

浙江大学

数据库系统实验报告

作业名称: 实验 5 数据库程序设计

姓 名: 朱理真

学 号: 3190101094

电子邮箱: 3190101094@zju.edu.cn

联系电话: 19817862976

指导老师: 孙建伶

2021 年 4 月 30 日

实验 5 数据库程序设计

一、 实验目的

1. 掌握数据库应用开发程序设计方法。

二、 系统需求

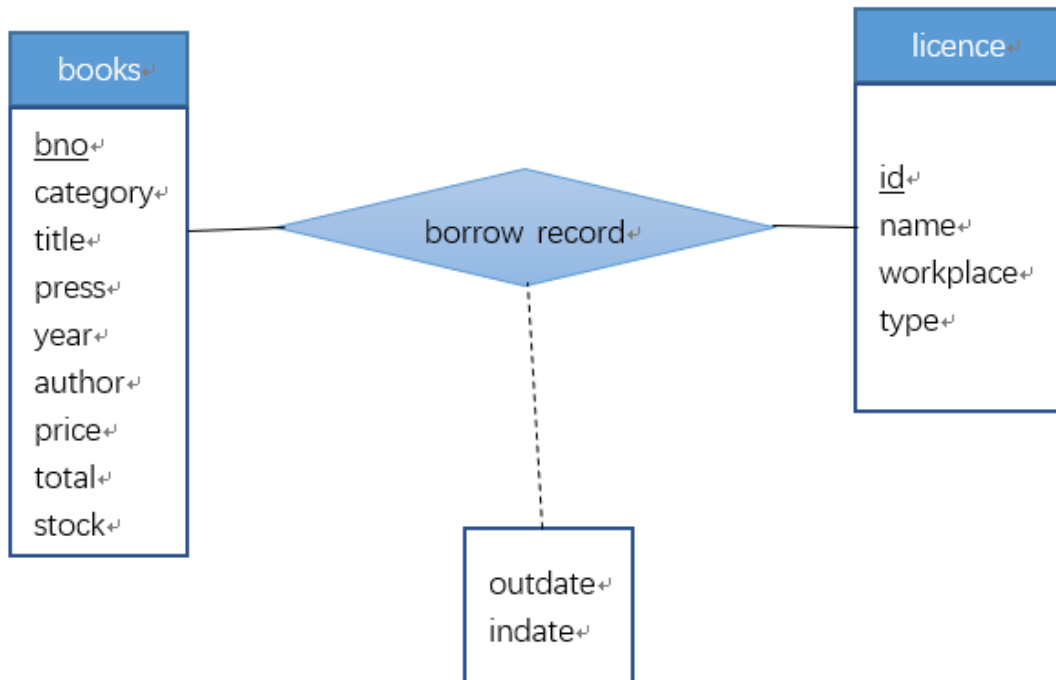
Windows xp/7/10, 32 位及以上

三、 实验环境

1. gcc 编译器: MinGW.org GCC-8.2.0-5
2. MySQL8.0
3. ODBC32 位 (与 MySQL 链接)

四、 系统设计及实现

1. 实体之间的关系 E-R 图



2. 数据库逻辑结构设计

定义:

books(bno,category,title,press,year,author,price,total,stock)

licence(id,name,workplace,type)

borrow_record(bno,id,outdate,indate)

DDL:

```
create table borrow_record
(
    bno varchar(30),
    id int,
```

```

        Outdate datetime,
        Indate datetime,
        Primary Key(bno,id,Outdate),
        foreign key(bno) references books(bno),
        foreign key(id) references licence(id)
    );

create table licence(
    id int,
    name varchar(30),
    workplace varchar(30),
    type varchar(30),
    Primary Key(id),
    check (type in ('teacher','student','administrator'))
);

create table books(
    bno varchar(30),
    Category varchar(30),
    Title varchar(30),
    Press varchar(30),
    Year int,\
    Author varchar(30),
    Price float(2),
    Total int,
    Stock int,
    Primary Key(bno),
    check (Stock>=0),
    check (Total>=Stock),
    check (Price>0)
);

```

3. 程序运行结果场景以及截图说明

程序开始时，是欢迎界面和提示界面，如下图

```
Welcome to mini library system!
Author: Zhu Lizhen
-----
Number of Options:

0.help
1.create a licence
2.borrow a book
3.delete a licence
4.show all books
5.find borrowed book
6.return a book
7.add a new book
8.add book benches
9.search book
10.update a licence
and enter any other words to quit

enter your option:
█
```

弹出 `enter your option` 时，即可按照提示键入菜单中的数字来选择所想执行的操作。根据菜单选项的不同，后续也会有不同的输入要求，最终产生输出。下面逐个分析。除成功示例外，还有异常输入处理，相似的异常输入不再重复赘述。

（其他辅助功能：帮助：0，查看所有图书：4，退出：其他任意键）

a. 图书入库和修改：

单本入库

键入 7，可以单本入库：

输入格式如下：

（bno,category,title,press,year,author,price,total）

若 bno 不在数据库中，则插入一本新书，其库存 `stock=total`

示例：

键入 7，首先输出插入前所有图书的信息

enter your option:

7

before insert

```
(1,cs,computer organization,zju,2021,zlz,100.199997,3,1)
(10,cs,co,mp,2010,zlz,21.000000,6,6)
(1024,cs,db,zju,2020,zlz,38.000000,2,2)
(2,CS,computer architecture,zju,2021,ljy,100.000000,3,1)
(9,CS,co,mp,1997,david,137.000000,5,5)
(bb,cs,db,zju,2020,zlz,38.000000,2,2)
(book_2,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_22,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_23,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_232,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_2362,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_32,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_3232,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_327,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_362,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_36232,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_3627,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_4,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_5,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_6,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_667,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_67,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_f1,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,1,1)
(book_f2,Computer Science,Computer Architecture,xxx,2005,xxx,90.000000,2,2)
(book_f3,Computer Science,Computer Architecture,xxx,2006,xxx,90.000000,2,2)
(book_f4,Computer Science,Computer Architecture,xxx,2007,xxx,90.000000,2,2)
(book_f5,Computer Science,Computer Architecture,xxx,2008,xxx,90.000000,2,2)
(book_no_1,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
```

提示键入:

```
puts the new book information in the followed format:
(book_id,category,title,press,year,author,price,total)
```

接下来我们输入信息:

```
(book_56,cs,database design,zju,2010,sjl,120,3)
```

若插入成功，则输出入库后的图书信息

```

after insert
(1,cs,computer organization,zju,2021,zlz,100.199997,3,1)
(10,cs,co,mp,2010,zlz,21.000000,6,6)
(1024,cs,db,zju,2020,zlz,38.000000,2,2)
(2,CS,computer architecture,zju,2021,ljy,100.000000,3,1)
(9,CS,co,mp,1997,david,137.000000,5,5)
(bb,cs,db,zju,2020,zlz,38.000000,2,2)
(book_2,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_22,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_23,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_232,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_2362,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_32,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_3232,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_327,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_362,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_36232,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_3627,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_4,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_5,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_56,cs,database design,zju,2010,sjl,120.000000,3,3)
(book_6,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)
(book_667,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_67,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,2,2)
(book_f1,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,1,1)
(book_f2,Computer Science,Computer Architecture,xxx,2005,xxx,90.000000,2,2)
(book_f3,Computer Science,Computer Architecture,xxx,2006,xxx,90.000000,2,2)
(book_f4,Computer Science,Computer Architecture,xxx,2007,xxx,90.000000,2,2)
(book_f5,Computer Science,Computer Architecture,xxx,2008,xxx,90.000000,2,2)
(book_no_1,Computer Science,Computer Architecture,xxx,2010,xxx,99.000000,2,2)

success!

```

可以看到，我们的图书被成功地插入到了数据表中。

修改图书

若图书号 bno 已存在，则对应的输入会变成对该本书的修改示例：

插入前和插入后的图书信息输出基本和单本入库一致，输入有所不同：

```

puts the new book information in the followed format:
(book id,category,title,press,year,author,price,total)

```

输入一个图书号已存在的图书信息：

```

(1,cs,computer organization,zju,2021,zlz,120,3)

```

就会提示是在更新这本书，并要求输入额外的库存更新信息：

```

You are updating a book.And what's the new stock?
1

```

键入完成后，即更新成功

批量图书入库

在菜单中键入 8，进入批量图书入库模式

根据提示键入文件名，即可完成导入

下面是完整的交互过程

```
enter your option:
8

before insert
no data

enter the filename
book

after insert
(bf1,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,1,1)
(bf2,Computer Science,Computer Architecture,xxx,2005,xxx,90.000000,2,2)
(bf3,Computer Science,Computer Architecture,xxx,2006,xxx,90.000000,2,2)
(bf4,Computer Science,Computer Architecture,xxx,2007,xxx,90.000000,2,2)
(bf5,Computer Science,Computer Architecture,xxx,2008,xxx,90.000000,2,2)

success!
```

b. 借书:

在菜单中键入 2，即选择借书选项

输入:

键入 id

键入 bno

示例:

```
enter your option:
2
your id:
1
book id:
bf1

success!
```

验证，查看该用户已借书籍:

```
enter your option:
5
your id:
1
(bf1,Computer Science,Computer Architecture,xxx,2004,xxx,90.000000,1,0)

success!
```

可以看到该书成功被借出，且库存减 1

异常处理：

id 不存在

```
enter your option:
2
your id:
2
book id:
bf1
id not found
```

书号不存在：

```
enter your option:
2
your id:
1
book id:
1
bno:1
book not found!
```

重复借一本书（未还）：

```
enter your option:
2
your id:
1
book id:
bf1
your have borrowed this book!
```

