# 目 录

[目 录 1](#_Toc105439197)

[引 言 3](#_Toc105439198)

[1数据库概念模型 4](#_Toc105439199)

[1.1 需求分析 4](#_Toc105439200)

[1.2 概念结构设计 4](#_Toc105439201)

[1.2.1抽象出实体，局部ER图 4](#_Toc105439202)

[1.2.2整体E-R图 6](#_Toc105439203)

[2数据库逻辑模型 6](#_Toc105439204)

[2.1将E-R图转化为关系模式 6](#_Toc105439205)

[3数据库物理设计与实施 7](#_Toc105439206)

[3.1创建数据库 7](#_Toc105439207)

[3.2创建和管理基本表 7](#_Toc105439208)

[3.2.1访问数据库 9](#_Toc105439209)

[4程序设计 11](#_Toc105439210)

[4.1创建应用程序 11](#_Toc105439211)

[4.2连接数据库 11](#_Toc105439212)

[4.3设计应用程序 11](#_Toc105439213)

[5测试用例 12](#_Toc105439214)

[6作业总结 15](#_Toc105439215)

[附录 15](#_Toc105439216)

# 引 言

图书管理系统

1、 背景资料

（1） 图书馆中有各种图书可以供学生借阅，需要设计一个管理系统提供给学生借书

（2） 每种图书都有书名、书号（ISBN）、一名或多名作者（译者）、出版社、定价和内容简介。

（3） 用户信息记录有借阅者的用户名密码等信息

（4） 凭账户借书，每次最多能借12本书。借书期限最长为7天。

2、 基本要求

能实现以下主要功能：

（1） 图书基本情况的录入。

（2） 实现借书功能。

（3） 实现还书功能。

（4） 实现对所有购进图书的分类查询和分类统计。

（5） 能够查询现有的图书的数量。

（6） 对超期的情况能自动给出提示信息。

（7） 图书管理员与借书者权限有区分。

# 1数据库概念模型

## 需求分析

- 系统管理员：

维护图书库存，添加新书，查看库存以及用户和游客能进行的一切操作

- 普通用户：

进行借阅，归还图书，查看已借阅图书和游客能进行的一切操作

- 游客：

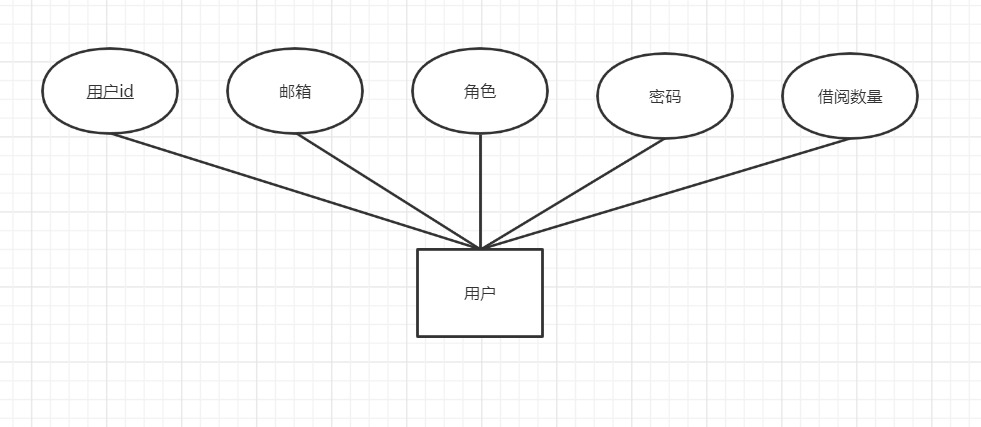
无需登录，可进行书籍查询

## 1.2 概念结构设计

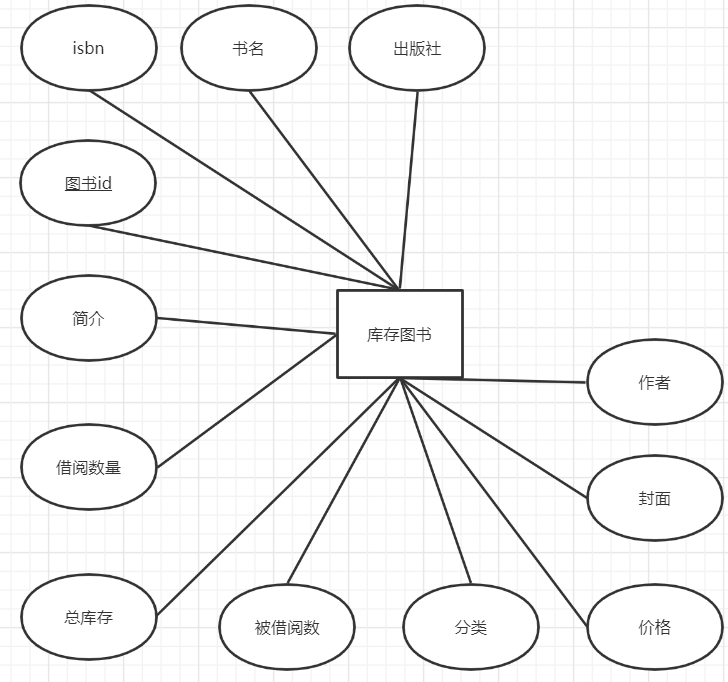
