

## 《数据库系统》课程设计报告

题目	图书销售管理系统		
小组成员信息			
姓名	学号	班级	分工
王明业	17343107	17 级软件工程 教务二班	各项数据表的构建、 逻辑的构建、 进货模块的实现。 实验报告一.二.三.四.
王子豪	17343110	17 级软件工程 教务二班	所有底层模块的实现、 退货、销售、统计模块的实现、 所有代码的测试、 实验报告五.系统功能的实现

提交时间： 2019 年 6 月 23 日

### 一. 开发环境与开发工具

MySQL 版本：

mysql Ver 8.0.16 for Win64 on x86\_64 (MySQL Community  
Server - GPL)

Connector 版本：

Python3 中的 pymysql 库

### 二. 系统需求分析

系统数据字典：

Book

字段	Book_id	Book_name	Quantity	Sell_Price
描述	书号	书名	库存数量	售价
数据类型	char(8)	varchar	int	float
约束	Primary key	Not null	Default 0	Not null

### Supplier

字段	Supplier_id	Supplier_name	Contact
描述	供应商号	供应商名	联系方式
数据类型	char(8)	varchar	char(8)
约束	Primary key	Not null	

### Quote

字段	Supplier_id	Book_id	Price
描述	供应商号	书号	单价
数据类型	char(8)	char(8)	float
约束	References Supplier(Supplier_id)	References Book(Book_id)	Not null

### Bookshop\_Purchase\_Log

字段	Purchase_id	Purchase_time	Supplier_id	Book_id	Price	Amount
描述	进货单号	进货时间	供应商号	书号	单价	数量
数据类型	int	datetime	char(8)	char(8)	float	int
约束	Primary key	Default now()	Not null	Not null	Not null	Not null

	Auto increment					
--	-------------------	--	--	--	--	--

Customer\_Return\_Log

字段	Return_id	Return_time	Book _name	Price	Amount
描述	退货单号	退货时间	书名	当前 单价	数量
数据类型	int	datetime	varchar	float	int
约束	Primary key  Auto increment	Default  now()	Not  null	Not  NULL	Not null

这里的 **Price** 表示的是当前该图书的单价，即顾客退货是按照当前的图书单价退货（而非购买时的单价）。而在此记录当前单价的原因是为了以后能准确计算销售额（因为 **Book** 表中的单价在以后可能改变）。

Bookshop\_Sell\_Log

字段	Sell_id	Sell_time	Book_name	Price	Amount
描述	销售单号	销售时间	书号名	当前 单价	数量
数据类型	int	datetime	varchar	float	int
约束	Primary key Auto increment	Default now()	Not null	Not null	Not null

