

bookstore项目汇报

汇报时间：2021年1月13日

汇报人：张硕闻、陈煜、杜涵悦

Agenda

目录页



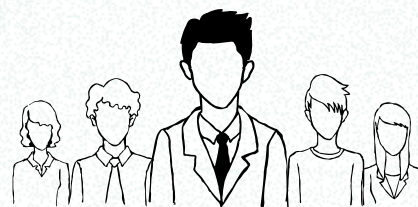
1

数据库设计



2

功能实现



3

测试及性能分析



4

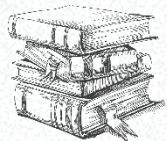
分工及版本控制



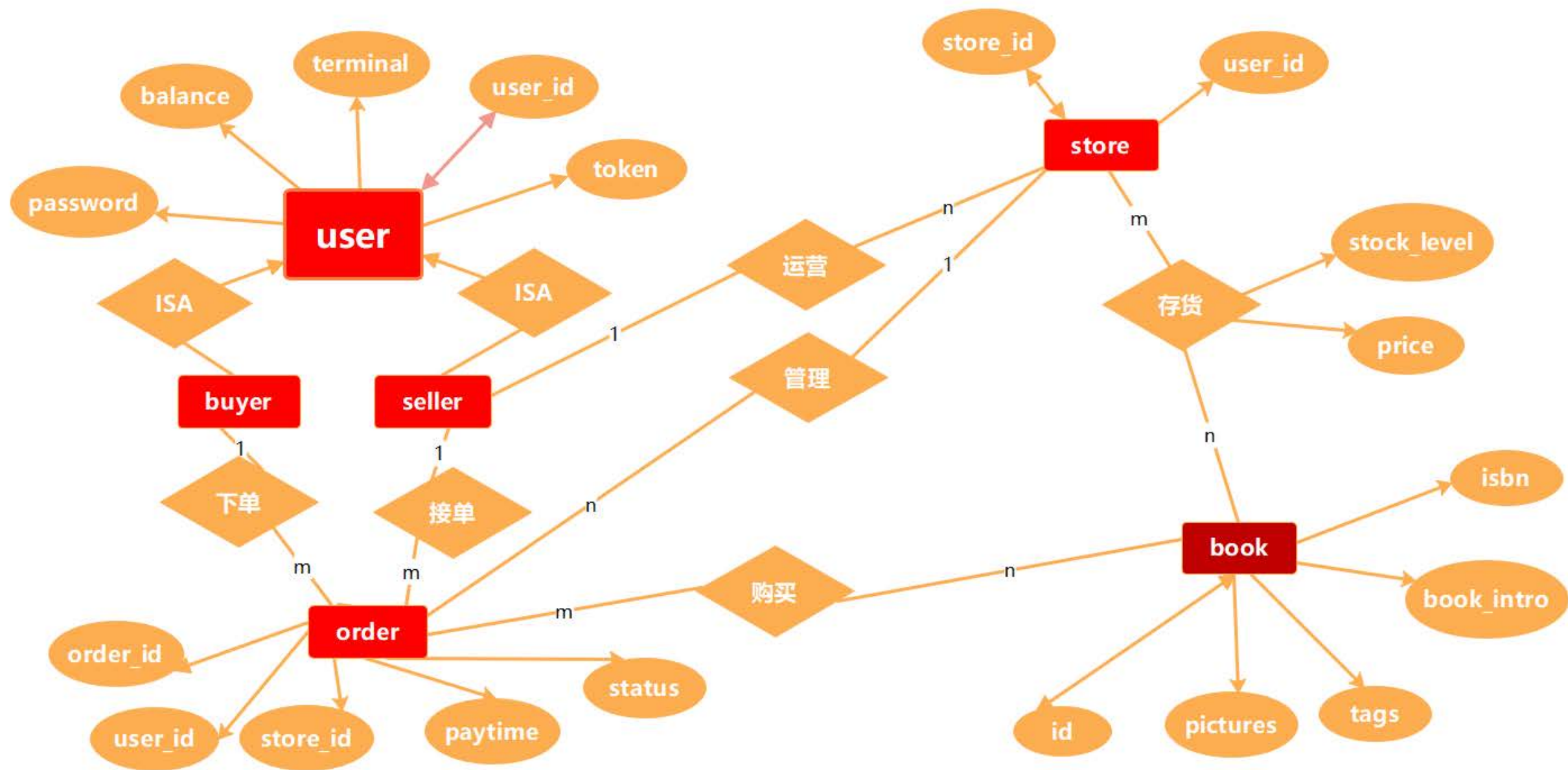
数据库设计

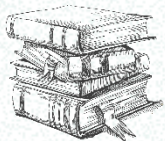
⊕ ER 图

⊕ 关系模式

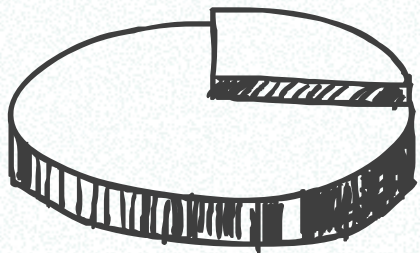


ER 图



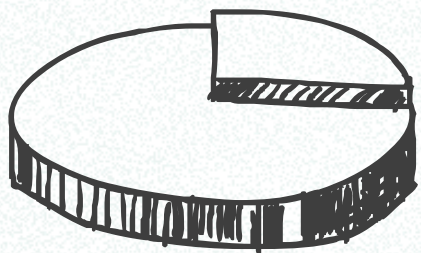


关系模式



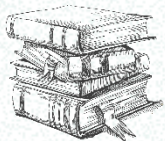
用户表(user)

user_id	password	balance	token	terminal
String(primary_key)	String	Integer	String	String
user_id是主键，通过user_id进行查找，例如登录时查看密码是否正确，用户充值时查看还剩多少余额。设计满足第三范式。				

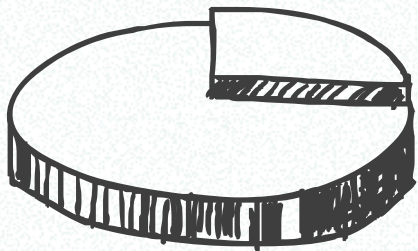


用户和商铺关系表(store)

store_id	user_id
String(primary_key)	String(Foreign_key)
store_id是主键，user_id作为外键，加快商家查看店铺操作的速度。满足第三范式。	

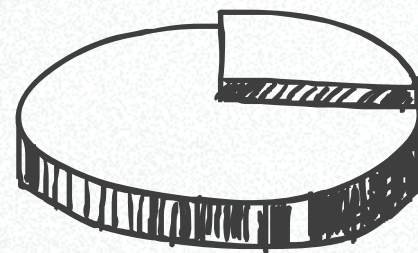


关系模式



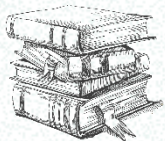
商铺详情表(store_detail)

store_id	book_id	stock_level	price
String(primary_key,ForeignKey)	String(primary_key,ForeignKey)	Integer	Integer