### 1.2.1抽象出实体，局部ER图

根据分析，图书管理系统主要应当包括3个实体：

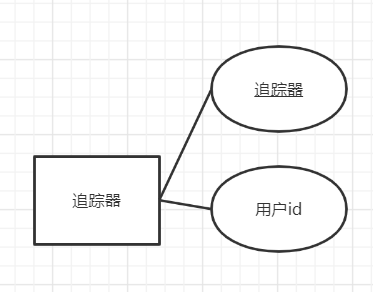
用户，书籍库存，用户追踪器



用户实体及其属性



库存图书实体及其属性



追踪器实体及其属性

### 图示 描述已自动生成1.2.2整体E-R图

# 2数据库逻辑模型

## 2.1将E-R图转化为关系模式

用户：（用户id，邮箱，角色，密码，借阅数量）为用户实体对应的关系模式，其中用户id是用户关系的主键

库存书籍：（图书id，isbn，简介，封面，书名，作者，出版社，价格，库存，被借阅数量）为库存书籍实体对于的关系模式，其中图书id是库存书籍关系的主键

借出图书：（id，用户id，isbn，时间戳）其中id为借出图书关系的主键

追踪器：（用户id，追踪器）其中用户id为追踪器的主键

# 3数据库物理设计与实施

## 3.1创建数据库

## 3.2创建和管理基本表

借出图书表：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 默认 | 注释 |
| id 主键 | INT | 自增 |  |
| uid | INT |  | 借阅的用户id |
| isbn | BIGINT(20) |  | 图书ISBN |
| time | TIMESTAMP | 当前时间戳 | 借出时间 |

库存表：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 默认 | 注释 |
| bookid 主键 | INT | 自增 | 图书id |
| isbn 索引 | BIGINT(20) |  | 图书ISBN |
| bookname | TEXT |  | 书名 |
| authors | TEXT |  | 作者 |
| description | TEXT |  | 简介 |
| photo | TEXT |  | 封面照片链接 |
| publisher | TEXT |  | 出版社 |
| price | TEXT |  | 价格 |
| category | TEXT |  | 分类 |
| stock | INT |  | 总库存 |
| borrowed | INT | 0 | 借出数 |

检查约束：

|  |  |  |
| --- | --- | --- |
| 名称 | 子句 | 注释 |
| CC1 | (`borrowed` <= `stock`) | 限定书籍被借阅次数 |

