

数据库课程设计报告

开发环境与开发工具

开发语言：**C++、MySQL**

操作系统：**macOS Mojave Version 10.14.5**

开发工具：**Xcode IDE、iTerm 命令行终端**

终端环境：**iTerm + zsh、Xcode 终端调试台**

MySQL Server 版本：**5.7.25 MySQL Community Server (GPL)**

系统需求分析

当今的图书销售行业有数据量大、数据结构复杂、数据变化和流动频繁的特点，加上如今的网商平台越来越发达，各行各业的电子化信息化管理的趋势非常明显，图书销售的管理将是一个非常具有实用性的工具系统。有了这样的系统，我们可以通过事务命令的封装、提供比较友好的用户界面，使管理人员便于使用和操作，便捷地实现以往繁琐的图书销售管理操作。

设计实现中要完成的数据库管理系统，要实现几个功能模块，方便管理人员对图书的销售进行比较全面的管理。具体来说大致具有以下功能：

- **基本管理**

- 打印图书完整库存信息，包括编号、名称、作者、库存量、销售量、单价、供应商、供应商单价；

- **进货**

- 可以显示当前所有图书的库存量、销售情况、供应商单价情况等基本信息；
- 支持管理员通过信息录入对具体图书进行进货操作，自动更新数据库，打印进货单；

- **退货**

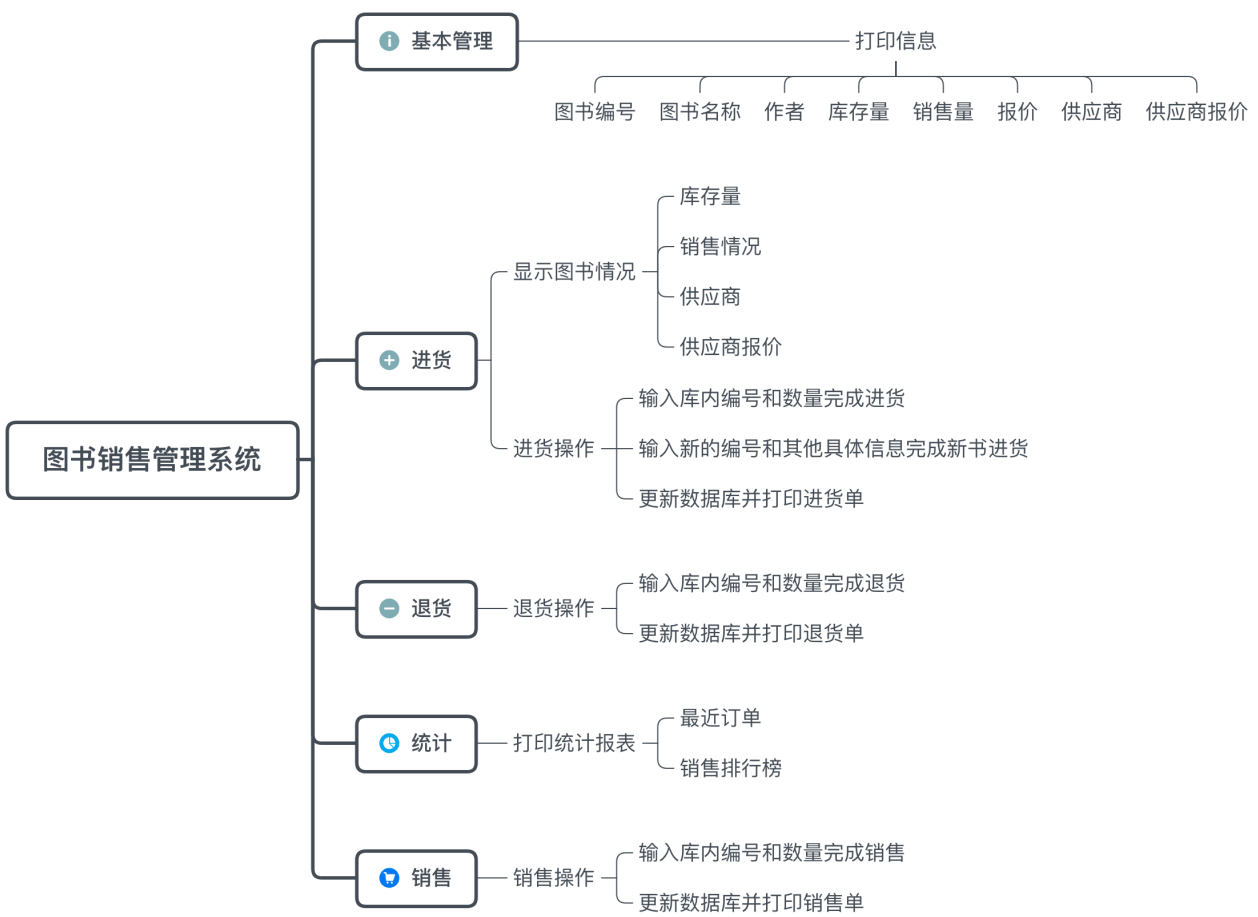
- 支持管理员通过信息录入对具体图书进行退货操作，自动更新数据库，打印退货单；

- **统计**

- 可以根据销售情况输出统计报表，具体内容有每月销售总额、总量和排行榜；

- 销售
 - 支持管理用通过信息录入输入用户要购买书籍的信息，自动更新数据库，打印购买情况；

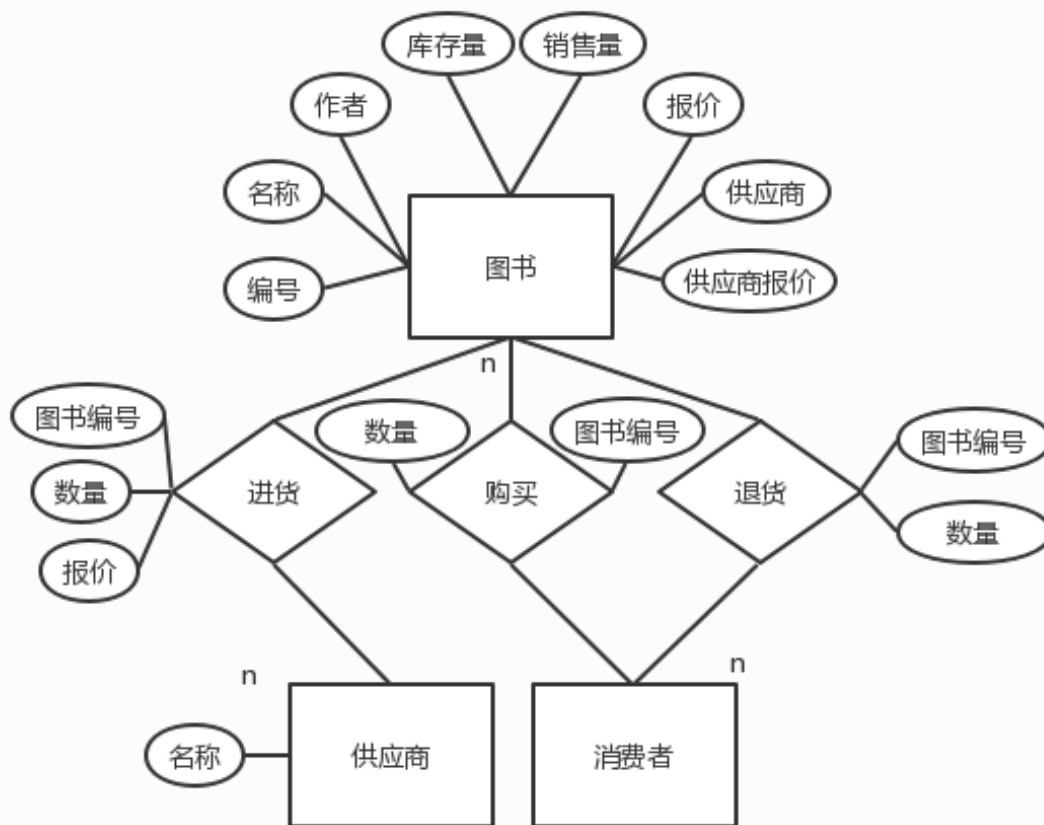
系统功能模块图



系统设计

数据概念结构设计

根据需求分析和系统设计，管理系统的系统 ER 图如下：



数据库关系模式设计

加粗下划线的属性为主码，加粗斜体的属性为外码

- 图书库存（编号，名称，作者，库存量，销售量，单价，供应商，供应商单价）
books (**bno**, bname, bauthor, bamount, bsaled, bprice, bsupplier, boffer)
- 图书进货（订单号，编号，数量，供应商，供应商单价）
purchase (**pno**, ***bno***, pnun, bsupplier, boffer)
- 图书销售（销售单号，编号，数量）
sale (**sno**, ***bno***, snum)
- 图书退货（退货单号，编号，数量）
refund (**rno**, ***bno***, rnum)