store_id和book_id组成联合主键，对store_id和book_id都作为外键，加快搜索时根据书找到对应的店铺和根据店铺找到店铺内的书。满足第三范式。			

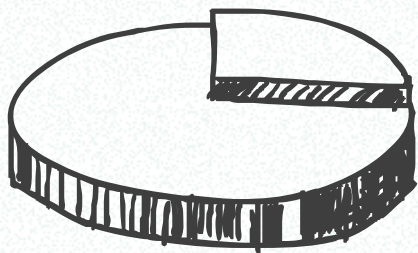


书籍表(book)

book_id	title	author	...
String(primary_key)	String	String	
book_id是主键。满足第三范式。			

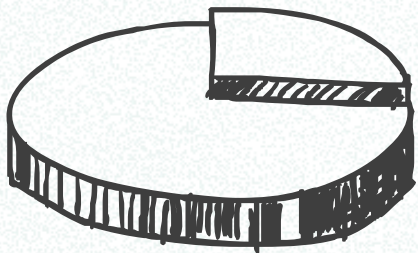


关系模式



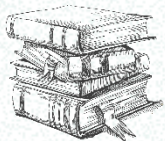
订单表(order)

order_id	user_id	store_id	paytime	status
String(primary_key)	String(ForeignKey)	String(ForeignKey)	DateTime	Integer
order_id是主键，user_id和store_id作为外键。满足第三范式。				

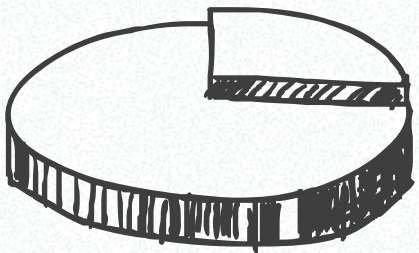


订单详情表(order_detail)

order_id	store_id	count
String(primary_key)	String(primary_key,ForeignKey)	Integer
order_id和book_id组成联合主键，由于订单和书本是多对多关系，因此我们需要建立一张额外的订单详情表。满足第三范式。		



关系模式



待付款订单表 (order_to_pay)

order_id	user_id	store_id	paytime
String(primary_key)	String(ForeignKey)	String(ForeignKey)	DateTime
order_id是主键，鉴于我们引入自动取消超时订单功能，设定每秒查询一次未付款订单，为降低检索复杂度增强检索效率，故分离出单独存储未付款订单的表，不再添加额外索引。满足第三范式。			