追踪器

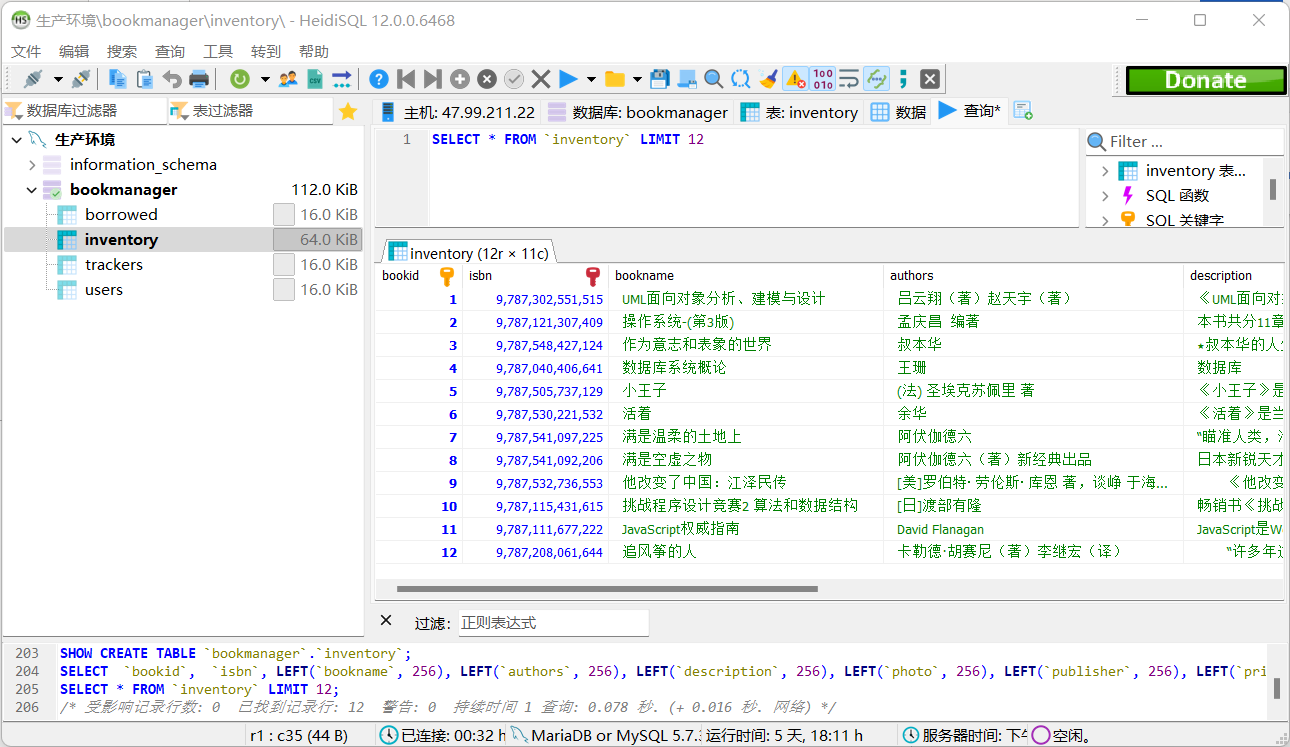
|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 默认 | 注释 |
| uid 主键 | INT |  | 用户id |
| tracker | TEXT |  | 追踪器 |