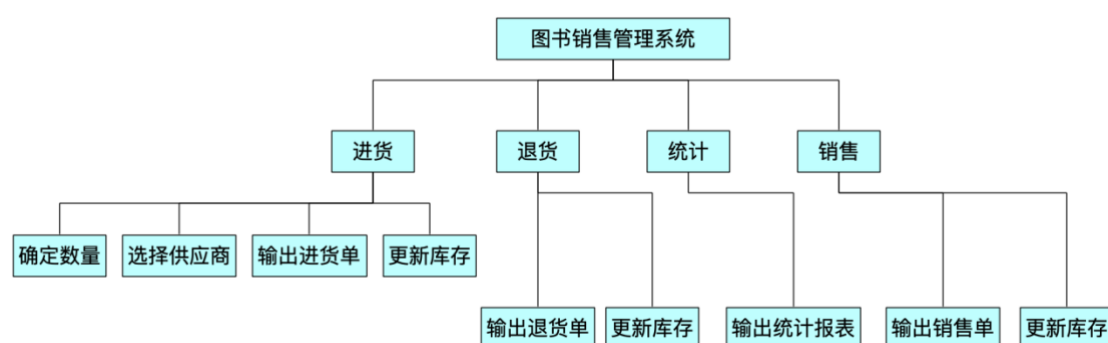
这里的 Price 同样是记录当前单价，以防 Book 表中单价改变。

关于是否要维护一个顾客数据表：

我的个人想法是：书店每天人流量大，顾客众多，且购买图书的顾客也不一定会留下个人信息。书店没有很大必要来维护一个记录顾客信息的数据表。

### 三. 功能需求分析

（使用 Visual Paradigm 绘图，若显示不清，请放大查看）



**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。

**退货：**顾客把已买的书籍退还给书店。输出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库中。

**统计：**根据销售情况输出统计的报表。一般内容为每月的销售总额、销售总量及排行榜。

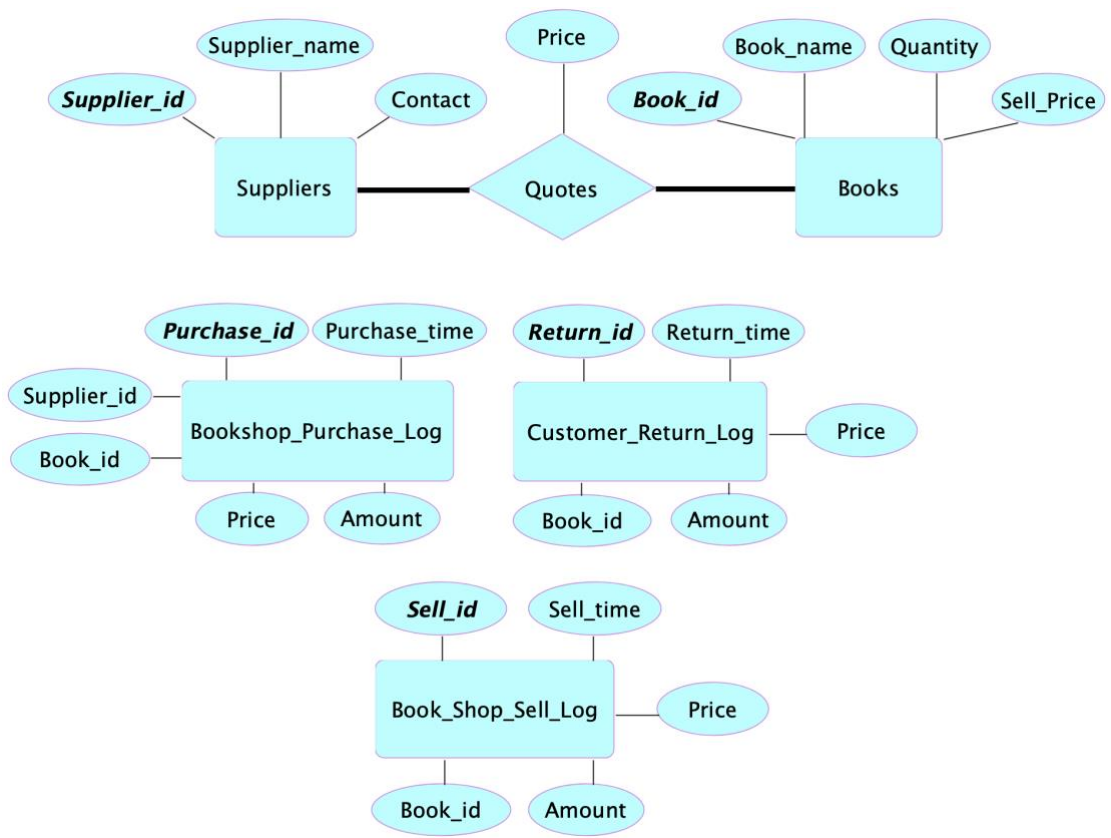
**销售：**输入顾客要买书籍的信息，自动显示此书的库存量，如果

可以销售。打印销售单并修改库存，同时把此次销售的有关信息添加到日销售库中。

四. 系统设计（10 分）

ER 图：

（使用 Visual Paradigm 绘图，若显示不清，请放大查看）



由于软件功能限制，这里使用粗斜体代替下划线表示主码。

正如先前数据字典中所显示的，供应商和书本两个实体间有报价关系，且每个供应商都有一些书的报价，每种书也都会有供应商提供报价。

而记录进货、退货和出售的三个日志单独成表，在相应操作发生时，向其中写入对应的数据条目。

关系模式：

供应商（供应商号，供应商名，供应商联系方式）

书（书号，书名，书店销售单价）

报价（供应商号，书号，供应商报价）

多对多的关系下，三种关系模式是不能进行合并的。而两个实体联系的关系模式即是中间表的结构。

物理结构：

分析数据表可知，易变部分有 Books 表（库存数量变化），和三个日志表；而 Supplier 表、Quotes 表不太容易变化。为了提高性能，可将数据的易变部分、稳定部分、经常存取部分和存储频率较低部分分开存放。