功能实现

⊕ 用户功能

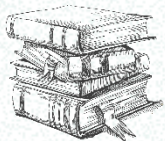
⊕ 买家功能

⊕ 卖家功能

⊕ 搜索功能

⊕ 收发货功能

⊕ 查询和取消订单功能



用户功能

1. 注册

1. 根据user_id判断该用户名是否已经存在。
2. 插入user_id、password、balance、token、terminal信息至user表。

2. 登录

1. 根据user_id获取用户密码。
2. 与用户输入密码对比。
3. 更新token, terminal。

3. 登出

1. 根据user_id查询该user是否处于登陆状态。
2. 更新token。

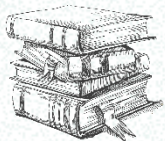
4. 注销

1. 根据user_id查询该user是否存在。
2. 删除对应user表中条目。

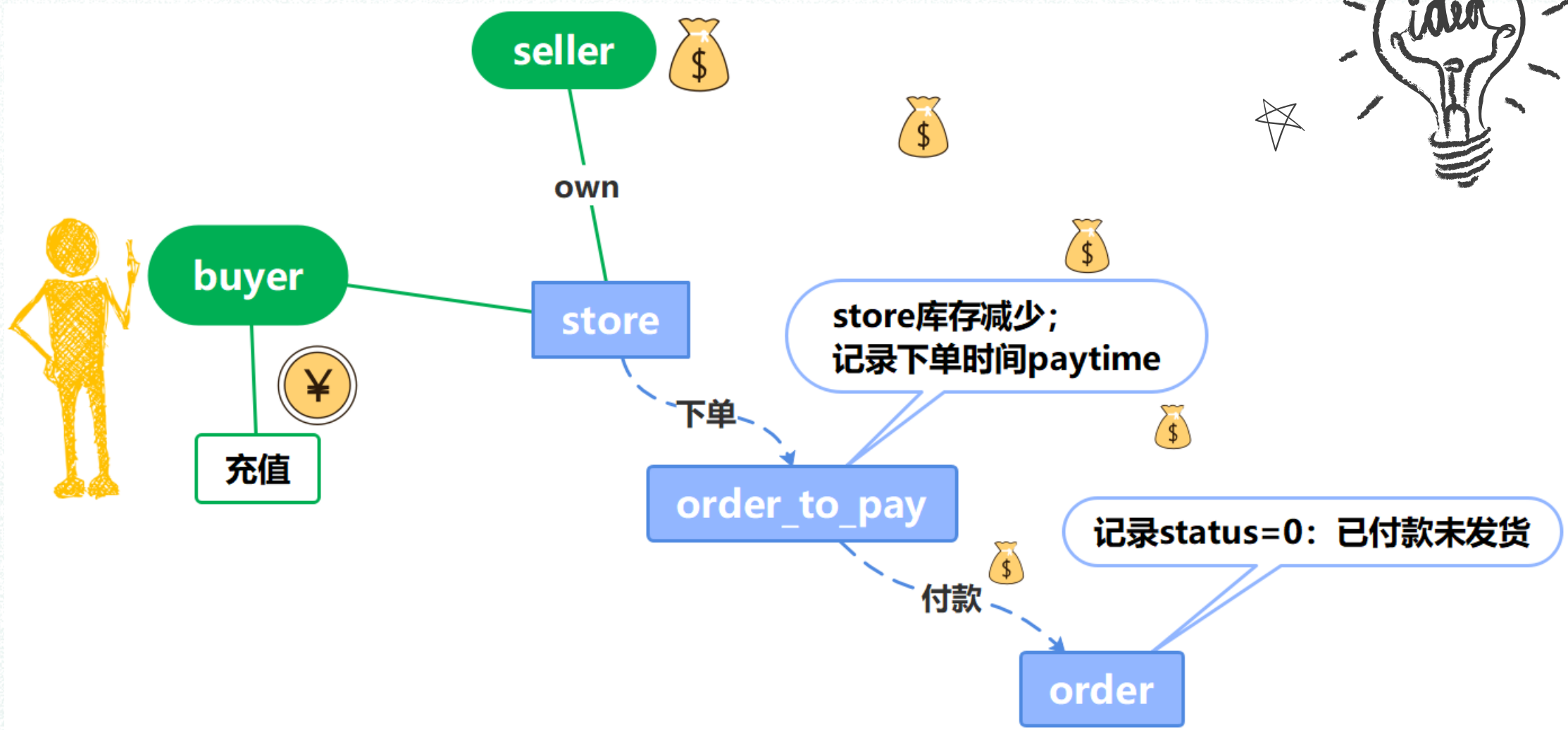
5. 更改密码

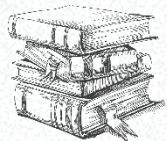
1. 根据user_id获取用户原有密码。
2. 与用户输入的旧密码对比。
3. 若相同，更新用户密码。





