

《数据库系统》课程设计报告

题目：图书销售管理系统（实现的功能模块：销售功能、退货功能）

小组成员信息：

姓名	学号	班级	分工
			全部内容

提交时间：2021年12月25日

一. 开发环境与开发工具

- (1) 开发环境：windows
- (2) 开发语言：C++, MySQL
- (3) 子语言：嵌入式SQL
- (4) 开发工具：
 - 数据库管理系统：MySQL, MySQL workbench 8.0 CE
 - 源程序编辑环境：visual studio 2019
- (5) 结构：C/S结构

二. 系统需求分析(2分)

(一)系统数据字典

1.数据结构

(1) 库存信息单

名字：库存信息单；
别名：图书信息；
描述：记录库存图书的详细信息
定义：库存信息单=图书id+书名+作者+库存量+销量+售价；
位置：保存到数据库book_sales_management的book表中。

(2) 供应商信息单

名字：供应商信息单；
别名：供应图书信息；
描述：记录图书供应商所供应的图书信息以及相应的报价；

定义：供应商信息=供应商id+供应图书id+报价；

位置：保存到数据库book_sales_management的supplier表中。

(3) 进货信息单

名字：进货信息单；

别名：进货图书信息；

描述：记录新华书店每次购进图书的信息；

定义：进货信息=进货单号+进货图书id+供应商id+进货数量；

位置：保存到数据库book_sales_management的goods_in表中。

(4) 退货信息单

名字：退货信息单；

别名：退书信息；

描述：记录顾客每次成功完成的退货信息；

定义：退货信息=退货单号+退货图书id+退货数量+年份+月份+日期；

位置：保存到数据库book_sales_management的returnbook表中。

(5) 销售信息单

名字：销售信息单；

别名：购书信息；

描述：记录顾客每次成功完成的购书信息；

定义：销售信息=销售单号+销售图书id+售出数量+年份+月份+日期；

位置：保存到数据库book_sales_management的purchase表中。

2.数据项

(1) “图书编号”数据项

名字：图书编号；

别名：bid；

描述：图书编号信息；

定义：int类型，由系统自增生成，每次增加1；

位置：销售信息单、退货信息单、进货信息单、库存信息单、供应商信息单。

(2) “书名”数据项

名字：书名；

别名: bname;

描述: 图书名称信息;

定义: 变长字符型名称, 字符数小于等于20;

位置: 库存信息单。

(3) “作者”数据项

名字: 作者;

别名: bauthor;

描述: 图书作者信息;

定义: 变长字符类型, 字符数小于等于20;

位置: 库存信息单。

(4) “库存量”数据项

名字: 库存量;

别名: stockn;

描述: 图书库存信息;

定义: numeric类型, 约束为正整数, 默认值为0;

位置: 库存信息单。

(5) “售价”数据项

名字: 售价;

别名: price;

描述: 图书售价信息;

定义: numeric类型, 精确度小数点后两位, 默认值为0.00;

位置: 库存信息单。

(6) “进货单编号”数据项

名字: 进货单编号;

别名: gno;

描述: 进货单编号信息;

定义: int类型, 由系统自增生成, 每次增加1;

位置: 进货信息单。

(7) “进货量”数据项

名字: 进货量;
别名: ginum;
描述: 进货数量信息;
定义: numeric类型, 约束为正整数, 默认值为0;
位置: 进货信息单。

(8) “供货商id”数据项

名字: 供货商id;
别名: sid;
描述: 供货商id信息;
定义: int类型;
位置: 进货信息单、供应商信息单。

(9) “供货商报价”数据项

名字: 供货商报价;
别名: offer;
描述: 供货商报价信息;
定义: numeric类型, 精确度小数点后两位, 默认值为0.00;
位置: 供应商信息单。

(10) “退货单编号”数据项

名字: 退货单编号;
别名: rno;
描述: 退货单编号信息;
定义: int类型, 由系统自增生成, 每次增加1;
位置: 退货信息单。

(11) “退货数量”数据项

名字: 退货数量;
别名: renum;
描述: 退货数量信息;
定义: numeric类型, 约束为正整数, 默认值为0;
位置: 退货信息单。

(12) “年份”数据项

名字: 年份;

别名: ryear, pyear;

描述: 订单生成的年份信息;

定义: int类型;

位置: 退货信息单、销售信息单。

(12) “月份”数据项

名字: 月份;

别名: rmonth, pmonth;

描述: 订单生成的月份信息;

定义: int类型, 处于区间 [1,12] 中;

位置: 退货信息单、销售信息单。

(12) “日期”数据项

名字: 日期;

别名: rdate, pdate;

描述: 订单生成的日期信息;

定义: int类型, 处于区间 [1,31] 中;

位置: 退货信息单、销售信息单。

(13) “销售单编号”数据项

名字: 销售单编号;

别名: pno;

描述: 销售单编号信息;

定义: int类型, 由系统自增生成, 每次增加1;

位置: 销售信息单。

(14) “销售数量”数据项

名字: 销售数量;

别名: purnum;

