

# 实验报告 18307130104

---

## 工程地址

---

[工程地址](#)

<https://github.com/unicornt/IDBS-Library>

## 数据表组成

---

总共创建了5张表。

Book(id, title, author, ISBN) 之所以添加一个id是考虑到所有信息相同的书可能有很多本在图书馆中，因此选择一个id来作为书的唯一标识符。

student(id, username, password) 为所有学生添加一个id也是为了标识方便，username和password是为了登陆系统而设计的，不过实际上目前没有用到。

Record(book\_id, student\_id, ddl, ext\_times) 记录借书且没有还的记录。

Drecord(book\_id, student\_id, return\_time) 记录已经还书的记录。

Removedbook(book\_id, title, author, ISBN, removedreason) 记录删除的数以及删除的原因。

## 函数功能以及实现

---

### AddBook title, author, ISBN

实现过程：向Book中插入一条新的记录。

### RemoveBook book\_id, reason

考虑到所有信息相同的书可能有很多本，所以函数要求输入一个book\_id来进行删除。同时需要检测该书是否存在于Book。

实现过程：将这条记录从Book中删去并加入到Removebook中，并且加上removedreason(=reason)这一属性。

### AddStudent username, password

插入新纪录之前要确保不存在用户名相同的学生，因为需要用户名用作登录系统的凭证。

实现过程：向Student中插入一条新的记录。

### QueryBook input, mode

考虑到实际应用中的不同需求，我实现了两种模式。一种是书名、作者、ISBN任意一个匹配输入字符串都会出现在结果中，另一种是指定其中一个来匹配输入字符串。这里的匹配要求是完全匹配。

实现过程中由于返回值个数的不确定，返回的是一个结构体(Book)的数组，为了统一之后的函数，结构体Book中包含了ddl和ext\_time两个无用参数，赋值为niltime(定义在开头，取值time.Time类型的默认值)以及-1。

## BorrowBook student\_id, book\_id, borrow\_time

给定学生的id和书本的id，以及当前的时间，来完成借书信息的写入。之所以要求提供当前信息，主要是为了测试的方便，实际应用的过程中完全可以取当前时间。未申请延期的书本还书时间设定为结束后28天，因为我查到学校图书馆还书时间为60天，但是有人预约这本书之后还书时间会被提前，由于我的系统没有预约功能，因此做了一个折中（我觉得能延期3次有点多）。

实现过程中首先检验书本是否存在以及是否被已经被借走。还需要确认学生的账户当前逾期未还的书本数是否小于等于3。全部通过之后，向Record中插入一条新的记录。

## QueryHistory student\_id

考虑到已经还过的书和还未还的书显示有所区别，因此我在设计的过程中特别分成两张表。这样可以着重显示需要还的书。

实现过程：函数返回两个Book数组，一个是未还借书记录以及书本信息，一个是已还借书记录以及书本信息。

## QueryNotReturn student\_id

只需要在Record和Book链接的表中查询student\_id等于给定id的记录即可，通过Book得到书本信息。

实现过程：返回一个Book类型的数组。

## QueryDeadline book\_id

需要检查是否存在这条借书记录。

实现过程：从Record查询book\_id记录并得到ddl。

## ExtendTime book\_id

设计成一次延期归还可以延期7天。由于只能延期3次，所以修改记录之前需要检查已经延期的次数。

实现过程：更新Record中的记录，ddl加7天，ext\_times加1。

## QueryOverdue student\_id, today

这里需要提供today是为了测试方便，实际上today应该就是系统当前的时间。

实现过程：函数返回一个Book数组。从Record与Book链接的表中查询ddl > today且是student\_id的记录。

## ReturnBook student\_id, Book\_id

需要先判定这条借书记录是否存在，然后再进行插入和删除的操作。

实现过程： 向Drecord中插入新记录并包含还书时间。从Record中删除这条记录。

## library\_test说明

---

将library.go和library\_test.go放在同一目录下之后，在终端输入go test -v，即可进行。

我使用的环境是macOS10.15，mysql8.0.19， go1.14.1。

除了测试每个函数是否能运行之外，也测试了几个要点：

1. 相同书多次加入。
2. 不同模式下的查询功能。
3. 逾期未还书超过三本能否借书
4. 能否延期还书超过三次。