《数据库系统》课程设计报告

|  |  |  |  |
| --- | --- | --- | --- |
| **题目** | 图片销售管理系统 | | |
| 小组成员信息 | | | |
| **姓名** | **学号** | **班级** | **分工** |
| **陈欣宇** | 21307347 | 人工智能与大数据 | 框架初始化，编写代码报告 |
| **高宇** | 21307350 | 人工智能与大数据 | ER图初设计，编写代码报告 |
| **陈华清** | 21307100 | 人工智能与大数据 | 数据初始化，编写代码报告 |

提交时间：2023年12月31日

1. 开发环境与开发工具

-开发环境：windows

-开发工具：jdk1.8，Mysql8.0.31，Redis5.0.14.1，Maven3.6，Node16.19.1，IDEA2023.1，VS Code1.85.1

-客户端界面：使用前后端分离的[若依框架](https://gitee.com/y_project/RuoYi-Vue)搭建Web界面

-数据库使用内网穿透(贝锐花生壳)映射一台主机端口进行协作使用

1. 系统需求分析(5分)

此次课程设计实现以下功能：

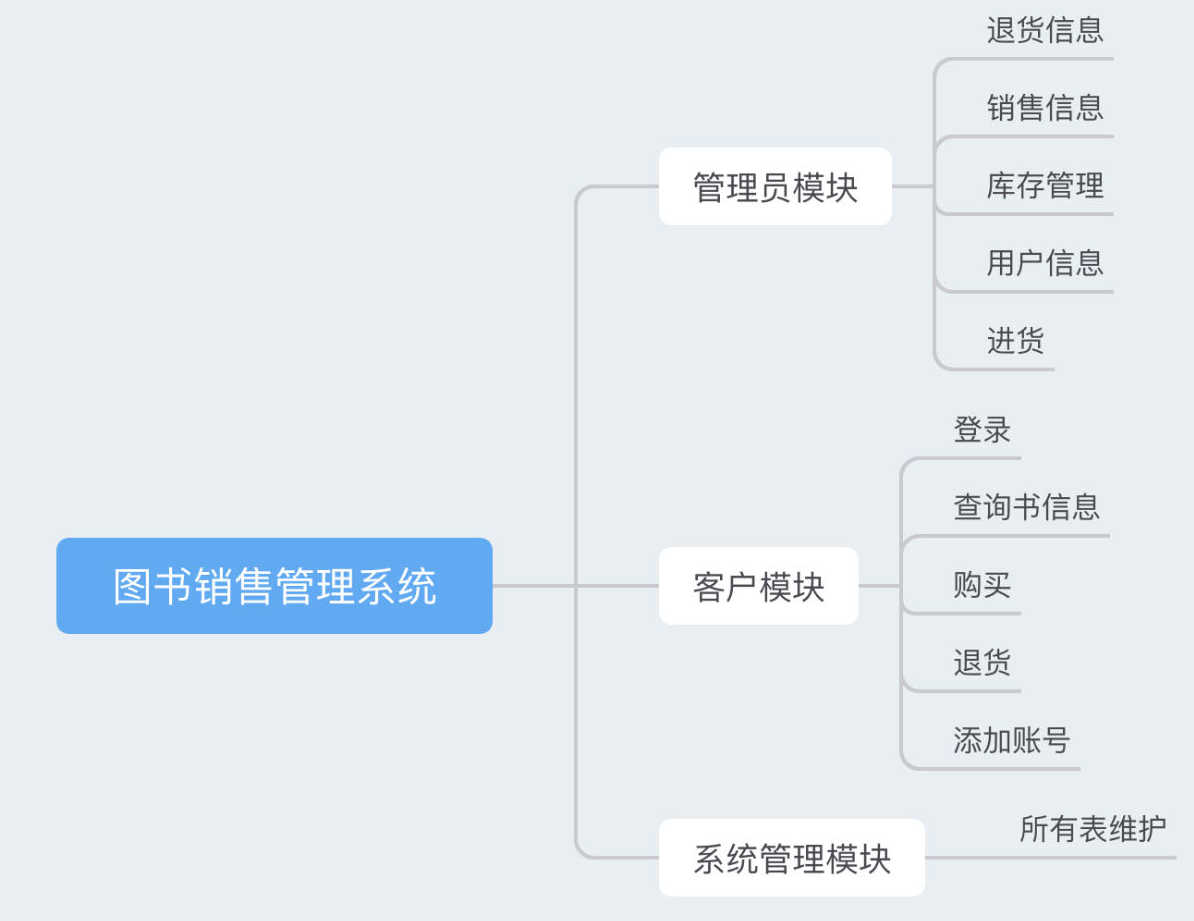
1. 进货:根据某种书籍的库存量及销售情况确定进货数量,根据供应商报价选择供应商。输出一份进货单并自动修改库存量,把本次进货的信息添加到进货库中。
2. 退货:顾客把已买的书籍退还给书店。输出一份退货单并自动修改库存量,把本次退货的信息添加到退货单中。
3. 销售:顾客搜索买书籍的信息，显示此书的库存量。得到销售记录并修改库存，同时把此次销售记录添加到销售表中。