描述: 销售数量信息;

定义: numeric类型, 约束为正整数, 默认值为0;

位置: 销售信息单。

3.数据流

本实验实现的是销售与退货功能，涉及的数据流如下，

(1) 名称：销售信息单数据流

来源：顾客提交购书需求

去向：销售功能模块处理事务

(2) 名称：退货信息单数据流

来源：顾客提交退货需求

去向：退货功能模块处理事务

4.数据存储

数据存储存储在book_sales_management库中，具体的存储示例数据在“四.系统设计”中给出。

5.处理过程

本实验实现的是销售与退货功能，具体的处理过程在“五.系统功能的实现”中给出。

(二)功能分析

1. 书店的库存信息管理：

书店需要对库存图书信息进行检查、增加、删除、修改。涉及的信息包括图书的id、书名、库存量、销量、售价、成本信息。

顾客需要对库存图书信息进行检查，可查看的信息有图书的id、书名、库存量、销量、售价信息。

2. 供应商的供货信息管理：

供应商需要对供应商信息进行检查、增加、删除、修改。涉及的信息包括供应商的id信息、供应图书的id信息和供应图书的报价。

书店需要对供应商信息进行检查。涉及的信息包括供应商的id信息、供应图书的id信息和供应图书的报价。

3. 书店的进货信息管理：

书店需要对进货信息进行检查、增加、删除、修改。包括进货单号、进货图书的id、供应商的id信息、进货数量、进货日期。

供应商需要对进货信息进行检查。涉及的信息包括进货单号、进货图书的id、供应商的id信息、进货数量、进货日期。

4. 顾客的退货信息管理：

书店需要对退货信息进行了查询。包括退货单号、退货图书的id、退货数量、退货日期。

顾客需要对退货信息进行了查询、增加、删除、修改。包括退货单号、退货图书的id、退货数量、退货日期。

5. 顾客的销售信息管理：

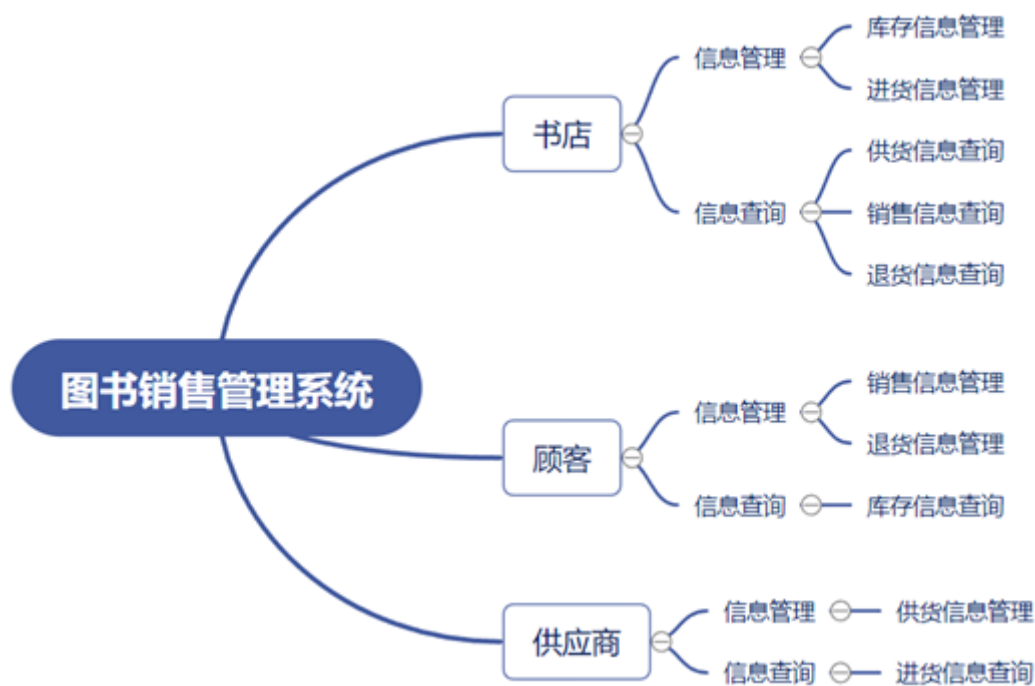
书店需要对销售信息进行了查询。包括销售单号、销售图书的id、售出数量、销售日期。