对于 Boks 表、Supplier 表和 Quotes 表，需要随机访问其中的内容，可将其放在靠内侧的磁轨。在进行数据读取时，数据库系统会交替访问磁盘上索引的页和文件内容的页，把这两部分内容放在相邻的位置可以节省查找时间。

而对于日志表这些需要顺序扫描的数据，应将其放在靠外的磁轨。因为此时顺序查找的速度是最关键的，把数据放在靠外的磁轨可以提升顺序查找速度。

五. 系统功能的实现

1. 数据库模块具体实现

book




Table Name:

Schema: **bookmanage**

Charset/Collation: 





utf8mb4

utf8mb4\_0900\_ai\_

Engine: 

InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 Book_id	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text" value=""/>
 Book_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 Quantity	INT(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0000000000'
 Sell_Price	FLOAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

bookshop\_purchase\_log

Table Name:

Schema: **bookmanage**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Purchase_id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Purchase_time	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP
Supplier_id	CHAR(8)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Book_id	CHAR(8)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Price	FLOAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Amount	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

bookshop\_sell\_log

Table Name:

bookshop\_sell\_log

Schema:

bookmanage

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Sell_id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Sell_time	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP
Book_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Price	FLOAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Amount	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

customer\_return\_log

Table Name: <input type="text" value="customer_return_log"/>		Schema: <b>bookmanage</b>									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
Return_id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Return_time	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP	
Book_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Price	FLOAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Amount	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

quote

Table Name: <input type="text" value="quote"/>		Schema: <b>bookmanage</b>									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
Supplier_id	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Book_id	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Price	FLOAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Amount	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

supplier

Table Name: <input type="text" value="supplier"/>		Schema: <b>bookmanage</b>									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
Supplier_id	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Supplier_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Contact	CHAR(8)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	

## 2. Python 代码具体实现

若代码截图字体太小请放大查看

### (1) 数据库操作实现 (SQLTest.py)

数据库相关操作主要分为六部分：

#### ①数据库连接与游标的初始化；



```
def init():
    global db,cursor
    # 打开数据库连接
    db = pymysql.connect("localhost","root","h4p3aYing","bookManage" )
    # 使用 cursor() 方法创建一个游标对象 cursor
    cursor = db.cursor()
```

②对数据库中的某个表格进行查询；

```
def select(table,content, judge):
    sql="select "+content+" from "+table+" where "+judge+" ;"
    try:
        # 执行SQL语句
        cursor.execute(sql)
        # 获取所有记录列表
        results = cursor.fetchall()
        return results
    except:
        print("Error: unable to fetch data")
```

③对数据库中的某个表格进行插入；

```
def insert(table,content):
    sql="insert into "+table+" values"+content+" ;"
    try:
        cursor.execute(sql)
        db.commit()
    except:
        db.rollback()
        print("Error: unable to insert")
```

④对数据库中的某个表格的一些条目进行删除；

```
def delete(table,judge):
    sql="delete from "+table+" where"+judge+" ;"
    try:
        cursor.execute(sql)
        db.commit()
    except:
        db.rollback()
        print("Error: unable to delete")
```

⑤对数据库中的某个表格进行更新；

```
def update(table,content,judge):
    sql="update "+table+" set "+content+" where "+judge+" ;"
    try:
        cursor.execute(sql)
        db.commit()
    except:
        db.rollback()
        print("Error: unable to update")
```