数据库物理结构设计

该系统的数据库物理结构设计通过创建表的各个操作的 SQL 语句命令呈现。

数据库创建

```
1 create database bsms character set=utf8; /* bsm for book sales
   management */
2 use bsms;
```

表的创建

创建图书库存表

```
1 create table books (
2   bno varchar(16) NOT NULL,
3   bname varchar(64) NOT NULL,
4   bauthor varchar(64),
5   bamount int NOT NULL,
6   bsaled int NOT NULL,
7   bprice int NOT NULL,
8   bsupplier varchar(64),
9   boffer int,
10  primary key(bno),
11  index (bno)
12 ) engine=innodb default charset=utf8;
13
14 /* test insert */
15 insert into
   books(bno,bname,bauthor,bamount,bsaled,bprice,bsupplier,boffer)
16 value("testnum","testname","testauth",99,9,9999,"testpbis",1);
```

创建图书进货表

```
1 create table purchase (
2   pno varchar(16) NOT NULL,
3   bno varchar(16) NOT NULL,
4   pnum int NOT NULL,
5   bsupplier varchar(64),
6   boffer int,
7   primary key(pno),
8   foreign key(bno) references books(bno)
9 ) engine=innodb default charset=utf8;
10
11 /* test insert */
12 insert into purchase(pno,bno,pnum,bsupplier,boffer)
13 value("testpurchase","testnum",123,"testsupplier",99);
```

创建销售表

```

1  create table sale (
2  sno varchar(16) NOT NULL,
3  bno varchar(16) NOT NULL,
4  snum int NOT NULL,