用户表

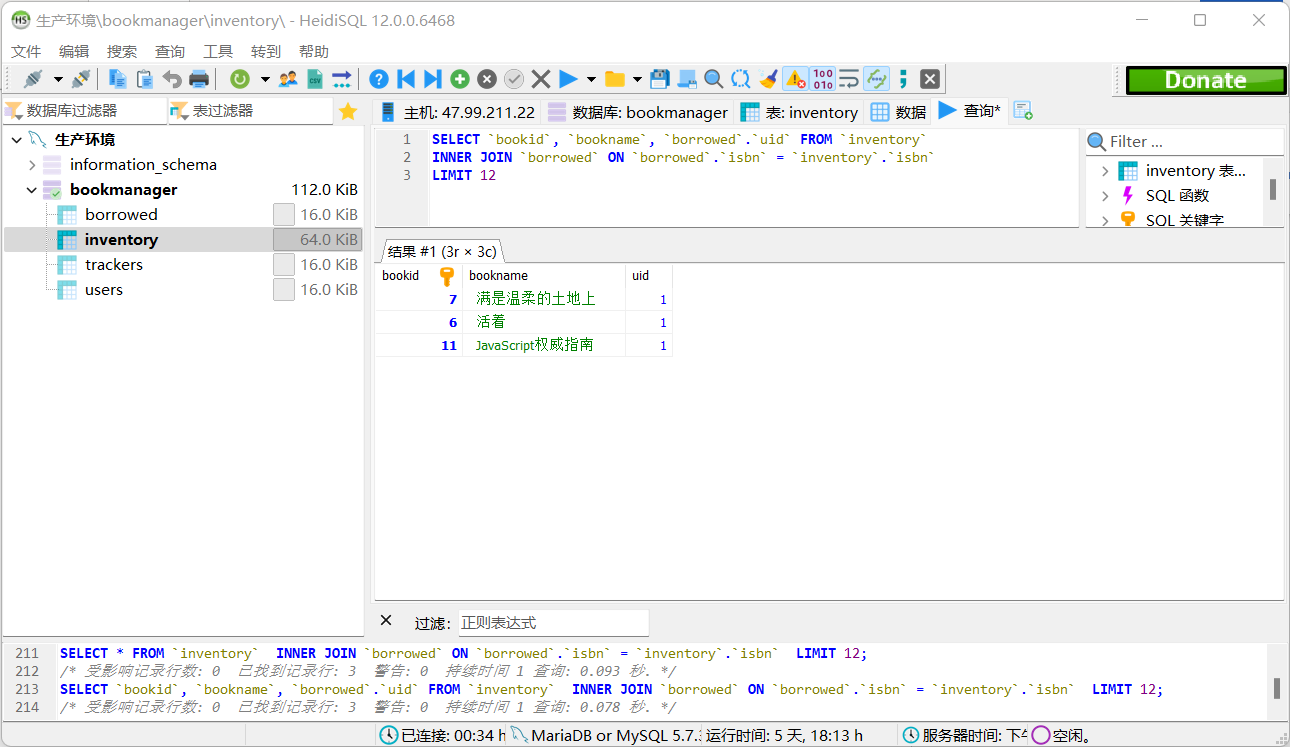
|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数据类型 | 默认 | 注释 |
| uid 主键 | INT | 自增 | 用户id |
| name | TEXT |  | 用户名 |
| email | TEXT |  | 邮箱 |
| pwd | TEXT |  | 密码Hash |
| role | INT |  | 用户权限 |
| borrowed | INT | 0 | 借阅数量 |