买家功能





卖家功能

1. 创建店铺

1. 检查user_id和store_id是否已存在。
2. 插入用户id, 新建店铺store_id至user_store表。

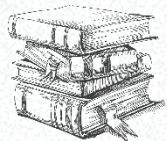


2. 上架图书

1. 检查user_id和store_id是否已存在。
2. 根据book_id从book表查询是否存在对应book。
3. 若不存在, 首先将书本信息插入book表。
4. 将store_id, book_id, 出售价格插入store表。

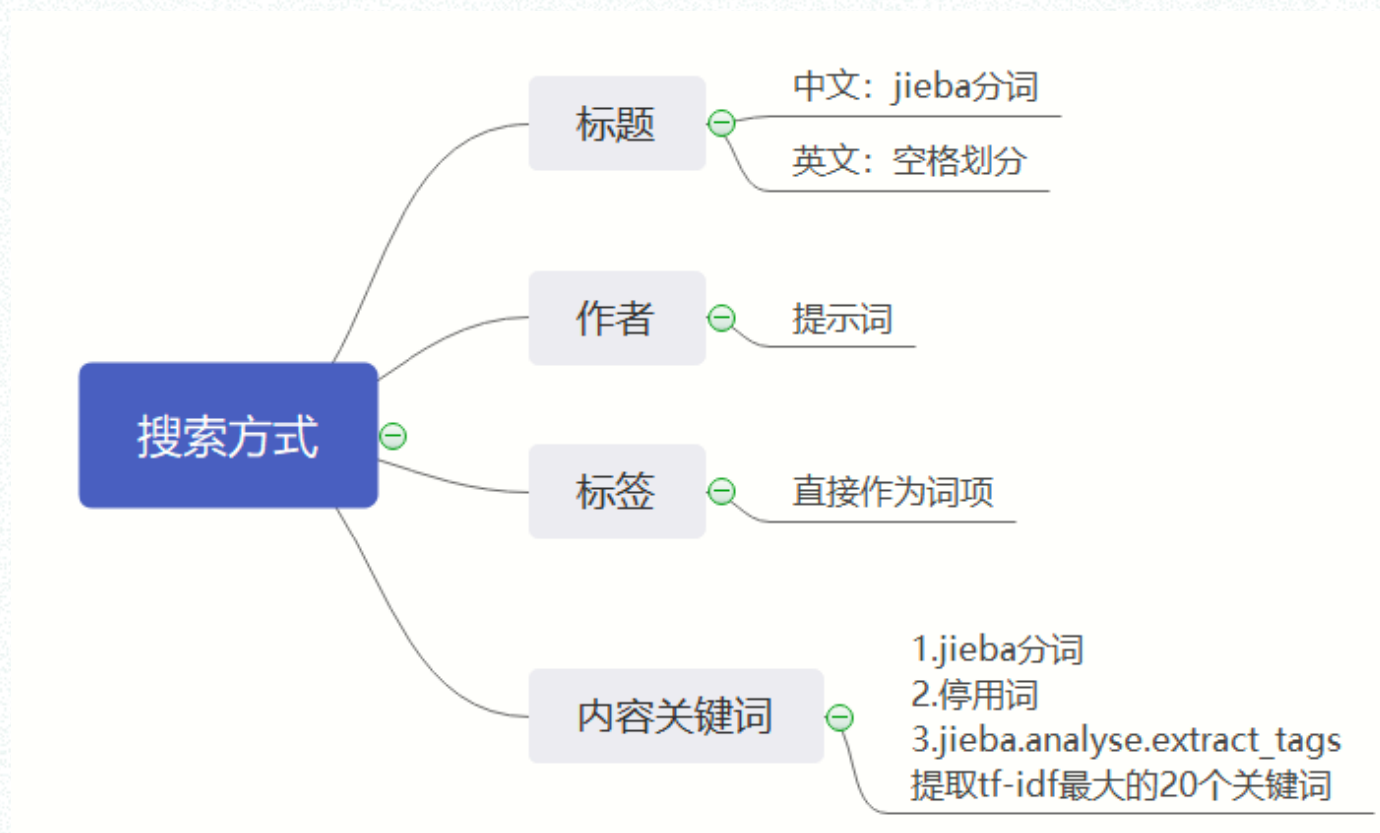
3. 添加库存

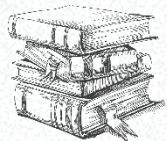
1. 检查user_id、store_id和book_id是否已存在。
2. 根据store_id, book_id寻找对应店家书本库存, 并在store表中更新库存。



