

数据库Assignment3实验报告

彭润宇 18307130213

一、目录

1. 目录
2. 实验概述
3. 数据库设计
4. 后端接口
5. 使用指南
6. 测试方法
7. 参考资料

二、实验概述

本项目实现了一个较简单的图书管理系统。可供专门的管理员实现管理学生账户、管理图书、协助同学查询书籍借书、还书等基础功能。项目由Golang语言编写，并可通过专门的test文件对library.go进行测试。library.go本身的main函数实现了一个简单的命令行交互界面。

本项目的开发环境如下：

```
system: ubuntu 18.04.  
go: go version go1.14.2 linux/amd64  
mysql: Server version: mysqlserver 5.7.25
```

使用了如下package：

```
import (  
    "fmt"  
    "time"  
    "strings"  
    _ "github.com/go-sql-driver/mysql"  
    sqlx "github.com/jmoiron/sqlx"  
)
```

三、数据库设计

本项目中涉及到的现实泛关系模式较为繁琐，需要尽可能的进行分解来将问题简单化。

为了实现项目要求的功能（见后文），我们构造了10个模式来完美解决这些问题。

这些表有：

```
create table S(S int not null, primary key(S));
-- S(S)（学生账号，可扩展以加入密码）

create table T(T int not null, primary key(T));
-- T(T)（管理账号，可扩展以加入密码）

create table BT(B int not null, TITLE varchar(40) not null, primary key(B));
-- BT(B, TITLE)（书号+书名）

create table BA(B int not null, AUTHOR varchar(40) not null, primary key(B));
-- BA(B, AUTHOR)（书号+作者名）

create table BI(B int not null, ISBN char(13) not null, primary key(B));
-- BI(B, ISBN)（书号+ISBN码）

create table BB(B int not null, BR char(1) not null, primary key(B));
-- BB(B, BR)（书号+是否当前可借阅）

create table BE(B int not null, EXP varchar(40) not null, primary key(B));
-- BE(B, EXP)（丢失书籍书号+丢失理由）

create table SBC(
    S int not null,
    B int not null,
    C int not null,
    primary key(S,B),
    foreign key(S) references S(S),
    foreign key(B) references BT(B)
);
-- SBC(S, B, C)（学生号+书号+延期次数）

create table SBD(
    S int not null,
    B int not null,
    DT datetime not null,
    primary key(S,B),
    foreign key(S) references S(S),
    foreign key(B) references BT(B)
);
-- SBD(S, B, DT)（学生号+书号+截止日期）

create table SBR(
    S int not null,
    B int not null,
    RD datetime not null,
    primary key(S,B,RD),
    foreign key(S) references S(S),
    foreign key(B) references BT(B)
);
-- SBR(S, B, RT)（学生号+书号+已归还日期）

-- 以上表格的分解指导规则为4NF，冗余程度很低。
```

附：需要实现的功能有：

1. 管理员模式添加学生账户。
2. 管理员模式添加书籍。
3. 管理员模式删除书籍并说明原因。
4. 学生模式按照书名、作者或ISBN（13位）来查找书籍。
5. 学生模式借书。
6. 学生模式查询借书历史。
7. 学生模式查询已借未还书籍。
8. 学生模式查询某书归还期限。
9. 学生模式申请延期归还（最多三次）。
10. 学生模式查询自己是否有书未还。
11. 归还图书。
12. 若有三本以上逾期书籍，则暂时冻结某学生账户，待归还后解冻。

四、后端接口

后端使用go语言内嵌mysql实现。

```
func (lib *Library) ConnectDB() error//用Const里的用户名和密码连接对应数据库

func (lib *Library) CreateTables() error//如果未建立，则建立数据库模式

func (lib *Library) AddS()error//管理员账户添加新的学生账户

func (lib *Library) ReturnB(S,B int) error//学生账户S归还书号为B的书籍

func (lib *Library) Check(S int) error//查询学生账户S是否有逾期未还书籍

func (lib *Library) Check_sus(S int) bool//登录时判断学生账户S是否应该冻结

func (lib *Library) ConT(S,B int) error//学生账户S为自己借走的B延期

func (lib * Library)DueT(B int) error//查询B书的归还期限

func (lib *Library) BRS(S int) error//查询学生账户S已借未还书籍

func (lib *Library) History(S int)error//查询学生账户S所有借阅历史

func (lib *Library) Borrow(S,B int)error//学生账户S借走书B

func (lib *Library) AddBook(title, author, ISBN string)error//添加书籍

func (lib *Library) RemoveBook(B int,Exp string)error//以Exp为借口删除书籍
```

```
func (lib *Library) FindT(title string)error//通过书名title查找书号B

func (lib *Library) FindA(author string)error//通过作者author查找书号B

func (lib *Library) FindI(ISBN string) error//通过ISBN码查找书号B
```

五、使用指南

1. 安装

[项目链接](#)

将项目clone到本机，修改Libar/go/文件夹下library.go中常数部分为本机mysql服务器的用户名、密码和想要使用的数据库名称并确保用户密码正确且数据库存在。

```
const (
    User      = "root"
    Password  = "123456"
    DBName    = "ass3"
)
```

为保证数据库的安全，用户需要在预运行一次程序后登录mysql，在对应数据库下的T模式中添加管理员账户，示例如下：

```
INSERT INTO T VALUES(any-int-you-want-to-be-the-account);
```

2. 使用方法

在文件夹Libar/go/中打开终端并输入下列指令。

```
go run library.go
```

进入下列界面：

```
pryest@pryest-Surface-Pro:~/Libar/go$ go run library.go
Welcome to the Library Management System!
Login as a student,please input 1, login as a teacher,please input 2
```

输入1进入学生界面，如果学生账户不存在则会在提示后返回此界面，如果学生账户存在且被挂起则只剩下还书功能，否则在输入学生账号后出现以下完整界面：

```
Login as a student,please input 1, login as a teacher,please input 2.
1
Please input your student id.
1
Input 1 to return book.
Input 2 to find books by title.
Input 3 to find books by author.
Input 4 to find books by ISBN.
Input 5 to borrow one book by the book id.
Input 6 to query the histroy.
Input 7 to query your books that are not returned.
Input 8 to query the duetime for one book by bookid.
Input 9 to push the duetime for one book by bookid.
Input 10 to check if there is overdue book(s) you need to return.
Input anyother number to exist.
```

在初始界面输入2并输入管理员id进入管理员界面：

```
Welcome to the Library Management System!
Login as a student,please input 1, login as a teacher,please input 2
2
Please input your teacher id.
110
Input 1 to add book.
Input 2 to remove book.
Input 3 to add student account.
Input anyother number to exist.
```

鉴于每一个界面都有清晰的指令集提示，在此不做赘述。

六、测试方法

可以通过下列指令测试是否成功安装：

在文件夹Libar/go/中打开终端并输入下列指令。

```
go test
```

成功安装则显示如下界面：

```
pryest@pryest-Surface-Pro:~/Libar/go$ go test
New student id: 1
Book number: 2.
Results: 1
Book number: 1.
Book number: 2.
Results: 2
Book number: 1.
Book number: 2.
Results: 2
Duetime: 2020-05-23 11:34:21.
Duetime pushed to 2020-5-23 11:34:21
No overdue book need be returned.
PASS
ok      _/home/pryest/Libar/go  0.117s
```

七、参考资料

《数据库系统教程》第3版 施伯乐, 丁宝康, 汪卫 编著

[A Tour of Go](#)