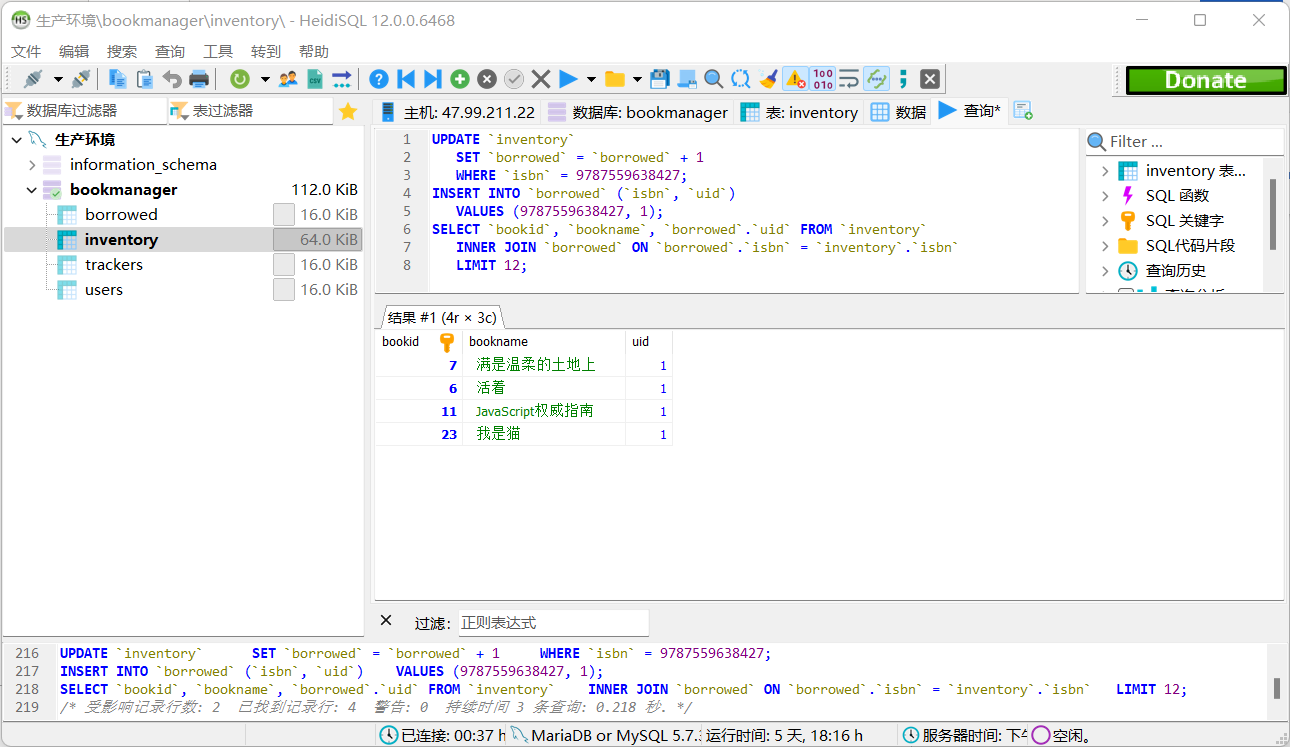
### 3.2.1访问数据库

对数据库进行访问，查询库存

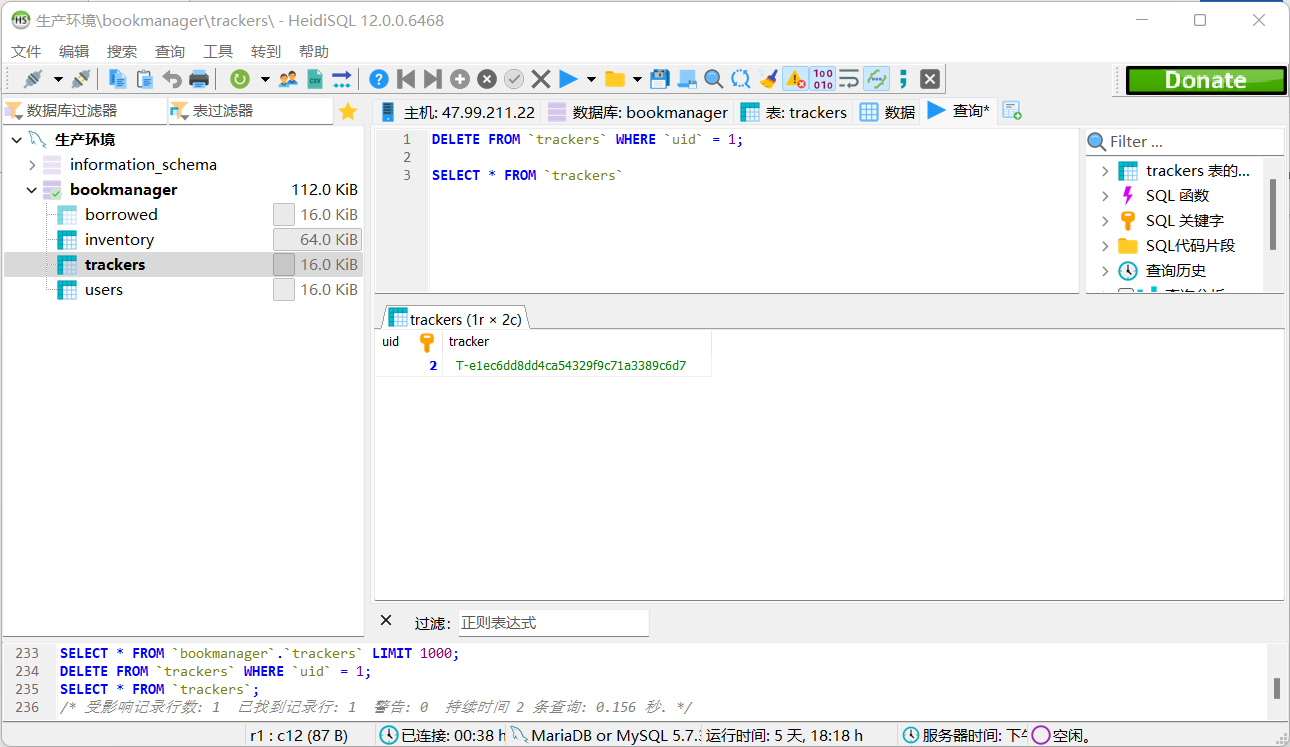


查询被借阅书籍



进行借阅操作

登出uid为1的用户



# 4程序设计

## 4.1创建应用程序

前端：npx create-react-app client

后端：mkdir server && cd server && npm init -y

## 4.2连接数据库

安装数据库驱动组件：npm i mysql2 —save

const mysql = require(‘mysql2’)

const connection = mysql.createConnection({

host: ‘localhost’,

user: ‘bookmanager’,

password: ‘bookmanager’,

database: ‘bookmanager’