图书搜索功能--倒排索引

搜索方式



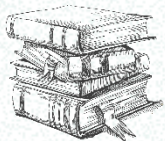


图书搜索功能--倒排索引

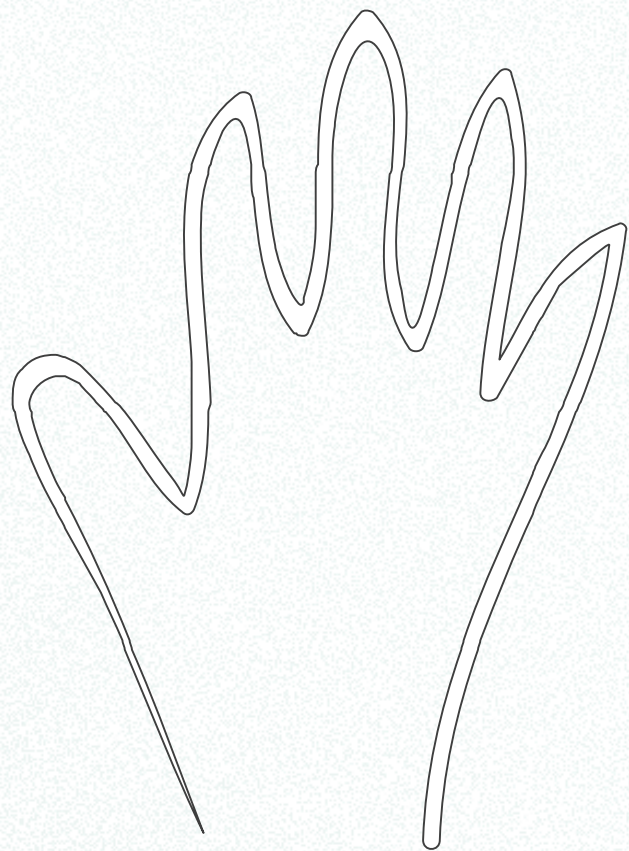
搜索范围

全局搜索：参数传入搜索方式（标题、作者或关键词），在对应的倒排索引表中查找book_id，再在book表中根据book_id找到这本书的相关信息。

店铺搜索：参数传入搜索方式（标题、作者或关键词）和store_id，在对应的倒排索引表中查找book_id，并且在店铺库存详情表中根据store_id找出所有的book_id，取book_id的交集再在book的表中根据book_id找到这本书的相关信息。



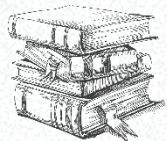
图书搜索功能--倒排索引



分页显示结果

由于某些查询词对应了多本书，查询返回结果可能太大，故采用分页显示查询结果的方式。每次查询时，传入一个参数 page（大于等于1，一般取1），实现每次返回10行结果作为一页。

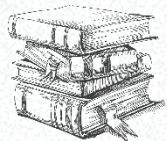
```
records = session.execute(  
    "SELECT title,author,publisher,book_intro,tags "  
    "FROM book WHERE book_id in "  
    "(select book_id from search_author where author='%s') LIMIT 10 OFFSET %d" % (  
        author, 10*page - 10)).fetchall()
```

收发货功能



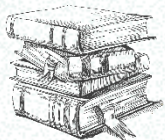
1. 检查order_id是否存在。
2. 查询order的状态是否为已付款。
3. 看store的店铺主是不是seller。
4. 将order的状态置为1。