书籍无库存

```
enter your option:
2
your id:
2
book id:
bf1
This book has been borrowed out.
You can borrow the book at 2021-6-30 11:49:55 at the soonest
```

c. 还书：

在菜单中键入 6，即选择还书选项

输入：

键入 id

键入 bno

示例：


```
enter your option:
6
your id:
1
book id:
bf1
success!
```

验证，查看该用户已借书籍：

```
enter your option:
5
your id:
1
no data
success!
```

说明书已正确归还

异常处理：

id 不存在；

书号不存在；

没有借过要还的书：

```
enter your option:
6
your id:
2
book id:
bf1
0
your haven't borrowed this book yet!
```

d. 借书证管理：

新建借书证：选择菜单 1

创建一个新的借书证，需填入必要的信息，格式如下。注意每次键入后需要回车。

键入 id

键入 name

键入 workplace

键入 licence type(根据提示选数字)

1.stdudent

2.teacher

3.administrator

若 id 未注册过，且其他参数不为空，则可以注册成功

下面是一个成功的新建图书证操作：

```
enter your option:
1
new id:
11
put your name:
diona
your work place:
ZJU
licence type:(default student)
1.student
2.teacher
3.administrator
1
success!
```

异常处理:

id 已存在:

```
enter your option:
1
new id:
1
id exist already!
```

无效输入:

```
enter your option:
1
new id:
3
put your name:

invalid input!
```

修改借书证:

菜单中选择 10，即可修改借书证

输入格式与新建借书证类似，但可以键入回车表示不修改某一个参数

示例:

```
enter your option:
10
new id:
1
please fill the update infomations.

you can press enter to pass some arguments.

name:

workplace:
yu quan
licence type:(default student)
1.student
2.teacher
3.administrator

success!
```

在本例中，跳过了对 name 和 licence type 的修改，只修改了 work place，结果运行成功。

异常处理：id 不存在

删除借书证：

菜单选择 3，删除借书证

键入要删除的 id 即可，如下

```
enter your option:
3
your id:
1
success!
```

异常处理：id 存在；无效输入

e. 图书查询：

选择菜单 9 进行图书的查询。

根据提示键入相应的查询条件。

直接回车，可跳过对某一参数的搜索

对于 year, price 这两种数值型属性，可以使用区间查询，即使用 “,” 来分隔两个参数
(注意，区间查询是闭区间)

示例：

```
enter your option:
9
please fill the query conditions.

you can press enter to pass some arguments.

category?

title?

press?

year?
2005,2007
author?

price?
90
(bf2,Computer Science,Computer Architecture,xxx,2005,xxx,90.000000,2,2)
(bf3,Computer Science,Computer Architecture,xxx,2006,xxx,90.000000,2,2)
(bf4,Computer Science,Computer Architecture,xxx,2007,xxx,90.000000,2,2)

success!
```

五、 遇到的问题及解决方法

1. SQLBindParameter 函数出现段错误

问题：调试发现，当 SQLBindParameter 的参数是一个字符串时，出现段错误。

查阅资料发现，自己对最后一个传指针的参数理解有误。课程 PPT 中显示，传入 SQL_NTS(-3)即代表以空结尾的字符串。然而，该值应该赋给指针参数所指向的内存空间，而非赋给指针本身。

解决方法：最后一个参数设为 NULL，或者为一个 long long int 类型变量赋值为 SQL_NTS,然后将其指针传给 SQLBindParameter 函数的最后一个参数。

2.在预处理语句执行后，使用其他语句出现错误

查阅其他资料发现，在绑定参数后，要执行另外的语句，需先进行解绑。

解决方法：在执行完上一个 SQL 语句后，调用 SQLFreeStmt，具体为：

```
SQLFreeStmt( stmt, SQL_UNBIND);
```

3.无效的游标状态

当上一个语句的 SQLFetch 未遍历完全部结果时，执行其他 SQL 语句，会发生该错误。

解决方法：重新关闭再打开连接池；或者在上一个查询完成后，执行语句 SQLCloseCursor(stmt);

4.fgets 的使用问题

使用 fgets 函数替代 gets 函数以增强安全性时，会发现每一行读入的回车符仍会保留，导致字符串与数据库中的各个属性不匹配，造成错误。

解决方法：用 gets 替代 fgets

5.一些语句无法成功执行

造成语句执行失败的原因有很多，基本上是语法错误。其中，字符串为加单引号是最为普遍的错误。因此需要注意对每个非预处理语句的中的字符串假设单引号。

六、 总结

此次实验，是对数据库课程中前半部分实验内容的综合考察，除了抽象设计层面的 ER 图和 DDL 设计外，还既要求掌握基本的 SQL 语句语法，又要求掌握 SQL API 的使用方法，结合自身的编程能力，进行程序的编写，因而还考察了系统地编写程序的能力。在此次实验中，我遇到了很多关于编程的 bug 调试问题和 SQL API 的实验问题，同时也因为自身混乱的代码风格。但同时，此次实验也让我成长了许多，更加理解了系统化编程以方便调试、减少冗余和提高效率的重要性，为以后同小组合作实现 Mini SQL 打下坚实的基础。