顾客需要对销售信息进行了查询、增加、删除、修改。包括销售单号、销售图书的id、售出数量、销售日期。

三．功能需求分析（3分）

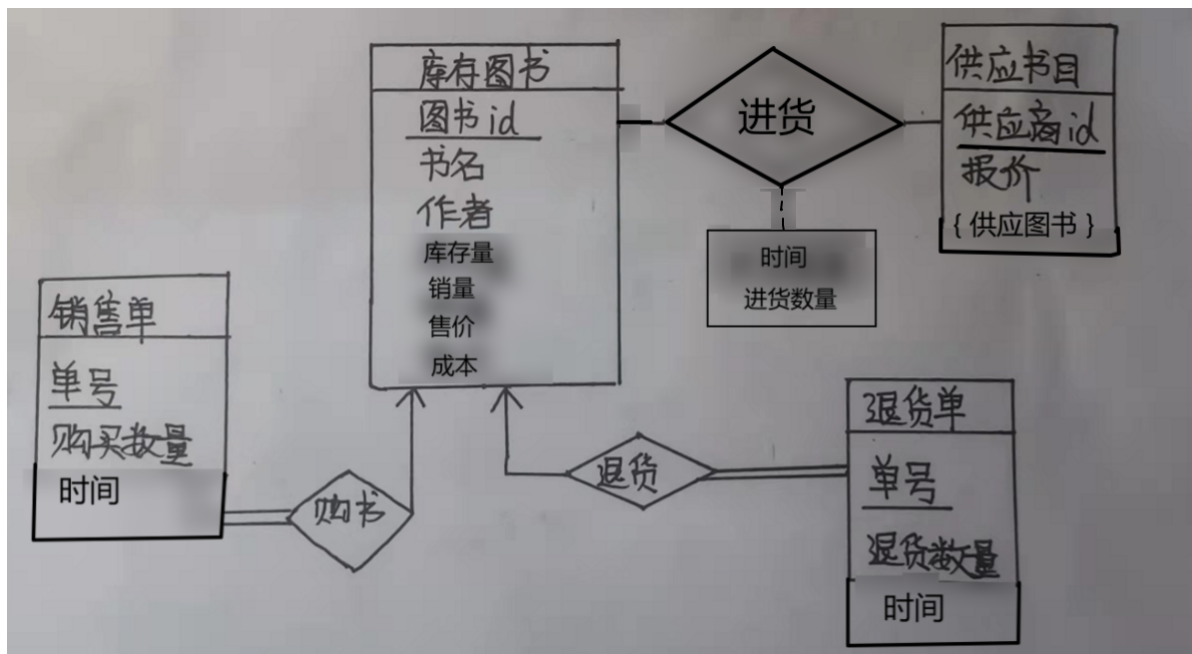
根据题目要求以及系统需求分析，主要需要实现对库存图书的信息、供货信息、进货信息、退货信息、销售信息的管理。同时根据题目要求需要有统计的功能，即根据销售情况输出统计的报表，这部分可通过视图实现。本人实现的功能是销售功能与退货功能。

系统功能模块图如下，其中信息管理包括查询、增加、删除、修改的功能：



四．系统设计（10分）

1.数据概念结构设计（系统ER图）（5分）



2.数据库关系模式设计（3分）

根据ER图生成的关系模式设计如下（带下划线为主码）：

- 库存信息（图书id，书名，作者，库存量，销量，售价，成本）
 - 在物理结构设计中，对应的命名如下: book (bid, bname, bauthor, stockn, selln, price, cost)
- 供应商信息（供应商id，供应图书id，报价）
 - 在物理结构设计中，对应的命名如下: supplier (bid, sid, offer)
- 进货信息（进货单号，进货图书id，供应商id，进货数量，日期）
 - 在物理结构设计中，对应的命名如下: goods_in (gno, bid, sid, ginum, year, month, date)
- 退货信息（退货单号，退货图书id，退货数量，年份，月份，日期）
 - 在物理结构设计中，对应的命名如下: returnbook (rno, bid, renum, ryear, rmonth, rdate)
- 销售信息（销售单号，销售图书id，售出数量，年份，月份，日期）
 - 在物理结构设计中，对应的命名如下: purchase (pno, bid, purnum, pyear, pmonth, pdate)

3.数据库物理结构设计（2分）

(1)创建库：

```
1 | create database Book_Sales_Management;
```

(2)创建库存信息表：


```

1  use Book_Sales_Management;
2  create table if not exists book
3      (bid          int auto_increment,
4       bname        varchar(20),
5       bauthor      varchar(20),
6       stockn       numeric(10,0)  check (stockn > 0)    default 0 not null,
7       selln        numeric(10,0)  check (selln > 0)    default 0 not null,
8       price        numeric(8,2)   check (price > 0)    default 0 not null,
9       cost         numeric(8,2)   check (price > 0)    default 0 not null
10     primary key (bid)
11     )auto_increment=1 engine=InnoDB;
12

```

(3)创建供应商信息表:

```

1  use Book_Sales_Management;
2  create table if not exists supplier
3      (bid          int,
4       sid          int,
5       offer        numeric(8,2)   check (offer > 0)    default 0 not null,
6       primary key (bid,sid)
7     )engine=InnoDB;
8

```

(4)创建进货信息表:

```

1  use Book_Sales_Management;
2  create table if not exists goods_in
3      (gno          int auto_increment primary key,
4       bid          int,
5       sid          int,
6       ginum        numeric(10,0)  check (ginum > 0)    default 1 not null,
7       year int,
8       month int check(pmonth>0 and pmonth<13),
9       date int check(pdate>0 and pdate<32)
10     CONSTRAINT fk_goods_in1 FOREIGN KEY(bid,sid) REFERENCES
11     supplier(bid,sid)
12     )auto_increment=1 engine=InnoDB;