5  primary key(sno),
6  foreign key(bno) references books(bno)
7  ) engine=innodb default charset=utf8;
8
9  /* test insert */
10 insert into sale(sno,bno,snum)
11 value("testsale","testnum",2);

```

创建退货表

```

1  create table refund (
2  rno varchar(16) NOT NULL,
3  bno varchar(16) NOT NULL,
4  rnum int NOT NULL,
5  primary key(rno),
6  foreign key(bno) references books(bno)
7  ) engine=innodb default charset=utf8;
8
9  /* test insert */
10 insert into refund(rno,bno,rnum)
11 value("testrefund","testnum",1);

```

完整性约束触发器

进货操作之后的相关更新

```

1  /* delimiter // */
2  /* don't have to and can't user delimiter in C API */
3  create trigger purchase_update
4  after insert on purchase
5  for each row
6  begin
7      update books set bamount=bamount+new.pnum where
bno=new.bno;
8      update books set bsupplier=new.bsupplier where bno=new.bno;
9      update books set boffer=new.boffer where bno=new.bno;
10 end
11 /* //
12 delimiter ; */

```

销售操作之后的相关更新

```

1 create trigger sale_update
2 after insert on sale
3 for each row
4     begin
5         update books set bamount=bamount-new.snum where bno=new.bno;
6         update books set bsaied=bsaied+new.snum where bno=new.bno;
7     end