⑥关闭数据库连接。

```
def endit():
    # 关闭数据库连接
    db.close()
```

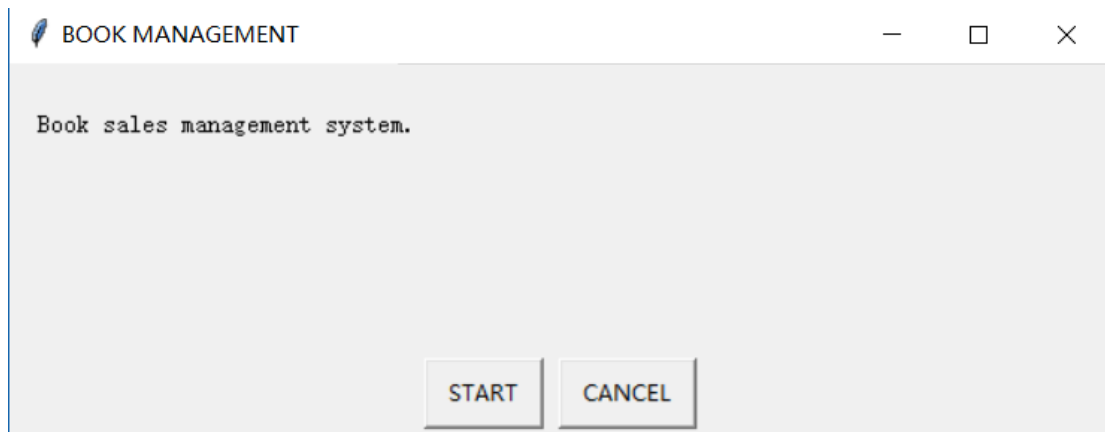
(2) GUI 界面与用户交互操作实现 (SQLGUI.py)

①主函数，系统启动

代码内容：

```
def main():
    init()
    if g.ccbox(msg='Book sales management system.', title='BOOK MANAGEMENT', choices=('START', 'CANCEL'), image=None):
        functionsOption()
    else:
        sys.exit(0)
```

界面：

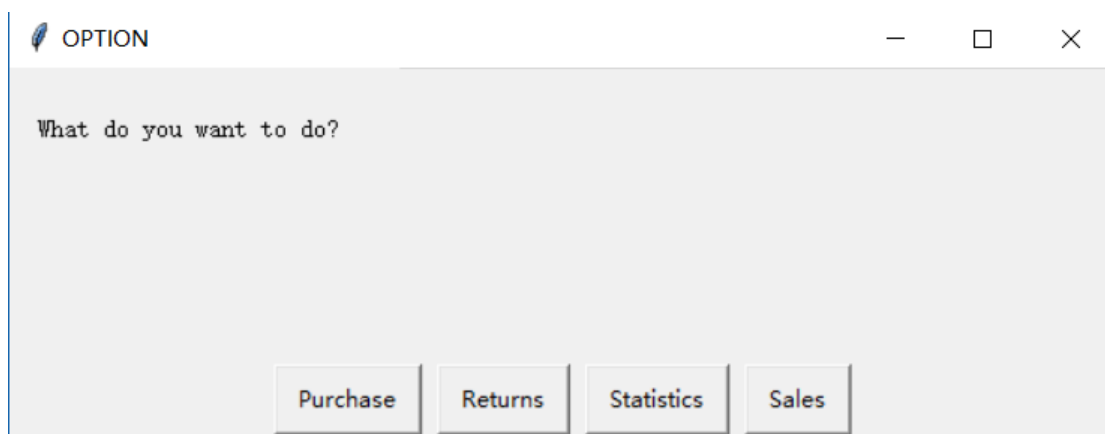


## ②功能选择界面

代码内容：

```
def functionsOption():  
    while(1):  
        option = g.buttonbox(msg='What do you want to do?', title='OPTION', choices=('Purchase', 'Returns', 'Statistics', 'Sales'), image=None)  
        if option==None:  
            sys.exit(0)  
        else:  
            if(option=='Purchase'):  
                purchase()  
            elif(option=='Returns'):  
                returns()  
            elif(option=='Statistics'):  
                statistics()  
            elif(option=='Sales'):  
                sales()  
    endit()
```