系统数据字典：根据以上需求，得到以下系统的数据结构

|  |  |
| --- | --- |
| 数据结构名 | 属性 |
| 图书book | 书号，书名，出版社 |
| 库存stock | 书号，价格，库存量 |
| 可进货表sales | 书号，进货单号，商家，商家库存，单价 |
| 进货单supply | 书号，进货单号，进货数量，进货时间 |
| 销售记录purchase | 书号，客户号，销售数量，是否退货，销售时间 |
| 客户consumer | 客户号，姓名，电话，地址，性别 |

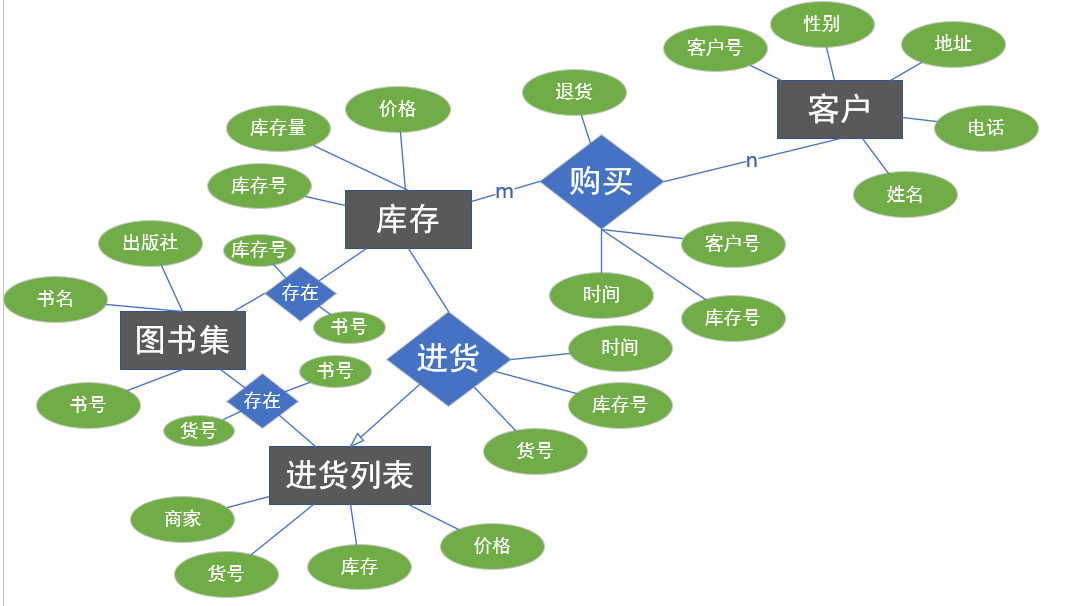
1. 功能需求分析（10分）

系统功能模块图



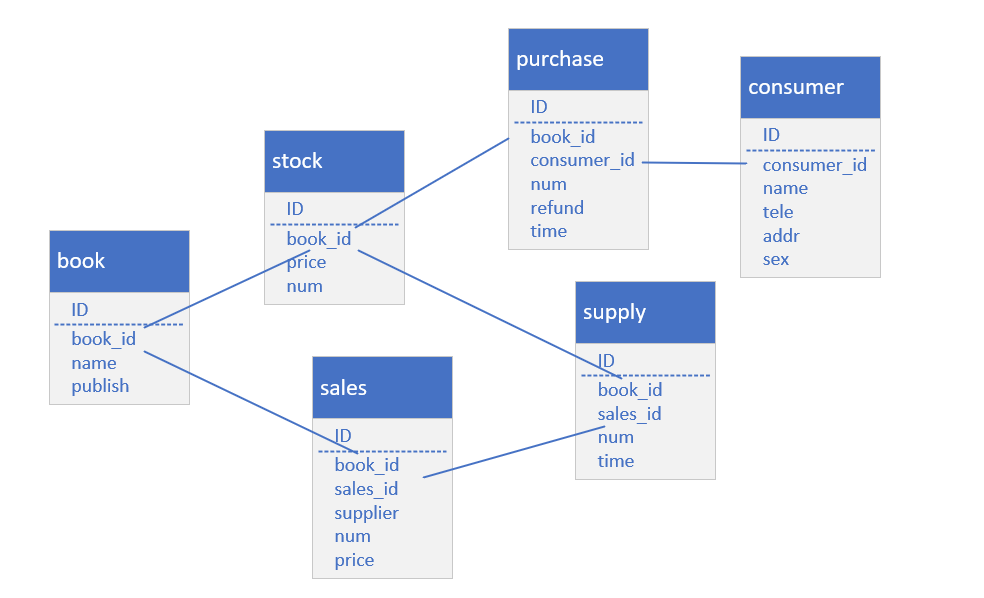
1. 系统设计（25分）

数据概念结构设计（系统ER图）（10分）



数据库关系模式设计（10分）

根据具体实现功能对ER图涉及字段进行简化，因为若依生成表要求id字段自动递增，故每个表中额外使用一个id字段



数据库物理结构设计（5分）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| book | ID | Book\_id | Name | Publish |
|  | Int | Char(5) (unique) | Varchar(45) | Varchar(45) |
| 解释 |  | 书号 | 书名 | 出版社 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stock | ID | Book\_id | Price | Num |
|  | Int | Char(5) (unique) | Float | int |
| 解释 |  | 书号 | 价格 | 库存量 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sales | ID | Book\_id | Sales\_id | Supplier | num | price |
|  | Int | Char(5) | Char(5) (unique) | Char(30) | Int | float |