```

(5)创建退货信息表:

```

1  use Book_Sales_Management;
2  create table if not exists returnbook
3      (rno          int auto_increment primary key,
4       bid          int,
5       renum        numeric(10,0)  check (renum > 0)    default 1 not null,
6       ryear int,
7       rmonth int check(pmonth>0 and pmonth<13),
8       rdate int check(pdate>0 and pdate<32)
9     )auto_increment=1 engine=InnoDB;
10

```

(6)创建销售信息表:

```

1 use Book_Sales_Management;
2 create table if not exists purchase
3     (pno      int auto_increment primary key,
4      bid      int,
5      purnum   numeric(10,0) check (purnum > 0) default 1 not null,
6      pyear   int,
7      pmonth  int check(pmonth>0 and pmonth<13),
8      pdate   int check(pdate>0 and pdate<32)
9      )auto_increment=1 engine=InnoDB;
10

```

(7)插入示例数据（部分）：

```

1 INSERT INTO `book_sales_management`.`book` (`bid`, `bname`, `bauthor`,
2 `stockn`, `selln`, `price`) VALUES ('1', '简爱', '夏洛蒂·勃朗特', '1000',
3 '100', '30.10');
4 INSERT INTO `book_sales_management`.`book` (`bname`, `bauthor`, `stockn`,
5 `selln`, `price`) VALUES ('飘', '玛格丽特·米切尔', '542', '227', '33.40');
6 INSERT INTO `book_sales_management`.`book` (`bname`, `bauthor`, `stockn`,
7 `selln`, `price`) VALUES ('将心注入', '霍华德', '754', '210', '34.56');
8 INSERT INTO `book_sales_management`.`book` (`bname`, `bauthor`, `stockn`,
9 `selln`, `price`) VALUES ('西西弗神话', '加缪', '779', '124', '29.45');
10 INSERT INTO `book_sales_management`.`book` (`bname`, `bauthor`, `stockn`,
11 `selln`, `price`) VALUES ('局外人', '加缪', '478', '241', '34.06');
12 INSERT INTO `book_sales_management`.`book` (`bname`, `bauthor`, `stockn`,
13 `selln`, `price`) VALUES ('鼠疫', '加缪', '879', '334', '27.45');
14 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
15 VALUES ('1', '1', '20.00');
16 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
17 VALUES ('2', '1', '15.20');
18 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
19 VALUES ('3', '1', '24.10');
20 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
21 VALUES ('4', '1', '17.25');
22 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
23 VALUES ('1', '2', '19.22');
24 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
25 VALUES ('5', '2', '24.31');
26 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
27 VALUES ('6', '2', '14.55');
28 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
29 VALUES ('1', '3', '12.33');
30 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
31 VALUES ('2', '3', '14.87');
32 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
33 VALUES ('3', '3', '17.88');
34 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
35 VALUES ('5', '3', '19.45');
36 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
37 VALUES ('4', '4', '14.69');
38 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
39 VALUES ('5', '4', '19.25');
40 INSERT INTO `book_sales_management`.`supplier` (`bid`, `sid`, `offer`)
41 VALUES ('6', '4', '14.79');
42 INSERT INTO `book_sales_management`.`goods_in` (`bid`, `sid`, `ginum`)
43 VALUES ('1', '3', '50');

```

```

22 INSERT INTO `book_sales_management`.`purchase` (`bid`, `purnum`) VALUES
   ('1', '2');
23 INSERT INTO `book_sales_management`.`purchase` (`bid`, `purnum`) VALUES
   ('2', '5');
24 INSERT INTO `book_sales_management`.`purchase` (`bid`) VALUES ('1');
25 INSERT INTO `book_sales_management`.`purchase` (`bid`) VALUES ('6');
26

```

(8) 由于库存中的书数目众多，为了加快检索的速率，在book表中基于bid（图书编号）建立索引。

Create Index for Table book_sales_management.book

Index Name: Unique

Columns:

Column	Length
bid	

Comments:

Create/Online Options

Algorithm: Default

Locking: Default (allow as much concurrency as possible)

Create Cancel

五. 系统功能的实现 (5分)

1.初始数据库准备

根据 四.系统设计 中的 3.数据库物理结构设计 的代码，创建好数据库，并插入部分数据。结果如下：

- book表

	bid	bname	bauthor	stockn	selln	price	cost
	1	Jane Eyre	Charlotte Brontë	770	100	30.10	25.00
	2	Gone with the wind	Margaret Mitchell	474	227	33.40	18.00
	3	将心注入	霍华德	754	210	34.56	19.00
	4	西西弗神话	加缪	774	124	29.45	14.00
	5	局外人	加缪	376	241	34.06	11.00
	6	鼠疫	加缪	879	334	27.45	10.00
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- goods_in表

	gno	bid	sid	ginum	year	month	date
	1	1	3	50	2021	1	5
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- returnbook表

	rno	bid	renum	ryear	rmonth	rdate
▶	200001	1	1	2021	12	23
	200002	1	1	2021	12	14
	200003	2	5	2021	12	1
	200004	2	12	2021	1	4
	200005	2	1	2021	1	4
	200006	1	1	2021	1	1
	200007	1	1	2021	1	1
	200008	2	2	2021	4	5
	200009	1	100	2020	1	1
	200010	1	1	2021	1	11
*	NULL	NULL	NULL	NULL	NULL	NULL