```

退货操作之后的相关更新

```

1 create trigger refund_update
2 after insert on refund
3 for each row
4     begin
5         update books set bamount=bamount+new.rnum where bno=new.bno;
6         update books set bsaied=bsaied-new.rnum where bno=new.bno;
7         update sale set snum=snum-new.rnum where bno=new.bno;
8     end

```

表格查询与连接

显示所有库存图书

```

1 select *
2 from books
3 order by bno asc;

```

显示所有进货信息

```

1 select p.pno,b.bno,b.bname,p.pnum,b.bsupplier,b.boffer
2 from purchase p
3 join books b
4     on b.bno=p.bno
5 order by pno desc;

```

按照时间降序显示销售单详细信息

```

1 select
2     s.sno,s.snum,b.bno,b.bname,b.bauthor,b.bamount,b.bsaied,b.bprice
3 from books b
4 join sale s
5     on s.bno=b.bno
6 order by sno desc;

```

按照销售量降序显示图书销售情况

```
1 select b.bno,b.bname,b.bauthor,b.bamount,b.bsaled,b.bprice
2 from books b
3 order by bsaled desc;
```

系统功能的实现

对系统运行的大致流程和其相关功能的代码实现进行分析。

源码中按照函数注释格式给出了大致的英文注释，这里为了方便分析和补充说明重新添加注释。

主要功能模块的实现过程简述

主要框架

先看到整个系统的主体框架：

```
1 int main() {
2     sm_welcome(); // 欢迎页面打印 系统起始界面
3     // 数据库连接 运行时需要先行手动在MySQL上创建数据库 bsms
4     mysql_init(&mysql);
5     if (NULL == mysql_real_connect(&mysql, "localhost", username,
6     password, database, 3306, 0, 0)) {
7         sm_error(mysql_error(&mysql));
8         return -1;
9     }
10    // 数据库初始化：创建表格、触发器约束
11    init();
12    // 循环获取用户操作，各对应操作函数进行封装处理
13    bool infinite = true;
14    while (infinite) {
15        sm_menu(); // 菜单打印
16        string tcmd = sm_get_cmd(); // 指令获取和处理
17        int cmd = tcmd[0] - '0';
18        switch (cmd) {
19            case 0:sm_bye();return 0;break;
20            case 1:book();break;
21            case 2:stock();break;
22            case 3:purchase();break;
23            case 4:sale(); break;
24            case 5:refund();break;
25            case 6:sale_time();break;
26            case 7:sale_ppl();break;
27            default:sm_error("Invalid command input
detected");break;
        }
    }
}
```

```

28         sleep(1); // 每次都挂起一秒
29             // 提示用户注意操作后的系统信息以免直接被菜单覆盖
30             // 防止生成的年月日时分秒时间戳重复
31     }
32     mysql_close(&mysql);
33     return 0;
34 }

```

主体的框架非常简单清晰，连接数据库成功后不断阻塞式地获取用户指令进行处理，利用 switch case 代码段进行功能划分和函数调用，接下来看到一些功能性函数库的基本信息，以便展示系统底层函数功能和实现支持。

功能库

Global.h

该头文件保存了全局性的可修改的变量和宏定义，方便统一管理、使用和修改。

```

1  #ifndef Global_h
2  #define Global_h
3
4  #include <iomanip>
5  #include <iostream>
6  #include <mysql.h>
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <string.h>
10
11  using namespace std;
12
13  MYSQL mysql;
14  char username[16] = "root";
15  char password[16] = "database2019";
16  char database[16] = "bsms";
17
18  #endif /* Global_h */

```

SystemMsg.h

该头文件申明定义了一些系统命令行用户界面的输出显示设置和各个层次都可以使用的一些显示类的功能性函数。

下面使用函数申明列表进行解释。

```

1  // 系统界面打印辅助函数，以 decorator 作为左右占位符号将 msg 内容在屏幕上居中输出
2  void sm_center(char decorator, string msg);
3

