

《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
王若琪	18340166	计科 7 班	总体结构设计, 进货、出售等相关功能函数的实现, 部分交互界面设计, 部分实验报告编写
吴晓淳	18340174	计科 7 班	总体结构设计, 退货、统计等相关功能函数的实现, 部分交互界面设计, 部分实验报告编写

提交时间: 2021 年 1 月 5 日

一. 开发环境与开发工具

应用系统开发环境是采用 MySQL 及其支持的 C, 具体包括:

- (1) 开发语言: C。
- (2) 编译与连接工具: VC++.NET。
- (3) 子语言: MySQL 嵌入式 SQL。
- (4) 数据库管理系统: MySQL 5.7。
- (5) 操作系统: Windows 10。

二. 系统需求分析(2 分)

设计的图书销售管理系统主要包括进货、退货、统计、销售、管理等功能, 具体如下:

(1) 管理

打印图书完整库存信息, 包括编号、名称、库存量、销售量、进货单价、销售单价、供应商。

(2) 进货

根据某种图书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。输出一份进货清单并自动修改库存量, 把本次进货的信息添加到进货清单中。

(3) 退货

支持顾客把已买的书籍退还给书店, 输出一份退货清单并自动修改库存量, 把本次退货的信息添加到退货清单中。

(4) 统计

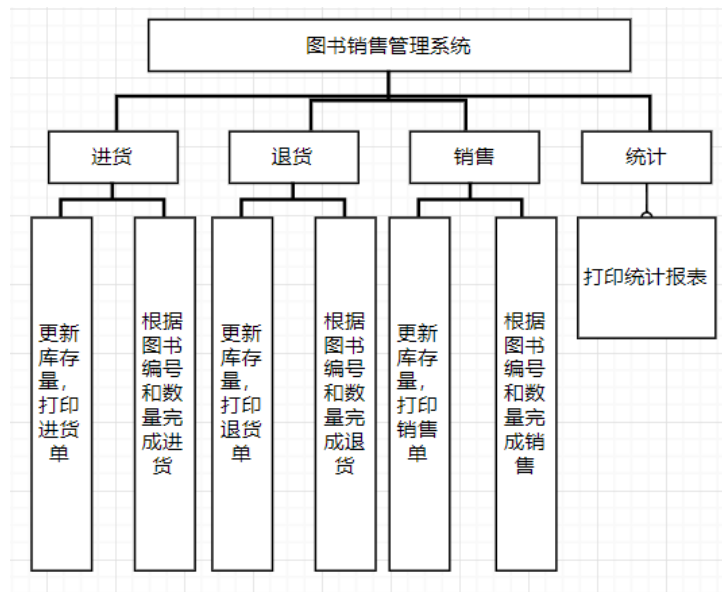
根据销售情况输出统计的报表, 具体内容为每月的销售总额、销售总量以及排行榜。

(5) 销售

输出顾客要买图书的信息, 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, 同时把这次销售的有关信息添加到销售清单中。

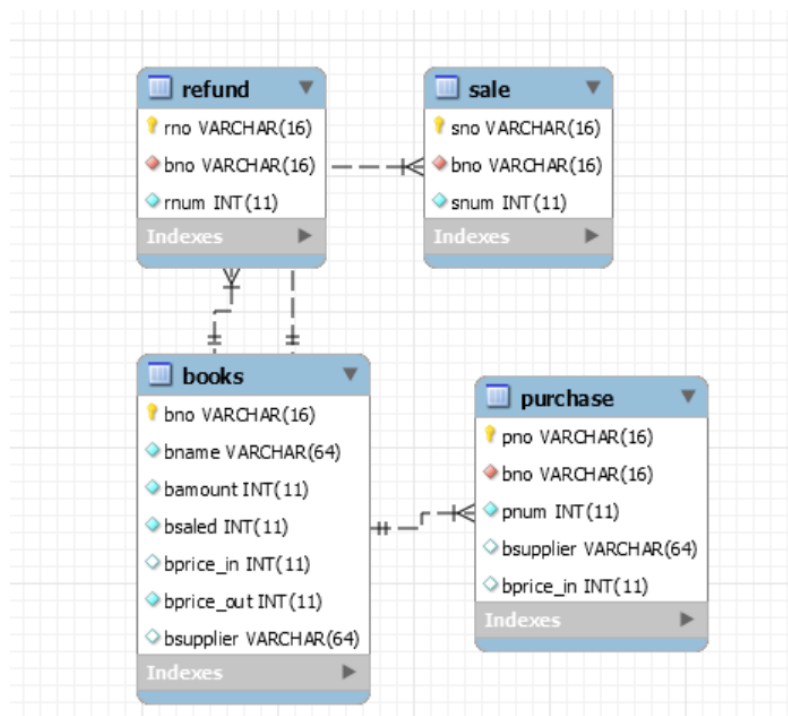
三. 功能需求分析 (3 分)

