

作业 3-图书管理系统 报告

技术科学试验班

孙若诗

19307130296

设计目的

1. 针对复旦大学图书馆基本需求，设计图书管理系统；
2. 体验数据库模式构建、功能设计过程；
3. 进一步掌握mysql和Go语言，提高调试能力；
4. 学习 test 的编写和使用方法。

版本信息

go1.41.1, mysql80, Microsoft Windows10.0.17763.1158

仓库地址

<https://github.com/moyiii-ai/ass3-library>

关系模式

user(*user_id*, *password*, *is_administrator*),

book(*book_id*, *title*, *author*, *isbn*, *is_available*, *explanation*),

record(*record_id*, *user_id*, *book_id*, *borrow_data*, *return_data*, *deadline*, *delay_time*)。

1. *user*存储管理员和普通用户账号信息：*user_id*为主键，VARCHAR 类型，表示用户名；*password*为 VARCHAR 类型，表示密码；*is_administrator*为 INT 类型，表示是否管理员。
2. *book*存储书籍信息：*book_id*为主键，系统自动生成递增的 INT 类型，表示书本编号；*title*为 VARCHAR 类型，表示书名；*author*为 VARCHAR 类型，表示作者；*isbn*为 VARCHAR 类型，表示书籍 ISBN 码；*is_available*为 INT 类型，表示书籍状态（1 为可出借，2 为已借出，3 为已删除）；*explanation*为 VARCHAR 类型，用于删除时记录原因。
3. *record*存储借书记录：*record_id*为主键，系统自动生成递增的 INT 类型，表示记录编号；*user_id*为外键，指向*user*表中*user_id*，VARCAHR 类型，表示借书用户名；*book_id*为外键，指向*book*表中*book_id*，INT 类型，表示出借书籍编号；*borrow_data*为 VARCHAR 类型，表示出借日期；*return_data*为 VARCHAR 类型，表示归还日期，归还前为""；*deadline*为可出借时长，INT 类型，初始为 90 天；*delay_time*为申请延期次数，INT 类型。