| 解释 |  | 书号 | 单号 | 商家 | 商家库存 | 单价 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Purchase | ID | Book\_id | consumer\_id | num | refund | time |
|  | Int | Char(5) | Char(5) | Int | Char(5) | datetime |
| 解释 |  | 书号 | 客户号 | 销售数量 | 是否退货 | 销售时间 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Supply | ID | Book\_id | Sales\_id | Num | time |
|  | Int | Char(5) | Char(5) | Int | datetime |
| 解释 |  | 书号 | 单号 | 入货数量 | 入货时间 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Consumer | ID | Consumer\_id | Name | Tele | Addr | sex |
|  | Int | Char(5) (unique) | char(20) | Char(20) | Varchar(45) | Char(5) |
| 解释 |  | 客户号 | 姓名 | 电话 | 地址 | 性别 |

1. 系统功能的实现 (10分)

主要功能模块的实现过程（简述）、运行界面

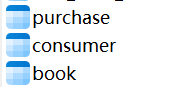
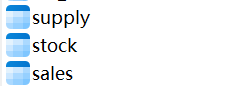
1. 初始化框架

首先下载[若依框架](https://gitee.com/y_project/RuoYi-Vue)于本地，具体搭建运行过程不详细描述。

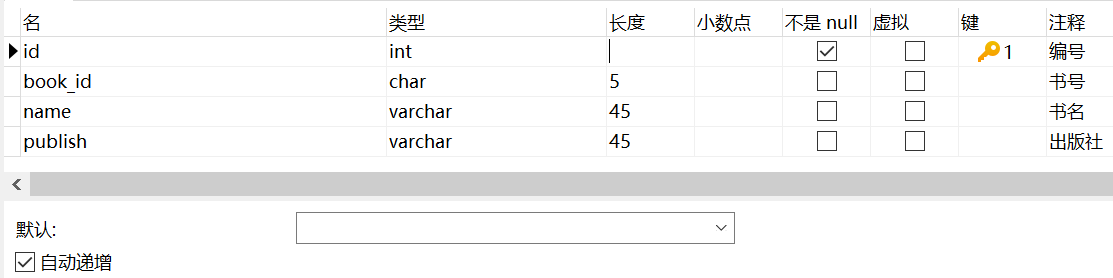
数据库：本地创建数据库tushu，创建用户tushu供访问，在贝锐花生壳得到本地3306的映射网址845719j0o5.goho.co端口57654，其他主机即可访问该数据库