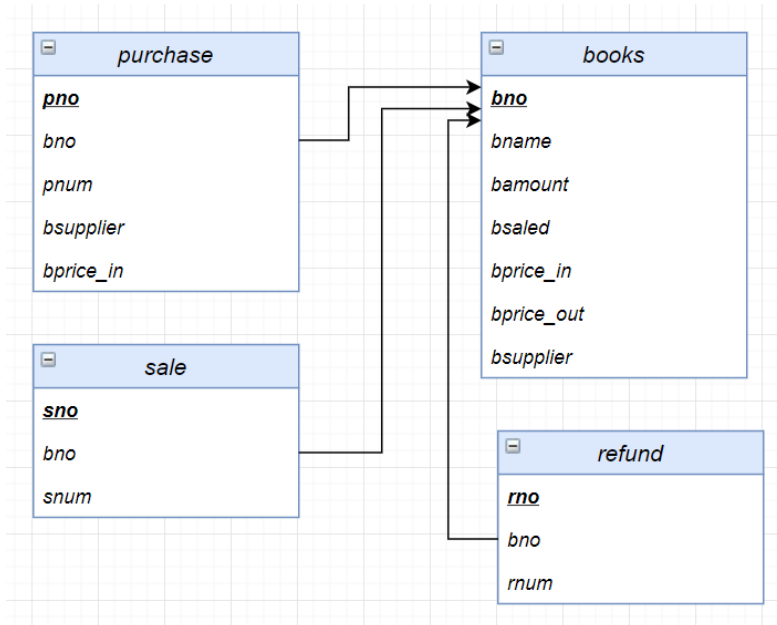
1. 系统功能模块图



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）

（1）进货（进货单号，编号，进货数量，供应商，进货单价）

purchase(pno, bno, pnum, bsupplier, bprice_in)

（2）退货（退货单号，编号，退货数量）

refund(rno, bno, rnum)

（3）销售（销售单号，编号，销售数量）

sale(sno, bno, snum)

（4）库存（编号，名称，库存量，销售量，进货单价，销售单价，供应商）

books(bno, bname, bamount, bsaled, bprice_in, bprice_out, bsupplier)

3. 数据库物理结构设计（2 分）

3.1 表的创建

（1）创建图书进货表

```

create table purchase (
    pno varchar(16) NOT NULL,
    bno varchar(16) NOT NULL,
    pnum int NOT NULL,
    bsupplier varchar(64),
    bprice_in float,
    primary key(pno),
    foreign key(bno) references books(bno)
) engine=innodb default charset=utf8;
  
```

（2）创建图书退货表

```
create table refund (  
    rno varchar(16) NOT NULL,  
    bno varchar(16) NOT NULL,  
    rnum int NOT NULL,  
    primary key(rno),  
    foreign key(bno) references books(bno)  
) engine=innodb default charset=utf8;
```

(3) 创建图书销售表

```
create table sale (  
    sno varchar(16) NOT NULL,  
    bno varchar(16) NOT NULL,  
    snum int NOT NULL,  
    primary key(sno),  
    foreign key(bno) references books(bno)  
) engine=innodb default charset=utf8;
```

(4) 创建图书库存表

```
create table books (  
    bno varchar(16) NOT NULL,  
    bname varchar(64) NOT NULL,  
    bamount int NOT NULL,  
    bsaed int NOT NULL,  
    bprice_in float,  
    bprice_out float NOT NULL,  
    bsupplier varchar(64),  
    primary key(bno)  
) engine=innodb default charset=utf8;
```

3.2 触发器的创建

(1) 进货完成时的触发器

```
create trigger purchase_update after insert on purchase
for each row
begin
    update books
    set bamount = bamount + new.pnum
    where bno=new.bno;

    update books
    set bsupplier = new.bsupplier
    where bno=new.bno;

    update books
    set bprice_in = new.bprice_in
    where bno = new.bno;
end
```