})

connection.connect(err => {

// Error handler

})

## 4.3设计应用程序

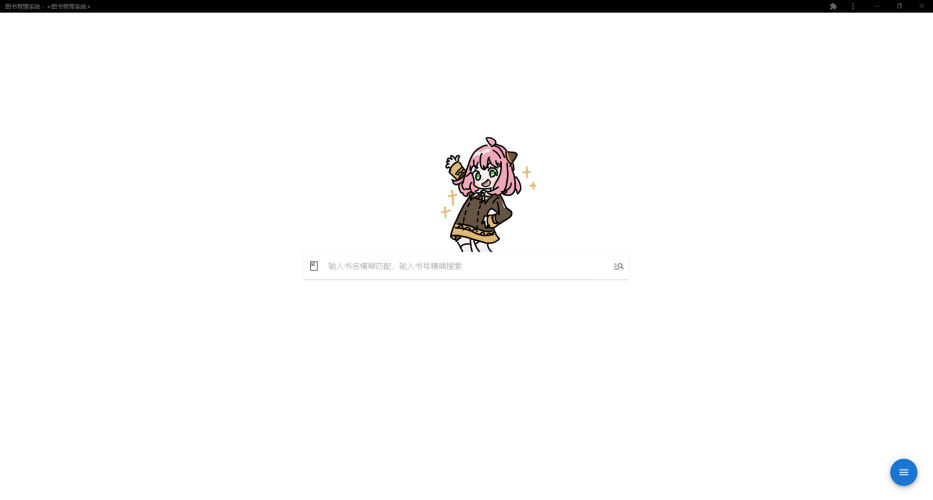
界面UI为采用React + Material UI制作的单页面应用程序，与后端程序的通讯使用websockst协议（主要是为了实现登陆时输入用户名与注册用户匹配时弹出密码框的功能）

服务端使用Node.js实现，数据库采用Mysql8.0

# 5测试用例

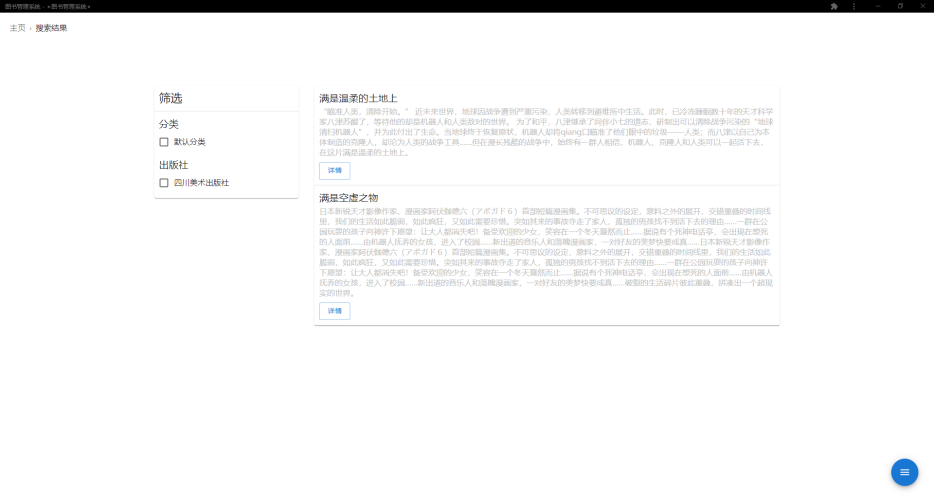
创建两个用户，分别为管理员和普通用户，首先使用管理员账号添加书本，并进行借阅，归还等操作