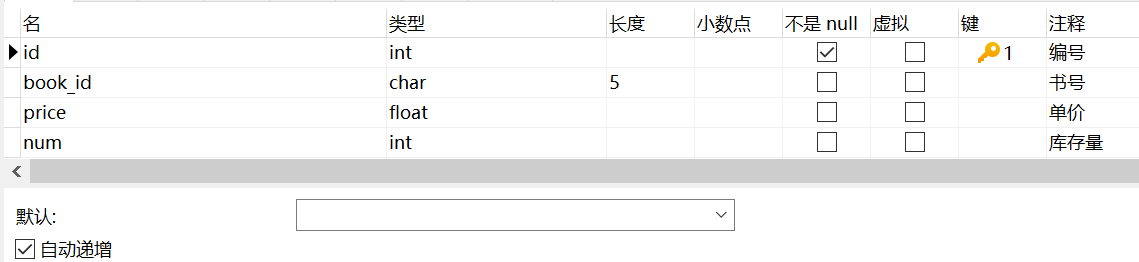
向初始化框架的数据库中新建数据表



表book：



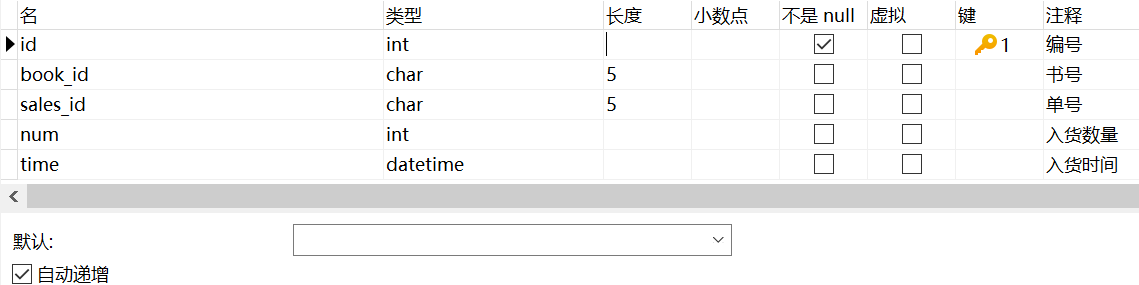
表stock：



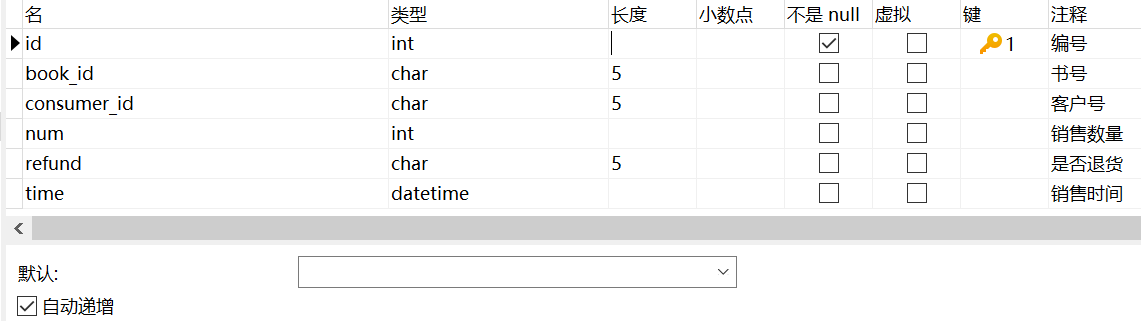
表sales：



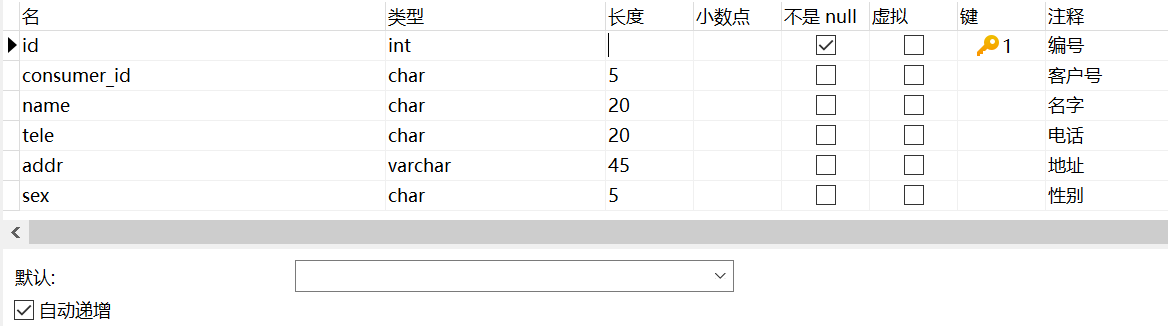
表supply：



表purchase：



表consumer：



1. Web界面生成表

使用代码生成功能将所有表的代码生成，如图将表导入->编辑->生成代码，将代码放入文件包中，





Web界面生成表效果：



在生成表之后，通过修改对应表的Mapper.xml和domain文件，以及修改vue文件的排版，能够实现多表关联查询，如下图，库存表stock中能够根据“书号”显示“书名”和“出版社”的信息，在这也不展开描述具体细节。管理员**库存管理**或是客户**查询书本信息**的功能也顺带实现。