(2) 退货完成时的触发器

```
create trigger refund_update after insert on refund
for each row
begin
    update books
    set bamount = bamount + new.rnum
    where bno = new.bno;

    update books
    set bsaled = bsaled - new.rnum
    where bno = new.bno;

    update sale
    set snum = snum - new.rnum
    where bno = new.bno;
end
```

(3) 销售图书完成时的触发器

```
create trigger sale_update after insert on sale
for each row
begin
    update books
    set bamount = bamount - new.snum
    where bno = new.bno;

    update books
    set bsaled = bsaled + new.snum
    where bno = new.bno;
end
```

3.3 表的显示

(1) 显示进货清单

```
select p.pno, b.bno, b.bname, p.pnum, b.bsupplier, b.bprice_in
from purchase p join books b
on b.bno=p.bno
order by pno desc;
```

(2) 显示当前书籍库存信息

```
select *
from books
order by bno asc;
```

(3) 按时间顺序显示销售清单

```
select s.sno, s.snum, b.bno, b.bname, b.bamount, b.bsaled, b.bprice_out
from books b join sale s
on s.bno=b.bno
order by sno desc;
```

(4) 显示销量排行榜单

```
select b.bno, b.bname, b.bamount, b.bsaled, b.bprice_out
from books b
order by bsaled desc;
```

五. 系统功能的实现 (5 分)

5.1 主要功能模块的实现过程

op_func.h

```
/* 建立数据库中四个表 */
/*
创建books表, 属性有bno、bname、bamount、bsaled、bprice_in、bprice_out、bsupplier, 主码为bno;
创建 purchase表, 属性有pno、bno、pnum、bsupplier、bprice_in, 主码为pno, bno是参照 books表的外码;
创建 sale 表, 属性有 sno、bno、snum, 主码为 sno, bno 是参照 books 表的外码;
创建 refund 表, 属性有 rno、bno、rnum, 主码为 rno, bno 是参照 books 表的外码。
*/
void create_table();

/* 创建三个触发器 */
/* 创建进货完成时的触发器、退货完成时的触发器和销售图书完成时的触发器。其中, 进货完成时的触发器
purchase_update在进货后往表purchase中插入记录时被触发, 实时更新表books中图书的库存量为
bamount=bamount+new.pnum、进货单价bprice_in和供应商bsupplier; 退货完成时的触发器refund_update在
退货后往表refund中插入记录时被触发, 实时更新表books中图书的库存量为bamount=bamount+new.rnum、销售量
为bsaled=bsaled-new.snum, 并更新表sale中的销售量为snum=snum-new.rnum; 销售图书完成时的触发器
sale_update在销售图书后往表sale中插入记录时被触发, 实时更新表books中图书的库存量为bamount=bamount-
new.sum、销售量为bsaled=bsaled+new.snum。 */
void create_trigger();

/* 向 books 表中插入记录 */
bool insert_book(string bno, string bname, int bamount, int bsaled, int bprice_in, int
bprice_out, string bsupplier);

/* 向 purchase 表中插入记录 */
bool insert_purchase(string pno, string bno, int pnum, string bsupplier, int bprice_in);

/* 向 sale 表中插入记录 */
bool insert_sale(string sno, string bno, int snum);

/* 向 refund 表中插入记录 */
bool insert_refund(string rno, string bno, int rnum);

/* 返回时间戳 */
string get_time_stamp();

/* 进货 */
/* 当管理员在用户界面选择进货时, 图书销售管理系统会要求管理员输入图书的编号、进货数量、供应商名称以及进价,
之后会在表books中查询是否已有相同的图书记录, 若否, 则系统提示管理员书店中没有图书的相关信息, 需要进一步完善
进货图书的信息, 即需要管理员输入书名以及售价, 并分别向表purchase和books插入相关记录。 */
void purchase();

/* 销售图书 */
/* 当管理员在用户界面选择销售时, 系统会要求管理员输入图书的编号、销售数量, 之后会在表books中查询是否有相同
的图书记录, 若是, 则进一步判断库存量是否满足需求, 即库存量是否大于等于销售数量, 若是, 则成功销售, 并向表
sale中插入相关记录, 否则, 库存量不足, 系统提示“本书的库存数目不足”。 */
void sale();

/* 退货 */
/* 当有客户退货时, 系统会要求管理员输入退货图书的编号、退货数量, 之后会在表books中查询是否存在相同的图书记
录, 若是, 则在表sale中查询是否存在相同的图书记录, 即判断退货图书是否是本店售出的, 如果表sale中没有这本书的
退货记录, 则系统提示“书店中没有出售过这本书”, 如果表sale中记录的图书的销售数量小于退货数量, 则系统提示“书
店中没有售卖过这么多的图书”, 反之, 如果表sale中确实存在这本书的退货记录并且图书的销售数量大于等于退货数量,
则认为满足退货要求, 成功退货, 并向表refund中插入相关记录。*/
void refund();
```

```

/* 显示当前书籍库存 */
//在表books中检索所有的元组
void show_book_table();

/* 显示进货清单 */
//在表purchase中检索所有的元组
void show_purchase_list();

/* 按时间顺序显示销售清单 */
//在表sale中检索所有的元组，查询结果按销售单号降序排列
void show_sale_list_by_time();

/* 显示销量排行榜单 */
//在表sale中检索所有的元组，查询结果按销售数量降序排列
void show_sale_list_by_amount();

