 提现金额大于余额，操作失败

#### 4. 信息修改模块的测试

在用户列表中点击“修改”，系统会提示是否确认修改，确认即可进入修改界面，界面中可以更改客户的名称、性别和会员类型，提交后即可完成修改。

### 7.5 其它模块的测试

本节已经将关键模块的测试通过图文的形式予以展示，由于本系统的功能较多，囿于篇幅，部分简单的测试点在此不再赘述，所有的测试点和测试结果如表 10 所示，测试结果表明，大部分模块能够完成预期的工作，少部分模块经过测试发现了存在的问题，经过改进后也达到了预期的效果，最后通过回归测试，所有的模块都实现了相应的功能，满足了设计要求。

表 10 系统测试结果汇总

主模块	子模块	测试点	结果
书刊管理	借阅书刊	客户查询、书籍查询、 借阅操作	能够根据客户和书籍是否存在给出提示信息，实现借阅操作
	借阅查询	条件限制、关键字查找	能够依据条件和关键字实现模糊查找
借阅管理	购买书刊	客户查询、书籍查询、 零售操作	能够根据客户和书籍是否存在给出提示信息，实现零售操作
	零售查询	关键字查找	能够依据条件和关键字实现模糊查找
零售管理	类型定义	添加信息、完整性约束	完成信息添加，有完整性约束，客户字段中增加会员信息
	信息修改	修改信息、完整性约束	完成信息修改，有完整性约束
	类型删除	包含类型和不包含类型	成功，能够给出相应的提示信息，实现假删除

会员管理	客户开卡	添加客户信息	在完整性约束下添加成功
	客户注销	删除会员信息	能够依据有无余额和借阅记录进行假删除
	余额充值和提现	充值和提现余额	当提现金额超过余额时出现问题，修改后测试正确
	信息修改	修改客户信息	在完整性约束下修改成功
其他	数据备份	备份数据库内容	成功，能备份到指定路径
	数据恢复	恢复数据库内容	成功，数据库内容能够导入
	密码修改	修改管理员密码	成功，密码以密文保存数据库

## 8 总结

在本次《数据库系统课程设计》中，我们设计并开发实现了书店书刊出租和零售管理系统，完成了需求分析、数据库设计、应用程序设计和运行测试，并取得了预期的效果。本次设计开发实现的书店书刊出租和零售管理系统能够支持书店管理员和客户的多项事务，包括书刊管理、借阅管理、零售管理、会员管理、客户管理等，能够形成业务闭环，满足书店管理员和客户的需求，提高了书店管理员和读者客户的业务执行效率。

本次课程设计以小组的形式开展，在小组的开发过程中，我作为组长，需要确定开发计划和分工，协调开发进度，在数据库设计中，我负责设计 ER 图，调整关系模式，设计存储过程，定义部分触发器。在应用程序开发中，我负责客户管理、会员类型管理、图书借阅、零售及其查询等模块的设计与实现。

通过本次课程设计，我锻炼了团队开发的能力，积累了团队协作的经验，在数据库的设计开发过程中，我们对数据库的规范化、完整性、安全性等要求有了更深刻的认识和体会，意识到保证系统的高效性和正确性需要经过不断的检验和测试。通过将课内讲授的数据库相关知识付诸实践，我们不仅加深了课内知识点的印象，而且能够将知识点融会贯通。

尽管本系统在功能上已经有了一定的雏形，但是还有很大的改进空间。由于课程设计时间有限，我们很难考虑到所有系统测试的用例，故在系统测试上还存在尚未发现的问题。在模块的设计上，系统在未来还可以加入更多的客户行为分析，基于客户产生的借阅与购买历史统计数据，计算统计量，如最大值、最小值、均值、方差等，便于书店管理员对书刊的数量进行有效的决策。

课程设计的最后，感谢胡敏老师，对本系统提出宝贵的意见，让我们意识到需求分析的方法和重要性，也让我们明白本系统在完整性设置上可能存在的不足之处，引导我们解决相关的问题。感谢沈明玉老师，对《数据库系统》理论课程的讲授，没有理论课程的知识我们很难着手完成数据库系统的设计，让我们意识到数据库设计的关键问题，使得我们在开发过程中少走了很多弯路。感谢朱紫晨同学，在数据库设计过程中将 ER 图转换成关系模式，使用 MySQL 创建基本表和视图，定义部分触发器，完整性约束实现。完成了图书管理、图书归还的开发，为系统的设计和开发分担了压力。

依托本次课程设计，希望自己在未来能够设计开发出功能更加完备、各项性能更加优越、用户体验更加友好的软件系统。

## 附录 参考文献

- [1] 王珊, 萨师煊. 数据库系统概论 第 5 版[M]. 北京: 高等教育出版社. 2014.
- [2] 张海藩, 牟永敏. 软件工程导论 第 6 版[M]. 北京: 清华大学出版社. 2013.
- [3] 赵芳. 基于 ASP.NET 书店管理系统的设计与实现[D]. 电子科技大学, 2014.
- [4] 赵智凤. 基于 J2EE 的网上书店系统的设计与实现[D]. 吉林大学, 2018.
- [5] 郑梁君. 基于 J2EE 网上书店系统的设计与实现[D]. 南昌大学, 2012.