1. 进货功能

在介绍如何实现之前首先说明本系统本身具备的功能。

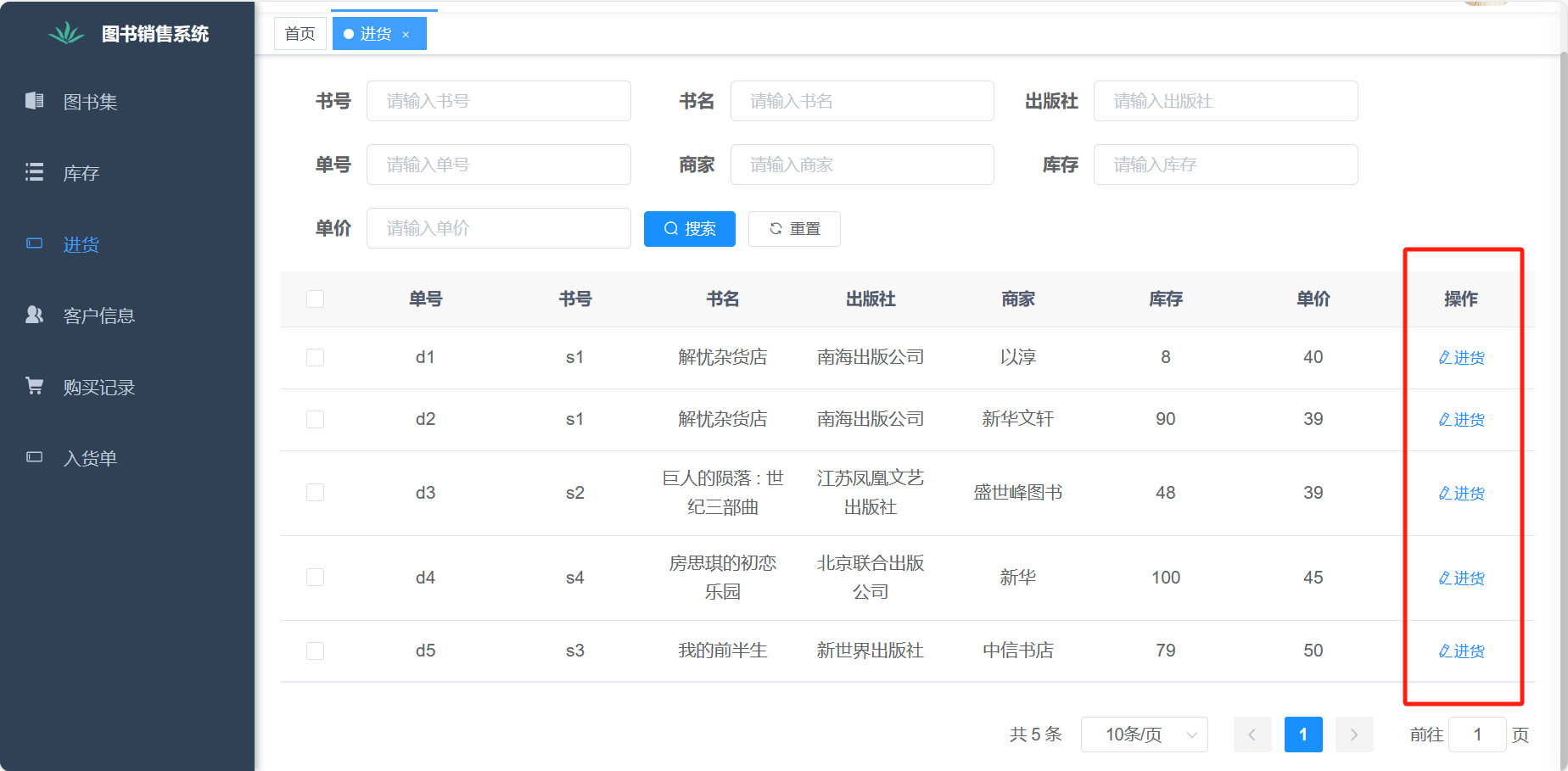
* 1. 角色管理：能够创建角色，本次设置了图书管理员和客户两个角色，用于权限判定以及界面区分



* 1. 用户管理：创建用户账号，账号能够指定角色，使得不同账号登录看到的界面不同，在同一界面上看到的按钮不同。



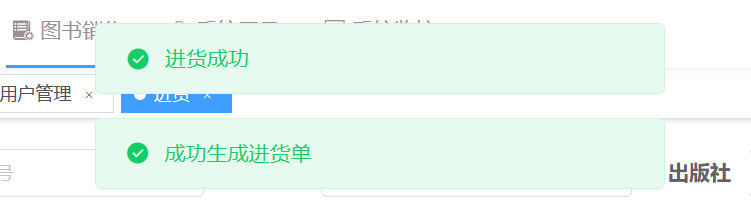
正式说明进货功能实现：效果图如下，通过管理员角色登入进到“进货”目录下，原先的“修改”“删除”按钮被“进货”按钮替代