```

5.2 运行界面

(1) 功能菜单

```

----- 操作选项 -----
~~~~~

1 ----- 显示图书库存
2 ----- 显示进货清单
3 ----- 进货
4 ----- 出售图书
5 ----- 退货
6 ----- 按时间顺序显示销售统计情况
7 ----- 显示销售量排行榜
8 ----- 退出

~~~~~

<选择操作> 8

***** 退出管理系统。 *****

```

(2) 系统功能演示

● 进货功能

```

<选择操作> 3
<系统提示> 请输入图书编号
[图书编号] 001
<系统提示> 请输入进货的数目
[进货数目] 10
<系统提示> 请输入图书供应商名称
[供应商] 电子工业出版社
<系统提示> 请输入图书的进价
[图书进价] 79.8
<系统提示> 书店中没有这部书的相关信息，请完善以下信息：
<系统提示> 请输入此书的名称
[图书名称] 操作系统
<系统提示> 请输入图书售价
[图书售价] 99.8
<系统提示> 成功插入本书。
<系统提示> 成功进货，以 79.80 的单价，进货 10 本 操作系统，来自供应商 电子工业出版社。

```



```

<选择操作> 3
<系统提示> 请输入图书编号
[图书编号] 002
<系统提示> 请输入进货的数目
[进货数目] 15
<系统提示> 请输入图书供应商名称
[供应商] 电子工业出版社
<系统提示> 请输入图书的进价
[图书进价] 69
<系统提示> 书店中没有这部书的相关信息，请完善以下信息：
<系统提示> 请输入此书的名称
[图书名称] 线性代数
<系统提示> 请输入图书售价
[图书售价] 79
<系统提示> 成功插入本书。
<系统提示> 成功进货，以 69.00 的单价，进货 15 本 线性代数，来自供应商 电子工业出版社。

```

● 显示图书库存

```
<选择操作> 1
```

bno	bname	bamount	bsaled	bprice_in	bprice_out	bsupplier
001	操作系统	10	0	79.8	99.8	电子工业出版社
002	线性代数	15	0	69	79	电子工业出版社

● 显示进货清单

```
<选择操作> 2
```

pno	bno	bname	pnum	bsupplier	bprice_in
20210105033936	002	线性代数	15	电子工业出版社	69
20210105033915	001	操作系统	10	电子工业出版社	79.8

● 销售功能

```

<选择操作> 4
<系统提示> 请输入图书编号
[图书编号] 001
<系统提示> 请输入销售的数目
[销售数目] 5
<系统提示> 成功出售。出售 5 本编号 001 的图书。

```

```

<选择操作> 4
<系统提示> 请输入图书编号
[图书编号] 002
<系统提示> 请输入销售的数目
[销售数目] 3
<系统提示> 成功出售。出售 3 本编号 002 的图书。

```

● 显示图书库存

```
<选择操作> 1
```

bno	bname	bamount	bsaled	bprice_in	bprice_out	bsupplier
001	操作系统	5	5	79.8	99.8	电子工业出版社
002	线性代数	12	3	69	79	电子工业出版社