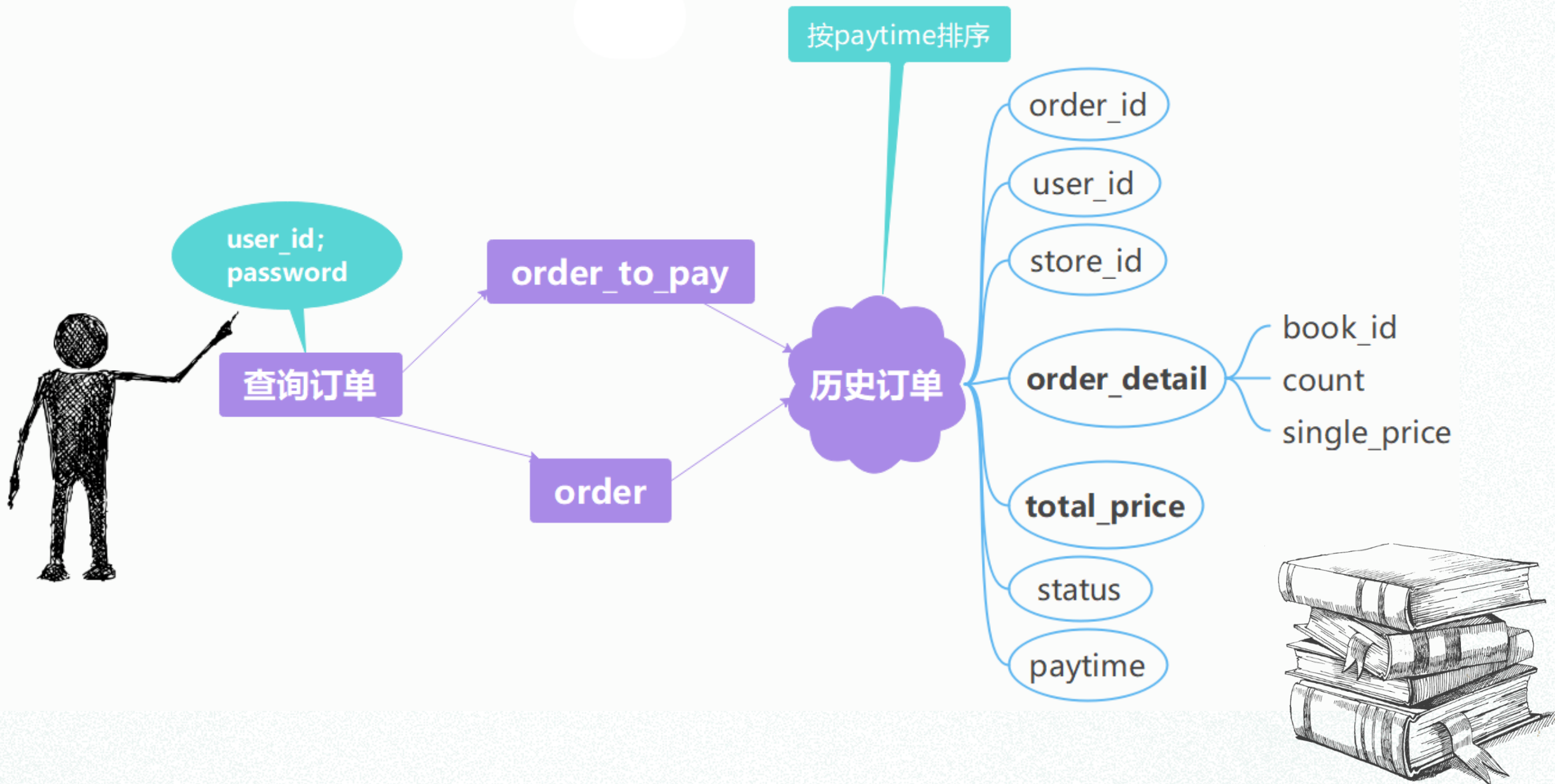
收发货功能

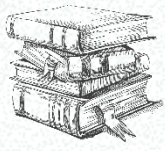


1. 检查order_id是否存在于order表中。
2. 检查order的status是否为0或2。
3. 查看user_id与buyer_id是否一致。
4. 若以上都满足，置order的status为2。



查询历史订单功能



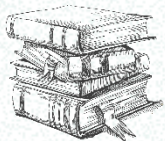


买家取消订单功能

只能取消**下单未付款**和**付款未发货**的订单

- 根据 order_id 和 user_id 在order_to_pay及order表中查找订单信息。
- 若订单记录存在于order_to_pay表中，只需获取store_id后为卖家加回库存，再删掉表中记录即可。最后，在order表加上相应记录，设置status为4，表示订单已关闭。
- 若订单记录存在于order表中，检测该记录status是否为0。之后，获取对应卖家user_id和store_id，加回库存的同时一同完成资金转移。最后，更改order表中记录，设status为4。





自动取消订单功能



- 调用**Timer**类实现自动取消订单。Timer是Thread的子类，通过Timer(t, fun)函数设置每t秒调用一次fun函数。并且，我们通过start()和cancel()函数设置线程开始或取消执行。
- 每次调用函数，将查询order_to_pay表。并获取所有存在记录中的paytime，与当前时间做差，若时间差大于15min，则自动取消该订单。
- 对每一个超时应关闭的订单，对其对应store_id的卖家商铺加回库存。
- 最后，将该订单的对应记录加入order表，并设置status为4，之后删除order_to_pay表中的记录。

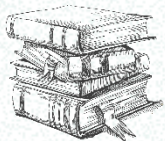


测试及性能分析

⊕ pytest测试

⊕ 覆盖率

⊕ 吞吐量测试



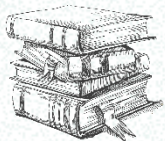
pytest测试

57个测试用例passed, 覆盖率96%

```
374 fe/test/test_send_books.py::Test_send_books::test_can_not_send PASSED [ 96%]
375 fe/test/test_send_books.py::Test_send_books::test_none_order PASSED [ 98%]
376 fe/test/test_send_books.py::Test_send_books::test_no_seller PASSED [100%]
377
378 ===== 57 passed in 353.79s (0:05:53) =====
379 2021-01-12 21:39:44,035 [Thread-4375 ] [INFO ] 127.0.0.1 - - [12/Jan/2021 21:39:44] "GET /shutdown
380 frontend end test
381 The command "coverage run --timid --branch --source fe,be --concurrency=thread -m pytest -v --ignore
382
```

test_new_order.py	39	0	100%
test_password.py	32	0	100%
test_payment.py	59	1	98%
test_receive_books.py	57	0	100%
test_register.py	31	0	100%
test_search.py	78	0	100%
test_search_order.py	59	0	100%
test_send_books.py	59	0	100%
D:\Python\junior1\Database System\bookstore\init_database.py	91	0	100%