点击“进货”弹出对话框：通过滑动框选择进货数量



确认入货后提示成功：并在supply表中插入对应的进货记录



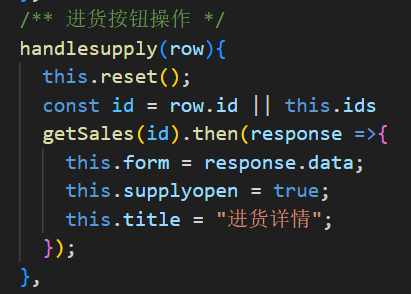
查看入货单：观察到入货单更新入货条目，库存表中对应书条目库存量也增加



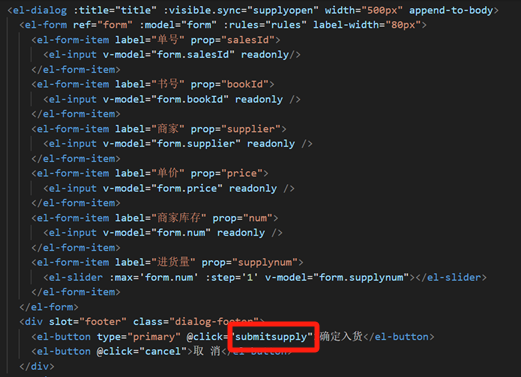
实现过程：主要操作该目录sales的前端vue文件：根据定义的character角色判断是否显示该按钮，以及定义handlesupply为“进货”按钮触发函数



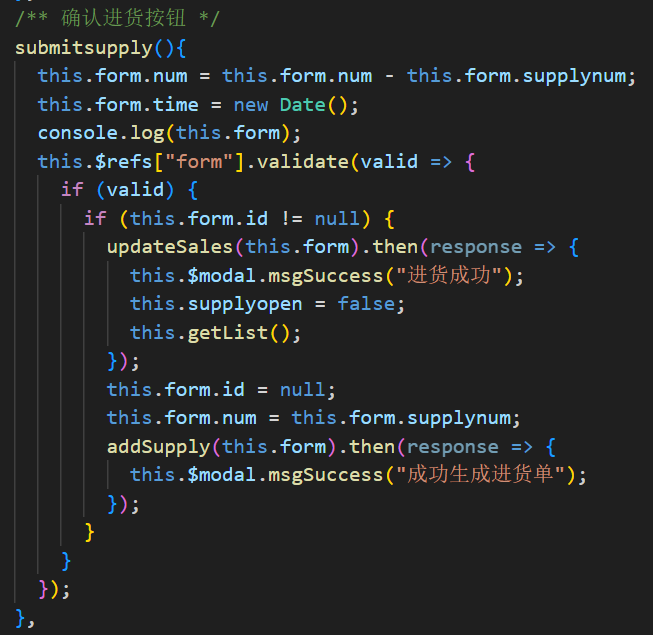
进货按钮函数通过supplyopen=true打开弹出



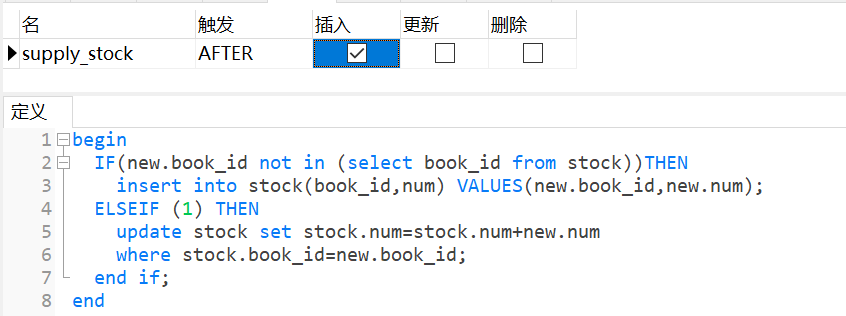
弹窗内容：将选择入货量赋值给form.supplynum，最后调用submitsupply确认入货



Submitsupply函数实现：将form.num也就是sales表对应行的库存量降掉前面赋值的form.supplynum，定义form.time=new Date()得到当前时间，调用updateSales更新可进货表，调用addSupply将得到的进货信息插入supply表中。



库存量增加实现使用了触发器，当入货表插入新入货单时，库存表进行更新，为supply定义触发器如下：



至此入货功能实现，效果已进行展示。

1. 销售功能

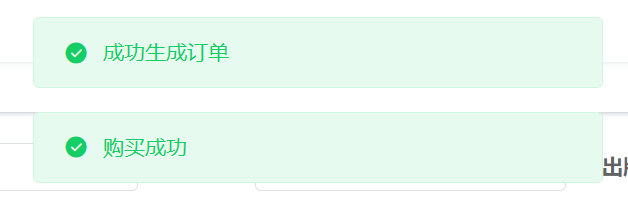
销售功能对应了客户的购买操作，主要在库存表stock上实现同入货功能相似操作，先展示效果图，以下为客户界面