● 按时间顺序显示销售统计情况

```
<选择操作> 6
```

sno	snum	bno	bname	bamount	bsaled	bprice_out
20210105034039	3	002	线性代数	12	3	79
20210105034020	5	001	操作系统	5	5	99.8

- 显示销售量排行榜

```
<选择操作> 7
```

bno	bname	bamount	bsaled	bprice_out
001	操作系统	5	5	99.8
002	线性代数	12	3	79

- 退货功能

```
<选择操作> 5
<系统提示> 请输入退书的编号
[图书编号] 001
<系统提示> 请输入退书的数目
[退货数目] 3
<系统提示> 成功退货。退货 3 本编号 001 的图书。
```

- 显示图书库存

```
<选择操作> 1
```

bno	bname	bamount	bsaled	bprice_in	bprice_out	bsupplier
001	操作系统	8	2	79.8	99.8	电子工业出版社
002	线性代数	12	3	69	79	电子工业出版社

- 故障处理

```
<选择操作> 5
<系统提示> 请输入退书的编号
[图书编号] 002
<系统提示> 请输入退书的数目
[退货数目] 5
!!!!!!!!!!!!!!!!!!!!操作出错!!!!!!!!!!!!!!!!!!!!

[错误报告] 书店中没有售卖过这么多图书。
<系统提示> 请重试。
```

```
<选择操作> 5
<系统提示> 请输入退书的编号
[图书编号] 003
<系统提示> 请输入退书的数目
[退货数目] 1
!!!!!!!!!!!!!!!!!!!!操作出错!!!!!!!!!!!!!!!!!!!!

[错误报告] 书店中没有出售过这本书。
<系统提示> 请重试。
```

```
<选择操作> 4
<系统提示> 请输入图书编号
[图书编号] 002
<系统提示> 请输入销售的数目
[销售数目] 20
!!!!!!!!!!!!!!!!!!!!操作出错!!!!!!!!!!!!!!!!!!!!

[错误报告] 本书的库存数目不足。
<系统提示> 请重试。
```

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。

(1) 使用基本 SQL 语句创建数据库和表;

(2) 设计物理结构创建表时添加完整性约束, 定义表的主码和外码体现了参照完整性, 创建触发器体现了用户自定义完整性;

- 主码

超码是一个或多个属性的集合, 可以使我们在一个关系中唯一地标识一个元组, 但是超

码中可能包含无关紧要的属性，因此，我们通常只对任意真子集都不能成为超码的最小超码感兴趣，并称之为候选码，由于几种不同的属性集都可以做候选码的情况是存在的，因此，我们用主码代表被选中的用来在一个关系中区分不同元组的候选码。在图书销售管理系统中，关系 **books** 的主码为 **bn**，**pn** 是关系 **purchase** 的主码，**rn** 是关系 **refund** 的主码，**sn** 是关系 **sale** 的主码。

- 外码

一个关系模式 r_1 可能在它的属性中包括另一个关系模式 r_2 的主码，这个属性在 r_1 上称作参照 r_2 的外码，关系 r_1 也被称为外码依赖的参照关系， r_2 叫做外码的被参照关系，在图书销售管理系统中，表 **purchase** 中的属性 **bn**、表 **refund** 中的属性 **bn** 和表 **sale** 中的属性 **bn** 都是参照表 **books** 的外码，因此，**purchase**、**refund** 和 **sale** 被称为外码依赖的参照关系，**books** 也被叫做外码的被参照关系，在输入或删除记录时，参照完整性保持表之间已经定义的关系，确保键值在所有表中一致，一致性要求不能引用不存在的值，如果键值改变了，那么在整个数据库中对该键值的所有引用就要进行一致的更改。

- 触发器

数据完整性分为 4 类：实体完整性、域完整性、参照完整性和用户自定义完整性，其中，用户自定义完整性可以借助触发器实现，触发器可以扩展约束、默认值等的完整性检查规则，用来实现未被 SQL 约束机制指定的某些完整性约束，当对数据库作修改时，只要指定的事件发生，并且相应的条件满足，触发器自动被系统执行，对用户发出警报或者自动开始执行某项任务。在图书销售管理系统中，创建了三个触发器，分别在向表 **purchase**、**refund** 和 **sale** 中插入记录时被触发，实时联动更新其他表中的对应元组。