再使用用户账户登录进行搜索，借阅和归还操作

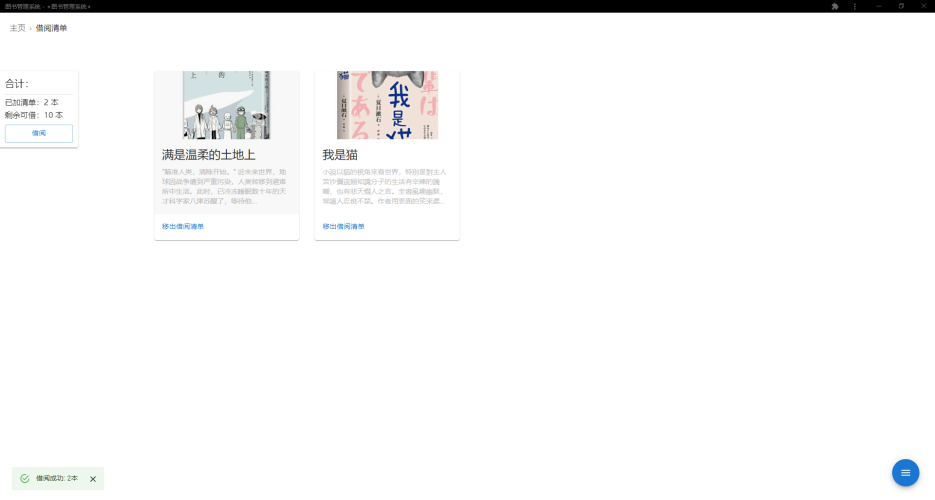


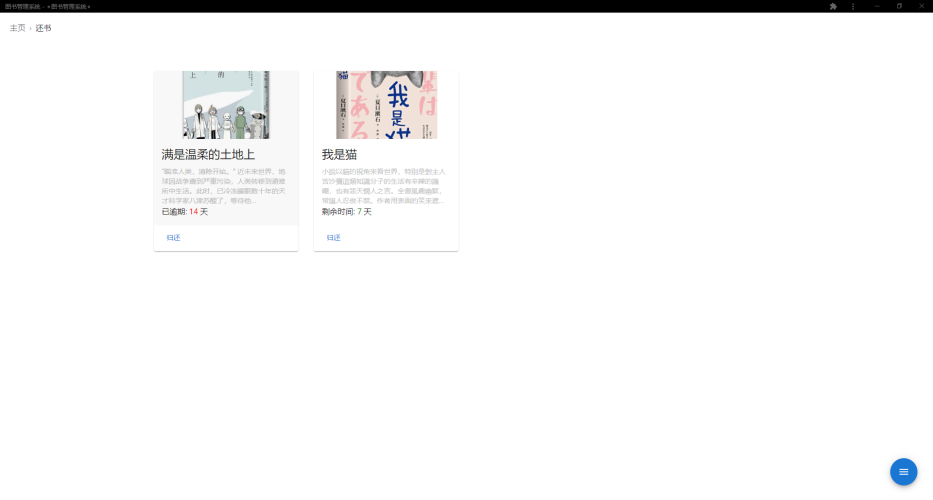


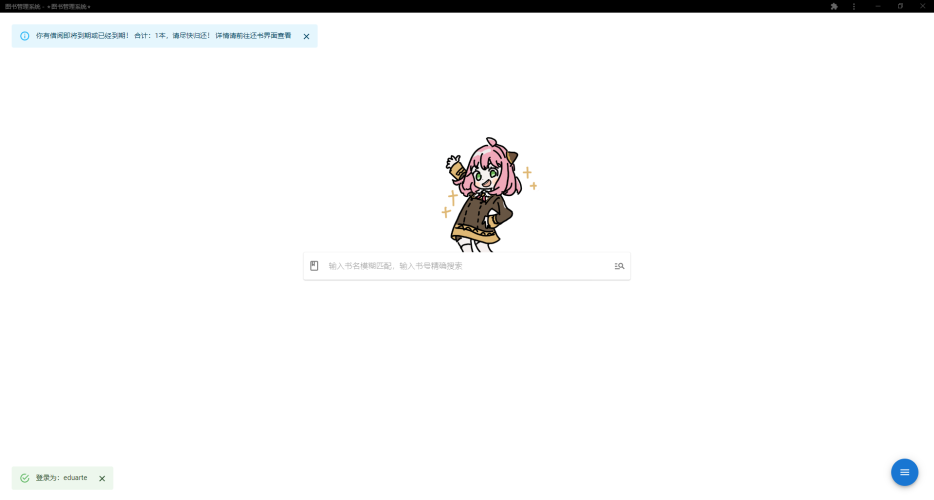


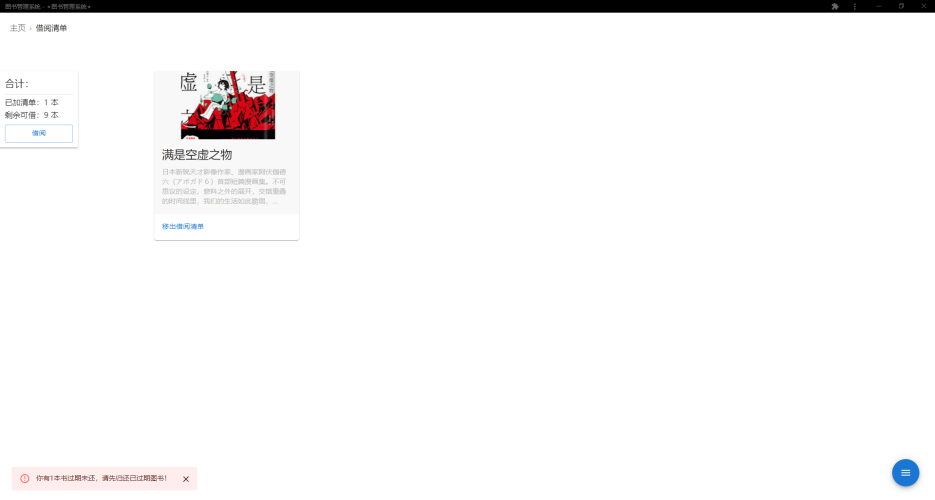












# 6作业总结

在这次课程设计之中，通过编写程序实践了学习到的数据库相关知识，通过具体的实例设计分析，是自己真正将所学转化为了实际的作品，加深了对数据库的理解；

在我看来，数据库课程设计主要的⽬标是利⽤课程中学到的数据库知识和技术较好的开发设计出数据库应⽤系统，去解决各⾏各业信息化处理的要求。通过这次的课程设计，可以巩固我们对数据库基本原理和基础理论的理解，掌握数据库应⽤系统设计开发的基本⽅法，进⼀步提⾼我们综合运⽤所学知识的能⼒。

# 附录

建表代码：

**CREATE** **TABLE** `users` (

`uid` INT(11) NOT NULL AUTO\_INCREMENT,

`name` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`email` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`pwd` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`role` INT(11) NULL DEFAULT NULL,

`borrowed` **INT**(11) **NULL** **DEFAULT** '0',

PRIMARY KEY (`uid`) USING BTREE

)

**COLLATE**='utf8mb4\_bin'

ENGINE=InnoDB

AUTO\_INCREMENT=1;

**CREATE** **TABLE** `trackers` (

`uid` **INT**(11) **NOT** **NULL**,

`tracker` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

PRIMARY KEY (`uid`) USING BTREE

)

**COLLATE**='utf8mb4\_bin'

ENGINE=InnoDB;

**CREATE** **TABLE** `inventory` (

`bookid` INT(11) NOT NULL AUTO\_INCREMENT,

`isbn` BIGINT(20) NULL DEFAULT NULL,

`bookname` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`authors` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`description` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`photo` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`publisher` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`price` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`category` **TEXT** **NULL** **DEFAULT** **NULL** **COLLATE** 'utf8mb4\_bin',

`stock` INT(11) NULL DEFAULT NULL,

`borrowed` **INT**(11) **NULL** **DEFAULT** '0',

PRIMARY KEY (`bookid`) USING BTREE,

UNIQUE INDEX `isbn` (`isbn`) USING BTREE,

**CONSTRAINT** `CC1` **CHECK ((**`borrowed` **<=** `stock`**))**

)

**COLLATE**='utf8mb4\_bin'

ENGINE=InnoDB

AUTO\_INCREMENT=1;

**CREATE** **TABLE** `borrowed` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`uid` INT(11) NULL DEFAULT NULL,

`isbn` BIGINT(20) NULL DEFAULT NULL,

`time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`id`) USING BTREE

)

**COLLATE**='utf8mb4\_bin'

ENGINE=InnoDB

AUTO\_INCREMENT=1;