界面：

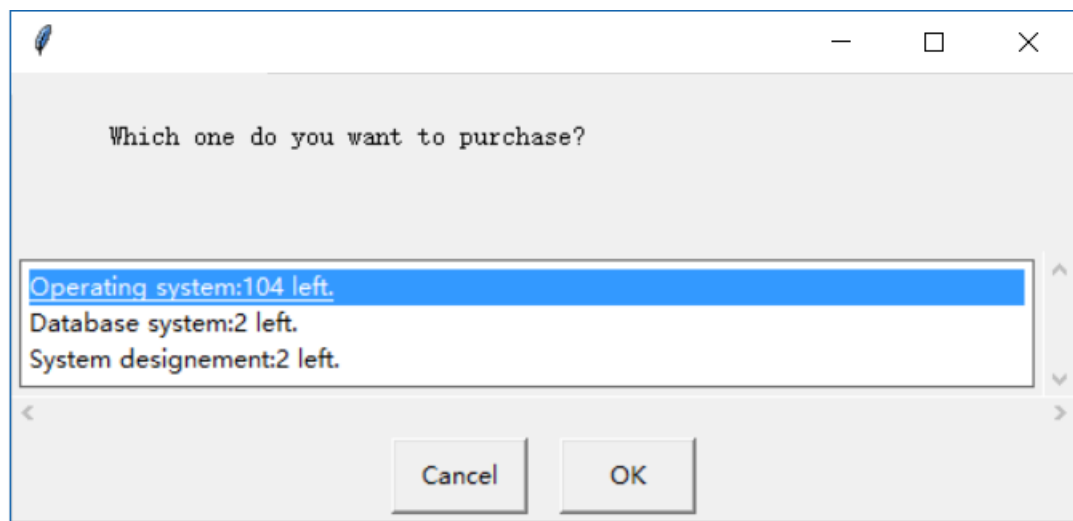


## ③进货选项界面

代码内容：

```
def purchase():
    results=select("book","*", "1=1")
    #print(results)
    temp=[]
    for row in results:
        message=row[1]+' '+str(row[2])+' left.'
        temp.append(message)
    inventory=tuple(temp)#将数据表条目的部分信息转换为元组, 便于choicebox使用
    option=g.choicebox(msg='Which one do you want to purchase?',title='',choices=inventory)
    num=temp.index(option)#获得选择的条目在数据表上的索引
    purchaseNum=100-results[num][2]#进货数量
    if(purchaseNum>0):#只有数量小于100时才进货
        quotes=select("Quote","*", "Book_id='"+results[num][0]+'")
        temp=('', '', 0x3f3f3f3f3f3f)
        for row in quotes:#找到最便宜的进货价
            if(row[2]<temp[2]):
                temp=row
        update("book","Quantity=100","Book_id='"+results[num][0]+'")
        #print(temp)
        insert("Bookshop_Purchase_Log(Supplier_id,Book_id,Price,Amount)","('"+temp[0]+'','"+temp[1]+'','"+str(temp[2])+'','"+str(purchaseNum)+'")')
        g.msgbox(msg="Successful!!!",title="")
    else:
        g.msgbox(msg="It's full",title="Warning")
```

界面:

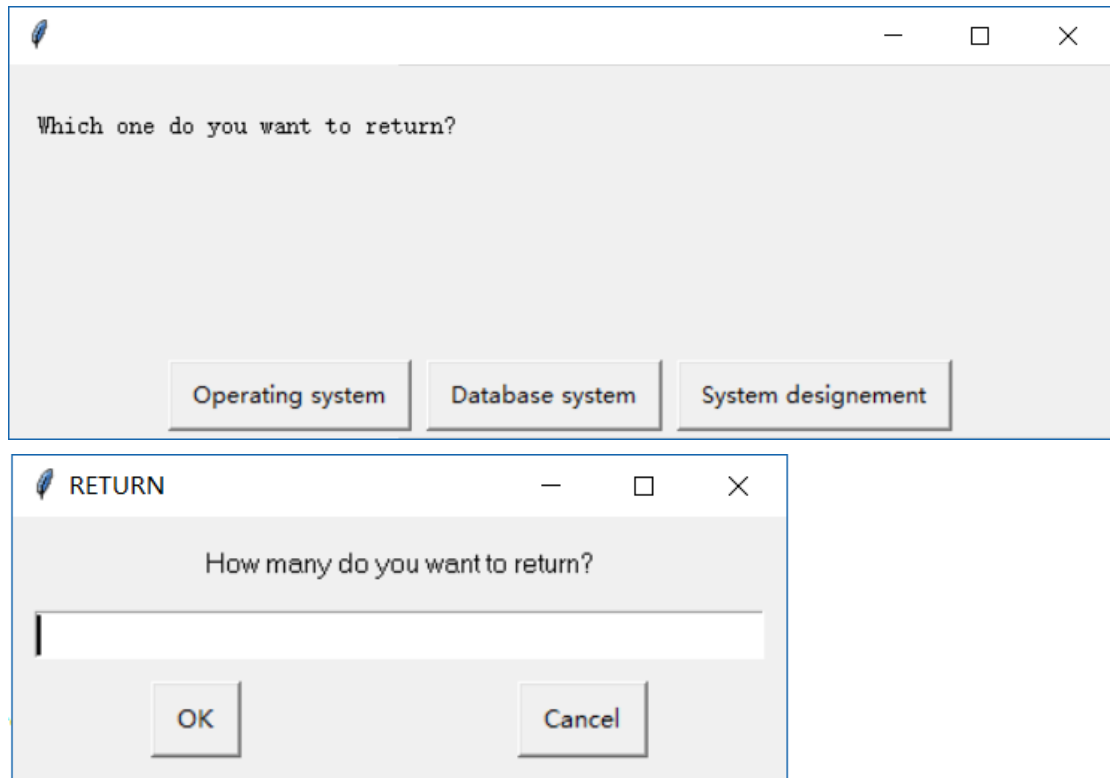


#### ④退货选项界面

代码内容:

```
def returns():
    results=select("book","*", "1=1")
    #print(results)
    temp=[]
    for row in results:
        message=row[1]
        temp.append(message)
    inventory=tuple(temp)#将数据表条目的部分信息转换为元组, 便于choicebox使用
    option=g.buttonbox(msg='Which one do you want to return?',title='',choices=inventory)
    num=temp.index(option)#获得选择的条目在数据表上的索引
    addNum=g.enterbox(msg="How many do you want to return?",title="RETURN")
    update("book","Quantity="+str(temp[num])+str(addNum),"Book_id='"+results[num][0]+'")
    insert("Customer_Return_Log(Book_name,Price,Amount)","('"+results[num][1]+'','"+str(results[num][3])+'','"+str(addNum)+'")')
```

界面:



### ⑤统计表格生成结果界面

代码内容：

```

def statistics():
    results1=select("Bookshop_Sell_Log","*", "1=1")#获得销售结果
    message={}
    for row in results1:
        if message.get(row[2])!=None:
            message[row[2]][0]+=row[4]
            message[row[2]][1]+=row[3]*row[4]
        else:
            message[row[2]]=[]
            message[row[2]].append(row[4])
            message[row[2]].append(row[3]*row[4])


    results2=select("Customer_Return_Log","*", "1=1")#获得客户退货结果
    for row in results2:
        if message.get(row[2])!=None:
            message[row[2]][0]-=row[4]
            message[row[2]][1]-=row[3]*row[4]
        else:
            message[row[2]]=[]
            message[row[2]].append((-1)*row[4])
            message[row[2]].append((-1)*row[3]*row[4])

    temp1=[]#统计出书本的买卖情况
    temp2=list(message.keys())
    for eachOne in temp2:
        temp3=message[eachOne]
        temp3.insert(0,eachOne)
        temp1.append(temp3)
    #print(temp1)

    label=['BookName','Amount','Saleroom']#写入表格
    wb = Workbook()
    ws = wb.active#第一个表
    ws.append(label)
    for i in temp1:
        ws.append(i)
    wb.save("销售记录.xlsx")