功能概要

管理员账号

1. 注册管理员账号

要求给出此数据库密码进行验证，随后设置管理员账号名及密码。

2. 增加图书

输入图书`title`、`author`、`isbn`，将图书加入数据库。

3. 删除图书，并解释原因

输入要删除图书编号、给出解释，删除图书仍在数据库中，但状态变为已删除。若此书正在出借状态，本条借阅记录视为在该日归还。

4. 增加普通用户账号

以管理员身份登录时，可以输入普通账号用户名进行创建。用户名不能与已存在用户重复，初始密码为 233。

用户账号

5. 借书

以普通用户身份登录时，可以输入书籍编号借阅；不可借阅已被借出书籍或已删除书籍；每本书初始归还期限 90 天。三本书逾期未还，或借阅册数达到三十本时禁止借书。

6. 还书

以普通用户身份登录时，可以输入借阅记录编号归还书籍。只能归还自己所借书籍，且本书不能被标记为已删除或已归还。

7. 申请延期还书；

每次借阅最多可以申请三次延期归还，每次延期 30 天。参照我校规定，延期不能在图书已到归还日期后申请。

共用功能

8. 修改本账号密码

登录状态下可修改本账号密码。

9. 查询图书

已知书名、作者或 ISBN，可给出三项信息中的 1~3 项来查询书单，对不清楚信息用“no”占位即可。

10. 查询借书历史

输入用户名，查询该用户所有已归还和未归还的借书历史；管理员可以查询任何用户，普通用户只能查询本账号记录。

11. 查询出借未还清单

输入用户名，查询该用户目前尚未归还的所有借书记录；管理员可以查询任何用户，普通用户只能查询本账号记录。

12. 查询还书截止日期

输入借阅记录编号，查询该借阅记录的还书截止日期。管理员可以查询任何用户，普通账号只能查询属于自己的借阅记录。

13. 检查是否有逾期未还的书

输入用户名，查询该用户是否有逾期未还的书；管理员可以查询任何用户，普通用户只能查询自己的记录。

使用指南

1. 首先应在`library.go`开始部分填充用户名和密码，使程序可以连接`mysql`。
2. `library_test.go`可测试各函数功能，但需要先使用`library.go`创建初始管理员（以`moyi`为例）和一个普通用户（以`zzh`为例）账号。

```
--- PASS: TestQueryBorrow (0.04s)
=== RUN TestQueryDeadline
Error 2: You cannot query deadline because the record do not belong to you.
Deadline : 2020-11-4
Error 2: You cannot query deadline because the record do not belong to you.
Error 12: You cannot query deadline because the book has been deleted or returned.
--- PASS: TestQueryDeadline (0.02s)
=== RUN TestCheckOverdue
No book overdue.
No book overdue.
No book overdue.
Error 2: You can only check your own information.
--- PASS: TestCheckOverdue (0.02s)
=== RUN TestReturnBook
Error 2: You cannot return book because you are administrator
You have successfully returned the book 2.
Error 12: You cannot return this book because it has been deleted or returned.
Error 2: You cannot return this book because the record do not belong to you.
--- PASS: TestReturnBook (0.03s)
PASS
ok      github.com/ichn-hu/IDBS-Spring20-Fudan/assignments/ass3/boilerplate 1.544s
```

3. 运行`library.go`，出现欢迎语句。按指示，输入`enroll`来注册管理员账号、输入`login`登录或输入`exit`离开。

```
Welcome to the Library Management System!
Please Input 'login' to sign in, 'enroll' to sign up an admin account or 'exit' to exit.
>enroll
Please enter the Password of database to prove you have the authority, or input 'quit' to leave.
> moyi
Error 4: Wrong Password!
Please enter the Password of database to prove you have the authority, or input 'quit' to leave.
> 123
Welcome to enroll.
Please input your user_id, or input 'quit' to leave.
> moyi
Please input your password
> 123
Please repeat you password
> 123
Administrator account 'moyi' has been created successfully!
```

4. 登录后可以输入：`addbook`增加书籍、`deletebook`删除书籍、`adduser`创建普通账号、`borrowbook`借书、`returnbook`还书、`searchbook`搜索书籍、`queryhistory`查询历史、`queryborrow`查询未还书籍、`querydeadline`查询归还截止日期、`checkoverdue`查询逾

期借阅、*askfordelay*申请延期、*changepassword*修改密码、*logout*登出。选择各功能后，可按提示继续操作。

5. 错误编号

- 1) Error 1: 无意义的输入
- 2) Error 2: 此账号没有这项操作权限
- 3) Error 3: 用户不存在（登录或查询时）
- 4) Error 4: 密码错误
- 5) Error 5: 用户已存在（申请新账号时）
- 6) Error 6: 此书籍已被删除
- 7) Error 7: 两次密码不匹配（注册新账号或修改密码时）
- 8) Error 8: 因逾期未还书籍达到三本，不允许借书
- 9) Error 9: 因出借书籍过多，不允许借书
- 10) Error 10: 此书籍已被借出
- 11) Error 11: 此书不存在
- 12) Error 12: 此借阅记录已被归还
- 13) Error 13: 此借阅记录不存在
- 14) Error 14: 已申请延期三次，禁止再申请
- 15) Error 15: 申请的借阅记录已到期，禁止申请延期

设计讨论

1. 管理员账户无借还书权限，管理员权限并不包含用户所有权限：管理员需要使用个人的普通账号进行借还书。系统的管理员账号和普通用户账号应当分开。
2. *user_id*为 VARCHAR 类型：初始设计*user_id*类似*book_id*，为一个自动增长的 INT 类型，而另外设置 VARCHAR 作为用户名。考虑到用户需要登录才能进行后续操作，不易像查找书籍编号、记录编号一样查询用户编号，应该允许登录方式为一个和用户个人有关的、可以通过管理员账户设定的用户名而非自动值。不作为主键的昵称固然可以优化用户体验，但是只在少数的几处登录、登出问候上使用，出于效率考虑取消了这个属性。
3. *explanation*、*title*、*author*均为 VARCHAR 类型：但显然实际使用中存在书名、解释中有空格、作者不止一人等情况，需要优化设计。
4. 借还日期使用 VARCHAR 类型：没有采用 DATE 类型，是因为 int 和 string 的转换模拟更加方便。初期版本使用了 Go 的时间戳、时间转换等函数，困扰于 string、int、int64 转换的繁琐和 Go 语言强类型的限制，最终还是改为手动模拟。
5. *book*表的主键仅为*book_id*：图书馆同一版本的书往往不止一本，其他属性都不能保证独一无二。