- purchase表

	pno	bid	purnum	pyear	pmonth	pdate
▶	100001	1	1	2020	5	6
	100002	5	80	2020	11	4
	100003	1	1	2020	11	4
	100004	2	11	2020	11	8
	100005	1	1	2021	5	10
	100006	2	10	2020	1	1
	100007	1	1	2020	12	25
	100008	2	12	2020	12	26

- supplier表

	bid	sid	offer
▶	1	1	20.00
	1	2	19.22
	1	3	12.33
	2	1	15.20
	2	3	14.87
	3	1	24.10
	3	3	17.88
	4	1	17.25
	4	4	14.69
	5	2	24.31
	5	3	19.45
	5	4	19.25
	6	2	14.55
	6	4	14.79

2. 功能实现过程

选择实现的功能模块是销售功能和退货功能，另外还需实现用户使用的功能界面。分别简述如下。

(1) 功能界面

本部分较为简单，在main函数中实现，实现步骤如下所示。其中2与3包含着一个循环语句中，从而保证能够满足用户多个功能需求。

1. 初始化MySQL并连接book_sales_management数据库
2. 打印功能界面信息：

```
//打印初始化功能界面
printf("\nWelcome to the Book Sales Management System of Xinhua Booksore!\n");
printf("Please select the function according to your needs:\n");
printf("    0--exit.\n");//退出
printf("    1--purchase books.\n");//购书
printf("    2--return boooks.\n");//退书
printf("    3--sales statistics.\n");//统计
printf("    4--goods in.\n\n");//进货
```

3. 读取用户需求，并根据需求执行相应函数或指令。

```
fu[0] = '0';
scanf("%s", &fu);//读取用户功能需求
//执行相应功能需求，本实验实现的功能是购书与退书功能
if (fu[0] == '0')exit(0);
if (fu[0] == '1')purchase_book();
if (fu[0] == '2')return_book();
if (fu[0] == '3')printf("\nSorry, this function is unavailable.\n");
if (fu[0] == '4')printf("\nSorry, this function is unavailable.\n");
```

4. 访问完毕，关闭mysql.

完整的实现代码如下：

```
1  int main(int argc, char** argv, char** envp)
2  {
3      int num = 0;
4      char fu[2];
5      mysql_init(&mysql);//获得或初始化一个MYSQL结构
6
7      //显示功能列表，并根据需求执行相应功能
8      if (mysql_real_connect(&mysql, "localhost", "root", "123456",
9      "book_sales_management", 3306, 0, 0)) {
10         for (;;) {
11             //打印初始化功能界面
12             printf("\nwelcome to the Book Sales Management System of Xinhua
13             Booksore!\n");
14             printf("Please select the function according to your needs:\n");
15             printf("    0--exit.\n");//退出
16             printf("    1--purchase books.\n");//购书
17             printf("    2--return boooks.\n");//退书
18             printf("    3--sales statistics.\n");//统计
19             printf("    4--goods in.\n\n");//进货
20             fu[0] = '0';
21             scanf("%s", &fu);//读取用户功能需求
22             //执行相应功能需求，本实验实现的功能是购书与退书功能
23             if (fu[0] == '0')exit(0);
24             if (fu[0] == '1')purchase_book();
25             if (fu[0] == '2')return_book();
26             if (fu[0] == '3')printf("\nSorry, this function is
27             unavailable.\n");
28             if (fu[0] == '4')printf("\nSorry, this function is
29             unavailable.\n");
30             //system("pause");
31         }
32     }
33     else {
34         printf("Sorry, the database doesn't exist.");
35     }
```

```

31     }
32     mysql_close(&mysql); //访问完毕，关闭mysql
33     result = mysql_store_result(&mysql);
34     mysql_free_result(result);
35     system("pause");
36     return 0;
37 }
38

```

(2) 退货功能

功能需求如下：

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库中。

本部分在 `int return_book()` 中实现。主体内容在while循环中，分别执行如下操作：

1.读取用户退书信息

```

while (1) {
    //读取退书信息，形成插入退货单的指令
    printf("\nPlease input book id:"); scanf("%s", &bid); strcat(strquery, bid);
    strcat(strquery, ",");
    printf("Please input renum:"); scanf("%s", &renum); strcat(strquery, renum);
    strcat(strquery, ",");
    printf("Please input year:"); scanf("%s", &ryear); strcat(strquery, ryear);
    strcat(strquery, ",");
    printf("Please input month:"); scanf("%s", &rmonth); strcat(strquery, rmonth);
    strcat(strquery, ",");
    printf("Please input date:"); scanf("%s", &rdate); strcat(strquery, rdate);
    strcat(strquery, "');");
    //形成更新库存量的指令
}