```

表格生成结果:

 销售记录.xlsx

2019/6/22 23:36 XLSX 工作表

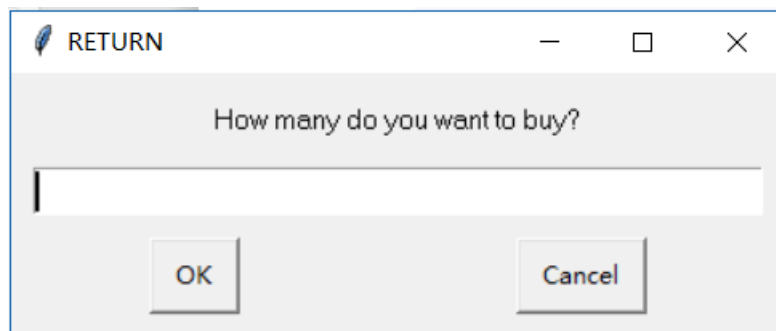
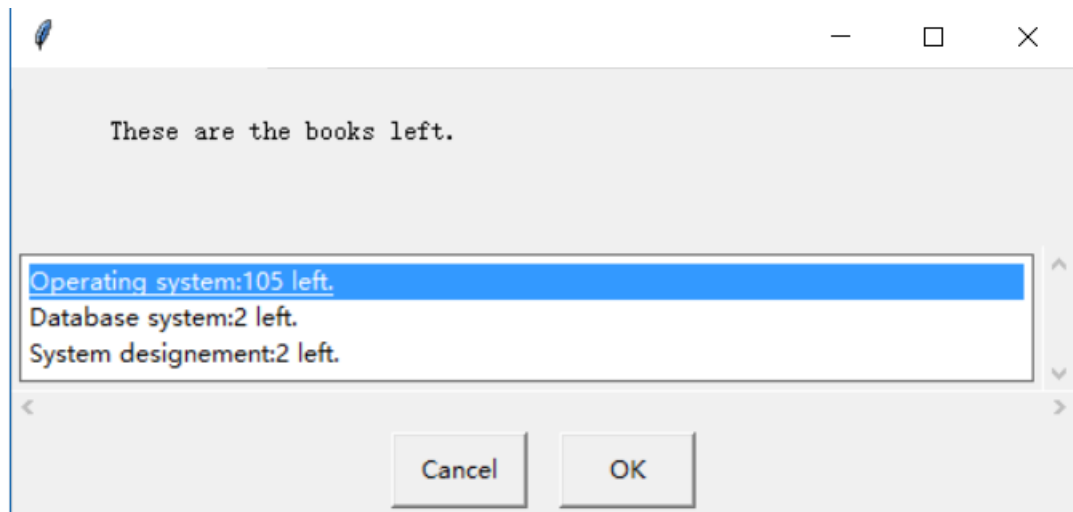
5 KB

## ⑥销售选项界面

代码内容:

```
def sales():
    results=select("book","*", "1=1")
    #print(results)
    temp=[]
    for row in results:
        message=row[1]+'-'+str(row[2])+' left.'
        temp.append(message)
    inventory=tuple(temp)#将数据表条目的部分信息转换为元组, 便于choicebox使用
    option=g.choicebox(msg='These are the books left.',title='',choices=inventory)
    num=temp.index(option)#获得选择的条目在数据表上的索引
    saleNum=g.enterbox(msg="How many do you want to buy?",title="RETURN")
    #print(saleNum)
    if(int(saleNum)<=results[num][2]):
        update("book","Quantity="+str(saleNum)+"Book_id='"+results[num][0]+'")
        insert("Bookshop_Sell_Log(Book_name,Price,Amount)","('"+results[num][1]+'','"+str(results[num][3])+"',"+str(saleNum)+")")
    else:
        g.msgbox(msg="Isn't enough.",title="Warning")
```

界面:



## 六. 总结

本课程设计中用到的《数据库系统》理论课概念与知识：

- ER 图
- 关系模式设计
- 物理结构设计
- 查询语句
- 数据库应用开发
- 文件索引
- 储存方式
- 物理结构设计
- 范式