TOTAL	1958	74	96%



订单吞吐量和延迟

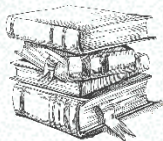
延迟: 0.0254秒/笔

吞吐量: 19156笔/秒



PS_C=18961, NO=OK:486045 Thread_num:990 TOTAL:486045 LA
PS_C=18980, NO=OK:487036 Thread_num:991 TOTAL:487036 LA
PS_C=19000, NO=OK:488028 Thread_num:992 TOTAL:488028 LA
PS_C=19019, NO=OK:489021 Thread_num:993 TOTAL:489021 LA
PS_C=19039, NO=OK:490015 Thread_num:994 TOTAL:490015 LA
PS_C=19058, NO=OK:491010 Thread_num:995 TOTAL:491010 LA
PS_C=19078, NO=OK:492006 Thread_num:996 TOTAL:492006 LA
PS_C=19097, NO=OK:493003 Thread_num:997 TOTAL:493003 LA
PS_C=19117, NO=OK:494001 Thread_num:998 TOTAL:494001 LA
PS_C=19136, NO=OK:495000 Thread_num:999 TOTAL:495000 LA
PS_C=19156, NO=OK:496000 Thread_num:1000 TOTAL:496000 L

5, P=OK:479901 Thread_num:988 TOTAL:484066 LATENCY:0.02543891228472773
3, P=OK:480846 Thread_num:989 TOTAL:485055 LATENCY:0.025438373562744317
, P=OK:481792 Thread_num:990 TOTAL:486045 LATENCY:0.025437829199399326
, P=OK:482739 Thread_num:991 TOTAL:487036 LATENCY:0.02543727876103045
5, P=OK:483687 Thread_num:992 TOTAL:488028 LATENCY:0.02543673516629097
, P=OK:484636 Thread_num:993 TOTAL:489021 LATENCY:0.025436186735325238
, P=OK:485586 Thread_num:994 TOTAL:490015 LATENCY:0.02543564169202271
, P=OK:486537 Thread_num:995 TOTAL:491010 LATENCY:0.025435107988276084
, P=OK:487489 Thread_num:996 TOTAL:492006 LATENCY:0.025434565282549598
7, P=OK:488442 Thread_num:997 TOTAL:493003 LATENCY:0.025434015662877656
4, P=OK:489396 Thread_num:998 TOTAL:494001 LATENCY:0.0254334571831375
2, P=OK:490350 Thread_num:999 TOTAL:495000 LATENCY:0.025432886831206505



持续集成

bookstore期末作业报告

build passing codecov 93%

项目目录结构

```
bookstore
|-- be          后端
|-- model      功能实现
```



swzhangslg / bookstore



build passing

Current

Branches

Build History

Pull Requests

✓ master finish

🔗 #49 passed

🔗 Commit 1bd72b8

🔗 Compare fa9a34d...1bd72b8

🔗 Branch master

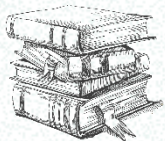
👤 swzhangslg

🕒 Ran for 6 min 46 sec

📅 less than a minute ago

🔧 </> Python: 3.7

🏠 AMD64



持续集成

```
374 fe/test/test_send_books.py::Test_send_books::test_can_not_send PASSED [ 96%]
375 fe/test/test_send_books.py::Test_send_books::test_none_order PASSED [ 98%]
376 fe/test/test_send_books.py::Test_send_books::test_no_seller PASSED [100%]
377
378 ===== 57 passed in 353.79s (0:05:53) =====
379 2021-01-12 21:39:44,035 [Thread-4375 ] [INFO ] 127.0.0.1 - - [12/Jan/2021 21:39:44] "GET /shutdown HTTP/1.1" 200 -
380 frontend end test
381 The command "coverage run --timid --branch --source fe,be --concurrency=thread -m pytest -v --ignore=fe/data" exited with 0.
382
383 $ coverage combine
385 $ coverage report
433 $ codecov
462 $ bash <(curl -s https://codecov.io/bash)
506
507 Done. Your build exited with 0.
```

after_success.1	0.08s
after_success.2	0.33s
after_success.3	1.09s
after_success.4	1.14s

testx



swzhangslg 9 minutes ago ✓ CI Pas

fa9a34d master 1d0e183

Diff

Files

Build

Graphs

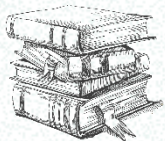
Files	≡	●	●	●	Coverage
be	748	634	21	93	84.76%
fe	1,154	1,139	8	7	98.70%
Project Totals (37 files)	1,902	1,773	29	100	93.22%