```

2.根据退书信息形成插入退货单和更新库存数量的指令

3.执行上面的指令，

(1) 执行成功，则自动打印退货单

```

if (mysql_query(&mysql, strquery) == 0 && mysql_query(&mysql, strquery2) == 0) {
    printf("Execute successfully!\n");
    //执行成功，输出退货单
    printf("\n退货单:  \n");
    //查询退货单号
    mysql_query(&mysql, "select max(rno) from returnbook;");
    result = mysql_store_result(&mysql);
    cout << " " << "退货单号: " << mysql_fetch_row(result)[0] << endl;
    cout << " " << "图书 id : " << bid << endl;
    cout << " " << "退货数目: " << renum << endl;
    cout << " " << "退货时间: " << ryear << "年" << rmonth << "月" << rdate << "日" << endl;
    cout << endl;
}

```

(2) 如果执行失败则返回出错的信息。

4.询问用户是否继续退出，如果是则重复上述操作，否则退出函数。

```

//判断是否继续退书
printf("Return again?(y--yes,n--no):");
scanf("%s", &yn);
if (yn[0] == 'y' || yn[0] == 'Y') { continue; }
else break;

```

完整的实现代码如下：

```

1  int return_book() {
2      char bid[] = "001"; char renum[] = "1"; char ryear[] = "2021"; char
rmonth[] = "12"; char rdate[] = "23";
3      char strquery[100] = "insert into
returnbook(bid,renum,ryear,rmonth,rdate) values('"; //存储插入退货单的指令
4      char yn[2];
5      while (1) {
6          //读取退书信息，形成插入退货单的指令
7          printf("\nPlease input book id:"); scanf("%s", &bid);
strcat(strquery, bid);
8          strcat(strquery, "','");
9          printf("Please input renum:"); scanf("%s", &renum); strcat(strquery,
renum);
10         strcat(strquery, "','");
11         printf("Please input year:"); scanf("%s", &ryear); strcat(strquery,
ryear);
12         strcat(strquery, "','");
13         printf("Please input month:"); scanf("%s", &rmonth);
strcat(strquery, rmonth);
14         strcat(strquery, "','");
15         printf("Please input date:"); scanf("%s", &rdate); strcat(strquery,
rdate);
16         strcat(strquery, "');");
17         //形成更新库存量的指令
18         char strquery2[100] = "update book set stockn = stockn - "; //存储更
新库存量的指令
19         strcat(strquery2, renum);
20         strcat(strquery2, " where bid = '");
21         strcat(strquery2, bid);
22         strcat(strquery2, "');");
23         //进行数据修改：增加库存量，并将退货信息加入退货表
24         if (mysql_query(&mysql, strquery) == 0 && mysql_query(&mysql,
strquery2) == 0) {
25             printf("Execute successfully!\n");
26             //执行成功，输出退货单
27             printf("\n退货单:  \n");
28             //查询退货单号
29             mysql_query(&mysql, "select max(rno) from returnbook;");
30             result = mysql_store_result(&mysql);
31             cout << " " << "退货单号: " << mysql_fetch_row(result)[0] <<
endl;
32             cout << " " << "图书 id : " << bid << endl;
33             cout << " " << "退货数目: " << renum << endl;
34             cout << " " << "退货时间: " << ryear << "年" << rmonth << "月" <<
rdate << "日" << endl;
35             cout << endl;
36         }
37         else { printf("Execute failed!\n\n"); }

```



```

38         //判断是否继续退书
39         printf("Return again?(y--yes,n--no):");
40         scanf("%s", &yn);
41         if (yn[0] == 'y' || yn[0] == 'Y') { continue; }
42         else break;
43     }
44     return 0;
45 }

```

(3) 销售功能

功能需求如下：

④ 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。

本部分在int purchase_book()中实现。主体内容在while循环中，分别执行如下操作：

1. 读取用户购书信息，其中需要根据用户输出的图书信息自动形成查询图书库存的指令，然后再界面上打印出图书的库存信息。

```

//读取购买信息，并形成查询图书库存的指令
printf("\nPlease input book id:"); scanf("%s", &bid); strcat(strquery, bid);
strcat(strquery, " ");
//查询图书的库存量并自动显示
mysql_query(&mysql, strquery);
result = mysql_store_result(&mysql);
cout << "图书库存量: " << mysql_fetch_row(result)[0] << endl;
//继续读取购买信息
printf("Please input purchase quantity:"); scanf("%s", &purchasenum);
printf("Please input year:"); scanf("%s", &year);
printf("Please input month:"); scanf("%s", &month);
printf("Please input date:"); scanf("%s", &date);

```

2. 根据购书信息形成更新库存数量的指令

```

//购买图书，即形成减少库存量的指令
char strquery2[100] = "update book set stockn=stockn-"; //存储更新库存量的指令
strcat(strquery2, purchasenum);
strcat(strquery2, " where bid = '");
strcat(strquery2, bid);
strcat(strquery2, " ");