```



```

4 // 系统消息提示统一模板, 将提示信息 msg 在命令行界面以统一格式输出, 用于输出正
  常操作后的提示信息
5 void sm_alert(const char * msg);
6
7 // 系统消息提示统一模板, 将错误信息 msg 在命令行界面以统一格式输出, 用于输出异
  常操作后的错误提示
8 void sm_error(const char * msg);
9
10 // 系统表格打印辅助函数, 模仿 MySQL 命令行终端的表格打印方法打印表格中的横向分
   隔符
11 void sm_display_table_line(vector<int> widths, int col);
12
13 // 系统表格打印辅助函数, 打印整个表格信息
14 void sm_display_table_func(vector<vector<string>> ele, vector<int>
   widths, int col, unsigned long long row);
15
16 // 系统表格打印函数, 将 mysql 查询得到的结果进行解析保存, 调用辅助函数完成打印
17 // 获取每个数据属性信息并保存在二维vector中, 获取每列数据最大长度并保存作为底层
   表格打印函数参数
18 void sm_display_table(MYSQL_RES * res);
19
20 // 系统获取用户指令统一模板, 显示获取指令提示信息并接收用户输入
21 string sm_get_cmd();
22
23 // 系统获取字符串类型数据信息统一模板, 显示提示输入 msg 和属性关键字 key 并接
   受用户录入的数据
24 string sm_get_info(const char * msg, const char * key);
25
26 // 系统获取整数类型数据信息统一模板, 显示提示输入 msg 和属性关键字 key 并接受
   用户录入的数据
27 int sm_get_amount(const char * msg, const char * key);
28
29 // 系统欢迎界面打印函数, 会输出 ascii 字符拼接海报体的 BOOK SALE 字样
30 void sm_welcome();
31
32 // 系统菜单打印函数, 会输出系统功能列表以及对应的代表关键字方便用户操作
33 void sm_menu();
34
35 // 系统关闭退出信息打印函数, 模仿 MySQL 命令行终端输出海报体 Bye
36 void sm_bye();

```

MySQLOpr.h

该头文件将插入数据的操作进行了封装, 提供插入数据 API 接受相关数据变量完成数据插入。

```

1 // 以其中一个函数为例, 初始化查询字符串并将接收到的参数利用 sprintf 整合其中
2 // 进行查询并完成错误处理和结果返回, 更详细有针对性的错误信息会在上层调用中进行
   打印
3 bool insert_book(string bno, string bname, string bauthor, int
   bamount, int bsaled, int bprice, string bsupplier, int boffer) {
4     char querys[256];

```

```

5     sprintf(querys, "insert into books(bno,\
6                                     bname,\
7                                     bauthor,\
8                                     bamount,\
9                                     bsaled,\
10                                    bprice,\
11                                    bsupplier,\
12                                    boffer) value\
13
14    (\"%s\", \"%s\", \"%s\", %d, %d, %d, \"%s\", %d);\
                                     \", bno.c_str(),
bname.c_str(),
15                                     bauthor.c_str(), bamount,
bsaled,
16                                     bprice, bsupplier.c_str(),
boffer);
17    if (0 == mysql_query(&mysql, querys)) {
18        return true;
19    }
20    sm_error(mysql_error(&mysql));
21    return false;
22 }
23
24 bool insert_purchase(string pno, string bno, int pnum, string
bsupplier, int boffer);
25
26 bool insert_sale(string sno, string bno, int snum);
27
28 bool insert_refund(string rno, string bno, int rnum);

```

系统功能库

SystemOpr.h

```

1  // 根据前述数据库物理结构设计中的表格、触发器创建语句完成表格和触发器创建
2  void init();
3
4  // 获取年月日时分秒14位时间戳作为唯一的订单、进货单、退货单标识
5  // 因为每次功能执行都会挂起一秒，所以不会出现重复情况
6  string get_time_stamp();
7
8  // 打印图书列表，查询所有图书，按照编号升序排列并调用功能函数完成打印
9  void book();
10
11 // 进货列表打印，查询进货列表并对图书列表进行join拼接，打印详细进货信息
12 void stock();
13
14 // 完成进货操作
15 // 要求用户输入要进货的书编号、数量、选择供应商和对应报价
16 // 如果书不存在，即说明是新书进货，要求用户录入完整书籍信息完成插入再进行下一步