点击购买弹出窗口，选择购买数量，确认购买



购买成功提示



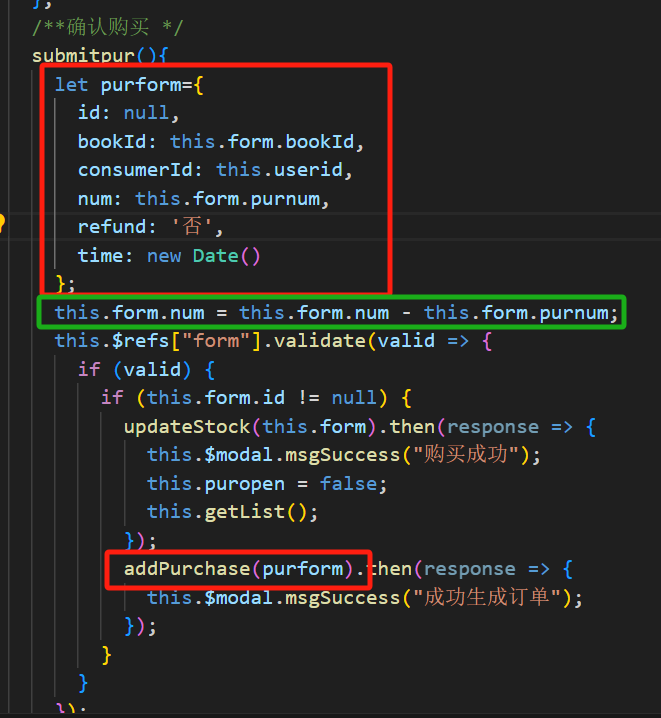
生成购买记录：对应库存中图书的库存量下降



实现过程：在目录stock的前端vue文件进行编写代码，以上文同样方式，根据用户角色为“客户”，得到该只存在“购买”按钮的界面，调用handlepur函数弹出窗口，窗口实现如下



通过submitpur确认购买，定义一个purform类型，存储将要插入purchase表的信息，调用addPurchase将本次购买记录插入purchase表，而对于此处库存减掉的实现，因为正是在本stock表操作，不需要触发器，直接将num减掉对应购买量purnum调用updateStock即可。

至此购买功能实现结束

1. 退货功能

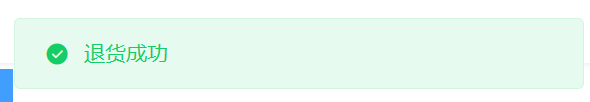
同上先看效果图：“客户”在购买记录中点击“退货”