```

3. 执行上面的指令。在数据库的book表中添加了图书图书库存量必须大于等于0的约束，如果图书购买量大于库存量会不满足约束，进而导致指令执行失败。因此如果执行成功，必然是图书购买量小于等于库存量

(1) 执行成功，则先形成将购买信息添加到日销售表中的指令并执行，之后自动打印出销售单。

- 形成将购买信息添加到日销售表中的指令并执行：


```
//将购买信息添加到日销售表中
char strquery3[100] = "insert into purchase(bid,purnum,pyear,pmonth,pdate) values('";
char strquery4[100] = "select bname from book where bid = '"; //存储查询书名的指令
strcat(strquery3, bid);
strcat(strquery3, "','");
strcat(strquery3, purchasenum);
strcat(strquery3, "','");
strcat(strquery3, year);
strcat(strquery3, "','");
strcat(strquery3, month);
strcat(strquery3, "','");
strcat(strquery3, date);
strcat(strquery3, "');"); //形成指令
if (mysql_query(&mysql, strquery3) == 0) { //执行指令
    printf("成功添加到日销售表中!\n");
}
else {
    printf("添加到日销售表失败!\n");
}
```

- 打印出销售单:

```
//修改库存执行成功, 输出销售单
printf("\n销售单: \n");
mysql_query(&mysql, "select max(pno) from purchase;"); //查询单号
result = mysql_store_result(&mysql);
cout << " " << "销售单号: " << mysql_fetch_row(result)[0] << endl;
cout << " " << "图书 id : " << bid << endl;
strcat(strquery4, bid); strcat(strquery4, "'; "); mysql_query(&mysql, strquery4); //查询书名
result = mysql_store_result(&mysql);
row = mysql_fetch_row(result);
cout << " " << "书名 : " << row[0] << endl;
cout << " " << "购买数目: " << purchasenum << endl;
cout << " " << "购买时间: " << year << "年" << month << "月" << date << "日" << endl;
cout << endl;
```

(2) 如果执行失败则返回出错的信息。

```
//如果购买数量大于库存, 无法成功购买, 减少库存指令执行失败
else {
    printf("库存不足, 购买失败! \n\n");
}
```

4. 询问用户是否继续购书, 如果是则重复上述操作, 否则退出函数。

```
//判断是否继续购书
printf("Purchase again?(y--yes, n--no):");
scanf("%s", &yn);
if (yn[0] == 'y' || yn[0] == 'Y') { continue; }
else break;
```

```
1 int purchase_book() {
2     char bid[3]; char purchasenum[100]; char year[] = "2021"; char month[] =
    "12"; char date[] = "23";
3     char yn[2];
4     while (1) {
5         char strquery[100] = "select stockn from book where bid= '"; //存储查
            询图书库存的指令
6         //读取购买信息, 并形成查询图书库存的指令
7         printf("\nPlease input book id:"); scanf("%s", &bid);
            strcat(strquery, bid);
8         strcat(strquery, "'; ");
```

```

9      //查询图书的库存量并自动显示
10     mysql_query(&mysql, strquery);
11     result = mysql_store_result(&mysql);
12     cout << "图书库存量: " << mysql_fetch_row(result)[0] << endl;
13     //继续读取购买信息
14     printf("Please input purchase quantity:"); scanf("%s",
    &purchasenum);
15     printf("Please input year:"); scanf("%s", &year);
16     printf("Please input month:"); scanf("%s", &month);
17     printf("Please input date:"); scanf("%s", &date);
18
19     //购买图书，即形成减少库存量的指令
20     char strquery2[100] = "update book set stockn=stockn-"; //存储更新库存
    量的指令
21     strcat(strquery2, purchasenum);
22     strcat(strquery2, " where bid = '");
23     strcat(strquery2, bid);
24     strcat(strquery2, "';");
25     //进行购买操作
26     //如果购买数量小于库存，则可以购买，执行后续操作则打印销售单，修改库存，并将购买
    信息添加到日销售表中
27     if (mysql_query(&mysql, strquery2) == 0) { //修改库存
28         printf("Execute successfully!\n");
29
30         //将购买信息添加到日销售表中
31         char strquery3[100] = "insert into
    purchase(bid,purnum,pyear,pmonth,pdate) values('"; //存储将购买信息添加到日销售表
    中的指令
32         char strquery4[100] = "select bname from book where bid ='"; //存
    储查询书名的指令
33         strcat(strquery3, bid);
34         strcat(strquery3, "','");
35         strcat(strquery3, purchasenum);
36         strcat(strquery3, "','");
37         strcat(strquery3, year);
38         strcat(strquery3, "','");
39         strcat(strquery3, month);
40         strcat(strquery3, "','");
41         strcat(strquery3, date);
42         strcat(strquery3, "');"); //形成指令
43         if (mysql_query(&mysql, strquery3) == 0) { //执行指令
44             printf("成功添加到日销售表中!\n");
45         }
46         else {
47             printf("添加到日销售表失败!\n");
48         }
49
50         //修改库存执行成功，输出销售单
51         printf("\n销售单: \n");
52         mysql_query(&mysql, "select max(pno) from purchase;"); //查询单号
53         result = mysql_store_result(&mysql);
54         cout << " " << "销售单号: " << mysql_fetch_row(result)[0] <<
    endl;
55         cout << " " << "图书 id : " << bid << endl;
56         strcat(strquery4, bid); strcat(strquery4, "';
    ");mysql_query(&mysql, strquery4); //查询书名
57         result = mysql_store_result(&mysql);
58         row = mysql_fetch_row(result);

