IDBS Assignment 3 Report

准备

在图书库管理系统，我们需要维护图书表，历史表，借阅图书-账号是多对一的关系，所以没必要创建一个表。但是需要一个表来记录历史。

本系统假定任何非管理员能够访问数据库的方法便是通过由 go 编写的终端程序（或者一个调用各个 go 函数接口的一个前端）。因为既然学生能够使续期次数增加，那么学生所使用的账号就能够执行使续期次数减少的 SQL 语句，为了使得学生不能这么做，我们必须要让学生只能访问到我们提供的接口，做到这一点的方法便是本 go 程序作为前端+只允许 localhost（如果有多台设备那么便是内网）的账户登录。如果你没有账号，根本访问不了数据库。

在系统设计中我还考虑了以下点：

还书

return a book to the library by a student account (make sure the student has borrowed the book)

恕我直言，没去过 Fairy Union of Defense and Attack Nebula University 的图书馆。但是我去的上海其他图书馆进行还书都是不需要某个确切账号的。

为了实现这个奇怪的要求，我写了一个“Checkifbookborrower”的 helper function，如果前端的设计者希望达成这个限制的话能够调用。

冗余信息

是不是应该在 db 里加一点冗余信息来加速查询?虽然很想加，但是我没有加。因为我的确不知道加什么冗余信息能加速查询，能加速多少也是我猜的，所以还是不加为好。

限制

因为不知道是怎么锁定书库中唯一的一本书的，所以我 假定 FUDANU 同样的书（相同 ISBN）最多只有一本。

图书删除、可用

续期是借阅的别名。不管借阅还是续期都是需要登录的。

如果图书借阅已经逾期，不允许续借。

可用与续借是同一个变量。续借为 0 代表所有人都可以借阅，否则<4 则代表可以续借。

图书被删除是将可用置 4，如果哪天同一本书找回来了还能直接修改。

当前持有书籍（Heldbooks）

query the books a student has borrowed and not returned yet

Heldbooks 是 QueryBooks(“holder”, User)的别名

SQL Injection

这点其实不需要编写者来考虑.用户的输入要作为 parameter 提供给数据库，而不是给 fmt.Sprintf (sql - The Go Programming Language)

Query

三种 query 用的是同一个接口 QueryBook。

Library.go

因为提供的这些函数都是接口，所以非致命的 error 都是返回而不会处理的。

测试

测试借书还书中，为了减小对其他接口的依赖（方便）直接修改了 user 变量来模拟一个其他人登录了系统。实际上应该是管理员创建两个账户，logout，一个人登录借书，另一个人登录借不到。

感想

本次 assignment 有较多杂事需要完成，除了花费了大量时间，让我感觉像是在小学学编程外，增长了一些使用 go 的经验。既要写一个 console 的交互又要写一个 test 感觉工作很重复。总的来说，效率比较低。

到最后也没能搞懂 sqlx.StructScan 怎么出错了，比较可惜。没能通过这样一个 none standard 的库享受到好处。

References

Google. (n.d.). *sql - The Go Programming Language*. Retrieved from Packages - The Go Programming Language: https://golang.org/pkg/database/sql/#DB.Query