```

```
17 // 如果书已经存在，或者已经完成插入，则按照编号和数量对书库、进货单两个表格进行更新
18 void purchase();
19
20 // 完成销售操作
21 // 要求用户输入要售卖的书编号、数量
22 // 判断书库中是否有该书，如果有数量是否足够
23 // 根据实际情况完成正常销售或是打印错误信息建议管理员进货
24 void sale();
25
26 // 完成退货操作
27 // 要求用户输入客户要退货的书编号、数量
28 // 判断该书是否在书库中，如果有是否售卖出该数量以上的书
29 // 根据实际情况完成退货或是打印错误信息提醒该书可能不是本店售出
30 void refund();
31
32
33 // 根据订单完成时间（即订单号）从新到旧打印销售单
34 void sale_time();
35
36 // 根据图书热度、受欢迎程度（总销售量）从高到低打印图书销售情况
37 void sale_ppl();
```

运行界面

下面演示正常操作下的用户操作流程，因为错误处理的运行分支过多，这里不进行详细演示。实际系统中已对各个步骤可能产生的错误进行妥善处理和用户提示。

系统基本界面

欢迎界面、菜单、退出界面

```
BOOKS SALES MANAGER

~~~~~ Welcome to the Book Sales Management System ~~~~~

---- Please enter the command related to the operations in [?] ----

[Opr'Menu]

    [0] -- Exit.
    [1] -- Show the book(s) in the store.
    [2] -- Show the stocking list of the store.
    [3] -- Purchase book(s) from the supplier(s).
    [4] -- Sale book(s) to a customer.
    [5] -- Refund to a customer with book(s).
    [6] -- Show the saling list of the store in time order.
    [7] -- Show the saling list of the store in popularity order.

[CommandK] 0

      TT  TT  TT  TT  TT
     HH  HH  HH  HH  HH
    LL  LL  LL  LL  LL
```

管理功能演示

图书进货

```
---- Please enter the command related to the operations in [?] ----

[Opr'Menu]

[0] -- Exit.
[1] -- Show the book(s) in the store.
[2] -- Show the stocking list of the store.
[3] -- Purchase book(s) from the supplier(s).
[4] -- Sale book(s) to a customer.
[5] -- Refund to a customer with book(s).
[6] -- Show the saling list of the store in time order.
[7] -- Show the saling list of the store in popularity order.

[CommandK] 3
[SysAlert] Please input the NO. of the Book
[bno      ] 2019
[SysAlert] Please input the Number of the Book to Purchase
[pnum     ] 6
[SysAlert] Please input the Supplier of the Book
[supplier] testspli
[SysAlert] Please input the Price of the Book from the Supplier
[boffer   ] 998
[SysAlert] It's a new book in the store,
           please complete the following information
[SysAlert] Please input the Name of the Book
[pname    ] sellingbookiebookie
[SysAlert] Please input the Author of the Book
[bauthor  ] authorrrr
[SysAlert] Please input the Price of one copy of this Book
[bprice   ] 1
[SysAlert] Book Inserted Successfully!
[SysAlert] Book Purchased Successfully!
[SysAlert] Purchase 6 copies of bookiebookie from testspl at the cost of $5988.
```

图书销售

---- Please enter the command related to the operations in [?] ----

[Opr'Menu]

- [0] -- Exit.
- [1] -- Show the book(s) in the store.
- [2] -- Show the stocking list of the store.
- [3] -- Purchase book(s) from the supplier(s).
- [4] -- Sale book(s) to a customer.
- [5] -- Refund to a customer with book(s).
- [6] -- Show the saling list of the store in time order.
- [7] -- Show the saling list of the store in popularity order.

[CommandK] 4

[SysAlert] Please input the NO. of the Book

[bno] 123

[SysAlert] Please input the Number of the Book to Sale

[snum] 2

[SysAlert] Book Saled Successfully

[SysAlert] Sold 2 copies of 123.

[CommandK] 4

[SysAlert] Please input the NO. of the Book

[bno] 2019

[SysAlert] Please input the Number of the Book to Sale

[snum] 999999

!!!!!!!!!!!!!!!!!!!! Errors occur during operation! !!!!!!!!!!!!!!!!!!!!!

[ErrorMsg] There aren't enough book(s) to sale!

[SysAlert] Please check and try again

[SysAlert] Please purchase some extra books if needed with the command [3]

图书退货

[CommandK] 5

[SysAlert] Please input the NO. of the Book

[bno] 123

[SysAlert] Please input the Number of the Book to Refund

[rnum] 1

[SysAlert] Book Refund Successfully!

[SysAlert] Refund 1 copies of 123.