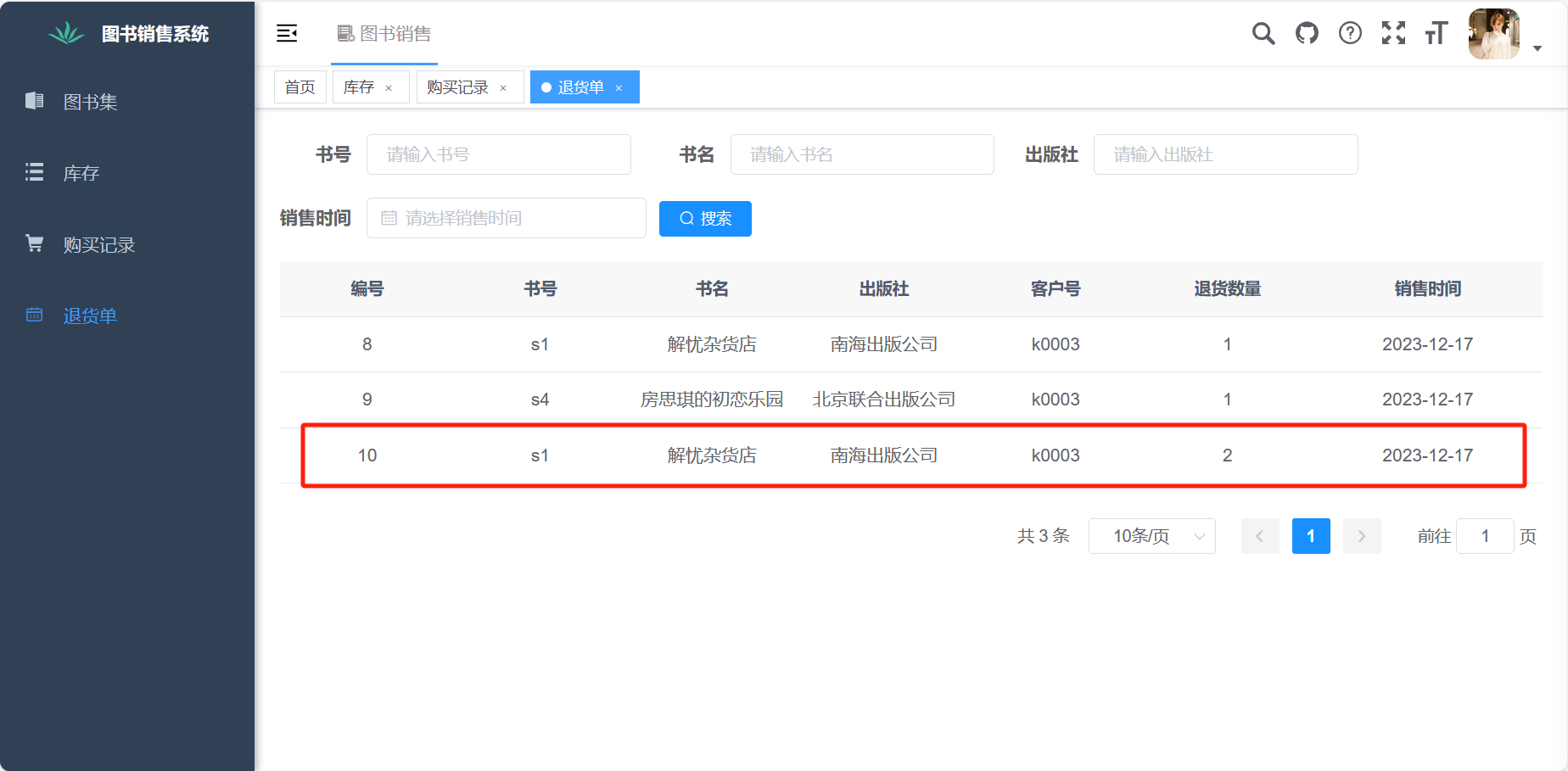
弹出窗口显示退货信息，点击“确认退货”



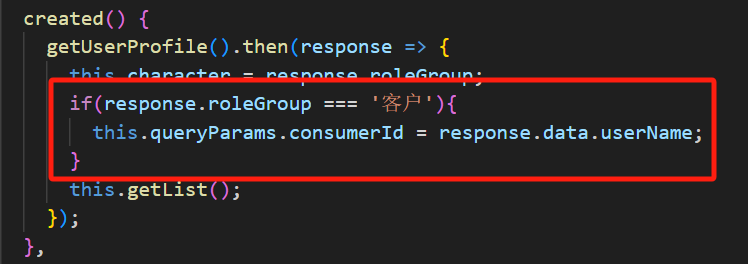
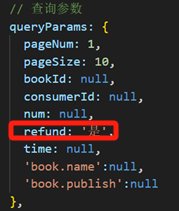
提示退货成功



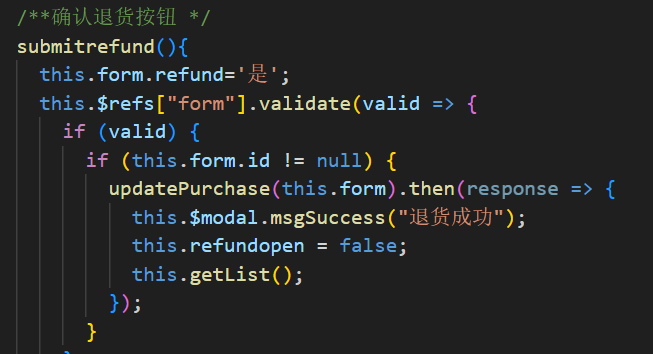
然后就可以在退货单中查找到退货记录



实现过程：首先说明此处退货单和购买记录同用一张purchase表，根据前端vue生成两个菜单文件，在web界面初始化时将查询参数的refund是否退货置为“是”，即可把以及退货的购买记录筛选为退货记录。以及对于“管理员”和“客户”看到同一个web界面，但“客户”只需要看到自己的购买记录，对此同理对查询参数进行初始化，如果角色为“客户”则将筛选客户号为该客户账号的记录。



对于退货操作，实现原理比进货和购买相同但更简单，只需要将字段refund改为“是”即可，同样的思路，弹出窗口，点击确认调用了以下更新purchase表操作



对于库存量增加的实现同样使用到了触发器，检测purchase的update，当有退货操作则将stock.num对应加上purchase的num。



六. 总结

本课程设计中使用到了基本mysql的语法如建表过程的create table。在生成代码中，包含了最基本的select、insert、update、delete操作；在多表关联查询中使用了left join拼接表；

使用触发器解决库存量改变的问题等理论课的概念与知识。