6. *record_id*为*record*表主键：一个用户可能多次出借和归还同一本书，因此用户名和书籍编号显然无法保证独一无二。既然*record_id*已经可以确定所有信息，再加入用户名和书籍编号就成为了超键，不利于系统效率。
7. *book_id*和*record_id*：以这两项 INT 作为各种查找的依据，对编程者较为友好，但用户可能难以记住现实意义不明确的编号，往往需要先查找书目或借阅记录，才能确定编号。
8. *deadline*与*delay_time*：这两项属性其实存在冗余，*deadline*是不必要的。由于初始借期和每次延期的时间都一定，很容易利用*delay_time*计算得出。代码几近完成才意识到这个问题，修改成本已经过大，因此未进行优化。有趣的是，在设计关系模式时，*deadline*是更直观、更早确定的属性，这启发我在为关系模式新加入每个属性时都要检查和之前属性是否存在冗余，在开始实践之前谨慎敲定最优设计。
9. 安全性：虽然要求管理员验证数据库密码，但管理员密码本身已经在同一份代码中写出。若运行此程序来进行应用，安全保证显然形同虚设。但是囿于我的计算机知识还十分浅薄，只是意识到了这个问题，而不了解如何解决。
10. 关于错误编号：基本是按编程过程中遇到错误的顺序排列，但写完后发现编号缺乏逻辑性，若系统规模更大则容易混乱。更优策略是采取多位编码而非简单的上升数字、用第一位表示此错误相关的数据类型或操作类型（如书籍相关、用户相关、记录相关等），之后再给各个错误编号，用户体验要更好，也便于开发者维护和查找错误。
11. 系统在设计过程中对效率的考虑较少，只是直观的根据基本功能要求编码，应有较大的优化空间。
12. *library_test.go*已经尽可能地测试了写出的每一种合法或错误情况，但仍然是参考代码本身给出的验证，隐藏的错误往往是编程者尚未意识到的情况，需要在长期使用中加以监测。另外，由于主程序中借书时间默认为当前日期，为测试逾期情况强行加入了已过期的借书记录，时间为给定的常量（“2018-7-21”），这意味着若在不同时间运行测试程序，需要考虑时间常量是否有效（事实上，若测试时间晚于 2019 年 2 月，应该不会产生影响）。

多余的话

1. 写完才发现自己几乎把所有的“*date*”都拼成了“*data*”，甚至两个词、大小写都有出现，不宜一键替换。注释中也大多为 Chinglishn，为自己的工地英语感到深深愧疚。
2. 第一次进行类似的开发，前期十分迷茫，完全不知道如何入手。后来了解到“原来命令行交互就是字符串输入输出”，豁然开朗。纵使五一假期 camp、大作业、期中复习三线作战焦头烂额，在编程调试的整个过程中仍旧非常快乐乃至兴奋。是目前学生生涯做的最有成就感的作业，尽管它的技术难度对学长学姐们来说不值一提，但是敝帚自珍，这份作业对我是有开创性的意义的。

3. 同理，这也是我的第一份大作业报告，似乎深受基础物理实验报告的影响。抱歉仍旧使用 Windows 系统，实在难以接受 Linux 虚拟机的操作体验。
4. 彻底熟悉了 Go 的语言规范和 mysql 相关功能，了解了 Go 的“强类型语言”等特性。
5. 在已经成功试用过命令行功能之后才开始写 test，本来以为意义不大，却仍旧发现了很多问题。体验到了 test 的便捷和强大，也逐渐喜欢上了 Go 语言。
6. 每次做完数据库实践作业都会感到不可思议。赞美助教学长。