```

[CommandK] 5
[SysAlert] Please input the NO. of the Book
[bno      ] 2019
[SysAlert] Please input the Number of the Book to Refund
[rnum     ] 1
!!!!!!!!!!!!!!!!!!!! Errors occur during operation! !!!!!!!!!!!!!!!!!!!!!

[ErrorMsg] There aren't so many books saled in the store!
[SysAlert] Please check and try again

[SysAlert] Maybe the books are not saled in this store

```

打印功能演示

书库打印

```

[CommandK] 1
+-----+-----+-----+-----+-----+-----+-----+
| bno | bname      | bauthor  | bamount | bsaled | bprice | bsupplier | boffer |
+-----+-----+-----+-----+-----+-----+-----+
| 123 | testbook   | testauthor | 99      | 1      | 25     | testsupplier | 20     |
| 2019 | bookiebookie | authorrr  | 1       | 5      | 1      | testspl     | 998    |
+-----+-----+-----+-----+-----+-----+-----+

```

```

[CommandK] 2
+-----+-----+-----+-----+-----+-----+
| pno      | bno | bname      | pnun | bsupplier | boffer |
+-----+-----+-----+-----+-----+-----+
| 20190629140736 | 2019 | bookiebookie | 6    | testspl   | 998    |
| 20190629135900 | 123  | testbook     | 100  | testsupplier | 20     |
+-----+-----+-----+-----+-----+-----+

```

```

[CommandK] 6
+-----+-----+-----+-----+-----+-----+-----+
| sno      | snun | bno | bname      | bauthor  | bamount | bsaled | bprice |
+-----+-----+-----+-----+-----+-----+-----+
| 20190629141845 | 5    | 2019 | bookiebookie | authorrr  | 1       | 5      | 1      |
| 20190629141305 | 1    | 123  | testbook     | testauthor | 99      | 1      | 25     |
+-----+-----+-----+-----+-----+-----+-----+

```

```

[CommandK] 7
+-----+-----+-----+-----+-----+-----+
| bno | bname      | bauthor  | bamount | bsaled | bprice |
+-----+-----+-----+-----+-----+-----+
| 2019 | bookiebookie | authorrr  | 1       | 5      | 1      |
| 123  | testbook     | testauthor | 99      | 1      | 25     |
+-----+-----+-----+-----+-----+-----+

```

总结

本课程设计中用到的《数据库系统》理论课概念与知识

- 使用基本 SQL 语句和命令创建数据库和数据库中的表格；
- 运用了索引，在创建最主要的 books 书库表格的时候添加了索引 bno；
- 使用应用比较广泛的 UTF8 编码字符集创建数据库和表，保证了显示的兼容性；
- 设计物理结构创建表格时运用基本数据库知识添加完整性约束，定义了表格主码、外码；
- 设计物理结构的时候设计创建了数据库中的触发器 trigger，对相关的表格进行联动更改；
- 运用 procedure 和 transaction rollback 的思想，对创建数据库时的基本数据合法性进行了检查，如果不合法将进行事务回滚，取消创建操作；