```

```

59         cout << " " << "书名      : " << row[0] << endl;
60         cout << " " << "购买数目: " << purchasenum << endl;
61         cout << " " << "购买时间: " << year << "年" << month << "月" <<
date << "日" << endl;
62         cout << endl;
63     }
64     //如果购买数量大于内存, 无法成功购买, 减少库存指令执行失败
65     else {
66         printf("库存不足, 购买失败! \n\n");
67     }
68     //判断是否继续购书
69     printf("Purchase again?(y--yes,n--no:");
70     scanf("%s", &yn);
71     if (yn[0] == 'y' || yn[0] == 'Y') { continue; }
72     else break;
73 }
74 return 0;
75 }

```

3.客户端运行界面（命令行形式）：

```

Welcome to the Book Sales Management System of Xinhua Booksore!
Please select the function according to your needs:
0--exit.
1--purchase books.
2--return boooks.
3--sales statistics.
4--goods in.

```

开始页面

3 输入统计指令

Sorry, this function is unavailable. 未实现该功能

```

Welcome to the Book Sales Management System of Xinhua Booksore!
Please select the function according to your needs:
0--exit.
1--purchase books.
2--return boooks.
3--sales statistics.
4--goods in.

```

4 输入进货指令

Sorry, this function is unavailable. 未实现该功能

Welcome to the Book Sales Management System of Xinhua Booksore!

Please select the function according to your needs:

0--exit.
1--purchase books.
2--return boooks.
3--sales statistics.
4--goods in.

2 输入退书指令

Please input book id:1
Please input renum:1
Please input year:2021
Please input month:1
Please input date:11

输入退书信息

Execute successfully!

成功修改库存并把信息加入退货库中

退货单:
退货单号: 200010
图书 id : 1
退货数目: 1
退货时间: 2021年1月11日

打印退货单

Return again?(y--yes,n--no):n

询问是否继续退书

Welcome to the Book Sales Management System of Xinhua Booksore!

Please select the function according to your needs:

0--exit.
1--purchase books.
2--return boooks.
3--sales statistics.
4--goods in.

1 输入购书指令

Please input book id:1

图书库存量: 772

自动显示库存量

输入购书信息

Please input purchase quantity:2

Please input year:2021

Please input month:5

Please input date:6

Execute successfully!

成功添加到日销售表中!

可以销售, 打印销售单并修改库存

将销售信息加入日销售表中

销售单:
销售单号: 100017
图书 id : 1
书名 : Jane Eyre
购买数目: 2
购买时间: 2021年5月6日

销售单

Purchase again?(y--yes,n--no):y 询问是否继续购书

Please input book id:2

图书库存量: 474

Please input purchase quantity:2000

不可以销售, 超过库存量

Please input year:2021

Please input month:1

Please input date:4

库存不足, 购买失败!

说明失败原因并结束购书进程

Purchase again?(y--yes,n--no):n 询问是否继续购书

```
Welcome to the Book Sales Management System of Xinhua Booksore!  
Please select the function according to your needs:  
0--exit.  
1--purchase books.  
2--return boooks.  
3--sales statistics.  
4--goods in.
```

0 输入退出指令

退出图书销售管理系统：

```
0  
D:\VisualStudioCommunity2019\projects\mysql\x64\Debug\mysql.exe (进程 4384) 已退出，代码为 0。  
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”  
按任意键关闭此窗口。 . . .
```

六. 总结

本课程设计在符合课本题目要求的基础上，结合现实需求进行设计。同时尽可能地结合课本的理论概念与知识，达到学以致用用的作用。设计中使用了大量的 MySQL 语法知识，同时运用了根据客户需求设计数据库的步骤与方法，在概念设计阶段、逻辑设计阶段和物理设计阶段均运用了理论和实验课上的知识。另外还使用了嵌入式SQL进行编程。具体梳理如下，

使用到理论课概念与知识如下（以第六版课本为准）：

- 第2章：关系模型、数据库模式、码、模式图、关系查询语言、关系运算
- 第3章：数据定义、基本运算、基本结构、集合运算、空值、数据库修改
- 第4章：视图、完整性约束、数据类型与模式、默认值、创建索引
- 第5章：嵌入式 MySQL、函数与过程、触发器
- 第7章：数据库设计、实体-联系模型、约束、删除冗余属性、实体-联系图、转化为关系模式、多值属性、复合属性
- 第8章：范式、函数依赖、分解算法
- 第11章：索引