分工及版本控制

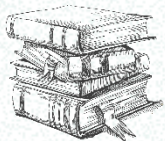
⊕ 项目分工

⊕ 版本控制



项目分工

项目成员	完成任务
张硕闻	数据库设计、卖家基础功能、订单发货收货功能、testcase编写测试、ppt及报告编写
杜涵悦	数据库设计、买家基础功能、手动取消订单功能、查询订单功能、自动取消订单功能、testcase编写测试、ppt及报告编写
陈煜	数据库设计、用户基础功能、图书搜索功能、testcase编写测试、ppt及报告编写



版本控制



具体使用如下：
合作开发，每位成员都是项目的合作者。
每个人建立自己的分支，将项目克隆到本地。

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Actions

Environments

Secrets

Moderation settings

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS

2 have access to this repository. [2 collaborators.](#)

Manage access

☐ Select all

Type ▾

☐

itstime

Collaborator

☐

Missjust

Collaborator

☐

itstime

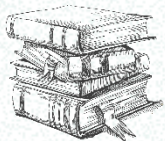
Collaborator

☐

Missjust

Collaborator

Invite a collaborator



版本控制



swzhangslg / bookstore

<> Code

! Issues

🔗 Pull requests

▶ Actions

📁 Projects

📖 Wiki

🛡 Security

📈 Insights

🔗 master ▾

🔗 4 branches

🏷 0 tags

Go to file

Add file ▾

📄 Code ▾

Switch branches/tags

×

Find or create a branch...

Branches

Tags

✓ master

default

dev_cy

dev_dhy

dev_zsw

[View all branches](#)

✖ 09aeb07 2 hours ago 🕒 47 commits

test

2 hours ago

finish

2 hours ago

finish

2 hours ago

finish

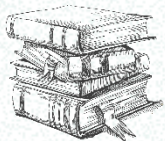
2 hours ago

test

2 hours ago

test

4 days ago



版本控制

项目开发pull request记录



8 Pull requests merged by 3 people

Dev dhy
#10 merged 3 hours ago

Dev zsw
#9 merged 22 hours ago

merge dev_cy
#8 merged yesterday

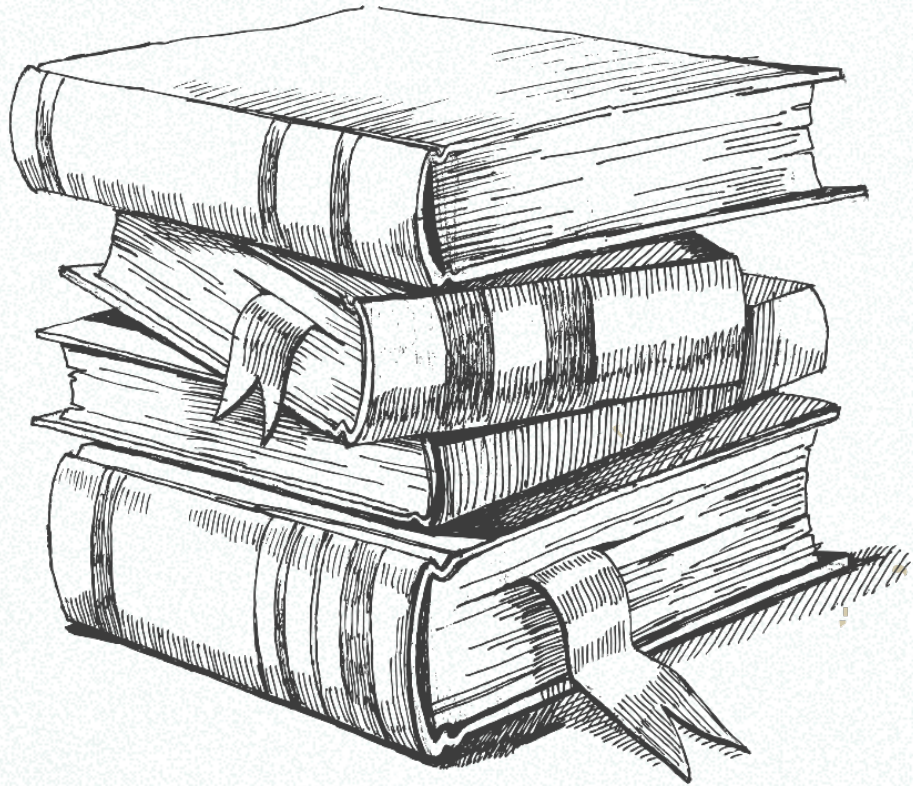
Dev dhy
#6 merged yesterday

Dev cy
#4 merged yesterday

update test
#3 merged 2 days ago

init database update
#2 merged 2 days ago

user update
#1 merged 3 days ago



感谢聆听

汇报时间：2021年1月13日

汇报人：张硕闻、陈煜、杜涵悦