深圳大学考试答题纸

(以论文、报告等形式考核专用)

二〇二四～二〇二五　学年度第＿一＿学期

课程编号 1500370003　课程名称 数据库系统　主讲教师 Basker George　评分 ＿＿＿

学号 ＿＿＿＿＿　姓名 ＿＿＿＿＿　专业年级 ＿＿＿＿＿＿＿＿＿＿

学号 ＿＿＿＿＿　姓名 ＿＿＿＿＿　专业年级 ＿＿＿＿＿＿＿＿＿＿

教师评语：The mark obtained for different categories as per the grading policy are as follows

序号	**评分点　（分值）**	**Marks Obtained** **成绩**
(1)	第一部分 ：Requirements Analysis（10 分）	
(2)	第二部分: ER Diagram/ Nomalization and Convert to Relations (20 分)	
(3)	第三部分: SQL DDL/ All the different kinds of DML query from PPT （30 分）	
(4)	第四部分: Application System Demonstration （20 分）	
(5)	第五部分: Report is completed as per the requirement and PPT（20 分）	
合计	100	

Basker George
10 January 2025

题目: Library management system

组员个人总分权重分配表:

排序	姓名	学号	项目个人权重	个人最后得分	评分人
1			50%		
2			50%		
3					

任务分工及所完成的状况。

（2 人总分为 200% *教师评分, 3 人总分为 270% *教师评分）

个人总分权重分配表

（1 人总分为 100%，2 人为 200%, 3 人为 270%）

排序	姓名	学号	项目个人权重
1			100%
2			100%
3			%

我组成员总共___2_____名，权重总和为: 200 %

一、数据库大作业 Report

Part1: Requirement Analysis:

Objective:

In today's era of rapid digital development, the dissemination and acquisition of knowledge has become more convenient and diversified.However, libraries, schools, enterprises and other organisations still face many challenges in library management.Traditional manual records and simple spreadsheet management methods can no longer meet the increasing number of books, frequent borrowing and returning operations and diversified user needs.With the continuous enrichment of book resources and the gradual expansion of user groups, the demand for an efficient, intelligent, comprehensive and easy-to-operate library management system is particularly urgent.

This library management system aims to provide one-stop solutions for all kinds of library management scenarios through the integration of advanced information technology, realize the fine management and optimal allocation of library resources, improve the utilization rate and circulation efficiency of books, and at the same time provide users with more convenient, fast and personalized book searching and borrowing services to adapt to the modern society's high standards of knowledge management and information services, and to enhance the library management system's efficiency.management organisations, promote the dissemination and sharing of knowledge, and satisfy the readers' demand for reading resources.

I. Functional requirement

(i) Administrator Functions

i.Book Inventory

Administrators are able to accurately enter details of newly purchased books through the system's intuitive and convenient interface.In addition to the basic information such as book title, author, publisher, publication date, ISBN number, book category, price, etc., the system should also support the entry of more detailed information such as the binding form (hardcover, paperback, etc.), the number of pages, the number of words, and the classification number of the Chinese Graphic Law, etc., in order to satisfy the requirements of different types of libraries for the comprehensiveness of the information of the books.The system should be equipped with an automatic checking function to ensure that the format and content of the entered information comply with certain specifications and avoid data inaccuracies caused by human errors.At the same time, it supports scanning ISBN code and other ways to quickly get some book information, improve the efficiency of stocking, and automatically generate stocking records, including stocking time, stocking batch number, etc., which is convenient for the subsequent inventory management and statistical analysis.

For batch stocking operation, the system should provide flexible import templates and batch editing functions.Administrators can follow the specified template format in Excel and other spreadsheet software to organise the book information, and then import it into the system at one time, and the system will automatically carry out data parsing and stocking operations.In terms of batch editing, administrators can make batch modifications and improvements to the imported book information, such as uniformly adjusting the price of a certain batch of books, modifying book categories, etc., which greatly reduces repetitive work and improves work efficiency.

ii.Book Deletion

When books need to be removed from the collection due to damage, loss, elimination or other reasons, the administrator can execute the deletion operation in the system.Before deletion, the system not only pops up a detailed confirmation box showing all the key information of the book to be deleted, including the title, author, publisher, ISBN number and the current inventory quantity, but also provides the borrowing history of the book (e.g., the most recent borrowing time, the borrower, etc.), so that the administrator can verify the information of the book again to ensure the accuracy and reasonableness of the deletion operation, and to prevent the deletion of valuable books or the damage caused by the incompleteness of data.books or misoperation caused by incomplete data.

The system should record the log of the deletion operation in detail, including the time of deletion, operator, deleted book information and other basic content, but also record the reasons for deletion (such as damage, loss, etc., the specific circumstances of the description) and the relevant approval process information (if there is an approval process).These logs should be kept for a long time for subsequent enquiry and traceability, and also provide strong data support for the library's resource management and auditing work.

iii.Book Updates

When any information of books changes, such as price adjustment due to market fluctuation, change of stock quantity due to new purchase or loan return, change of book category due to reclassification or subject adjustment, change of author information due to new research results or change of pen name, etc., administrators can make use of this function to make a comprehensive and accurate update of book information.

The system should provide an intelligent update guide interface, which can quickly locate the book records that need to be updated according to the key information of the book (e.g. ISBN number), and automatically load the existing book information for administrators to compare and modify.For the modified information, the system should carry out real-time verification and saving, and generate detailed update records, including update time, update content, operators, etc., to ensure the traceability of the update operation and data consistency.At the same time, the update operation should trigger the related business logic processing, such as automatic update of the loanable status after the change of inventory quantity, automatic synchronisation to the financial statistics module after price adjustment, etc., in order to ensure the linkage and accuracy of the data of the whole system.

(ii) User Functions

i.Book search

Users can find the books they need in the system in many ways, such as precise query by key information such as title, author, publisher, ISBN number, book category, etc., or fuzzy query by inputting keywords (this system uses query according to the title of the book).The system should quickly and accurately return detailed information about the book according to the query conditions entered by the user, including the basic information of the book, inventory status, etc., so as to facilitate the user's access to book resources.

ii.Book Borrowing

After finding the desired book, the user should be able to easily initiate a borrowing request if there is a borrowable copy of the book.The system first verifies the user's borrowing eligibility, including whether there are overdue books and whether the maximum number of books can be borrowed, etc. If the user meets the borrowing conditions, the system automatically generates a borrowing record to record the borrowing time.If the user meets the borrowing conditions, the system automatically generates a borrowing record, recording the borrowing time, the return time (calculated according to the borrowing period set by the library), and the borrowed books and user information, and updating the inventory status of the books, reducing the number of borrowable copies by 1. At the same time, the system should send a notification of the successful borrowing to the user, informing him of the details of the borrowing and the deadline for returning the books, which can be notified by means of internal mail, SMS or e-mail.The system shall send notification to the user of the successful borrowing, informing the details of borrowing and the deadline for returning the book, which can be notified by means of internal mail, SMS or email, etc., to ensure that the user can know the borrowing situation in time.

iii.Return of books

Users should be able to return books through the system after reading them within the borrowing period.Users only need to confirm the return of books by selecting the corresponding borrowing records in the system, and the system will automatically update the inventory status of the books, add 1 to the number of borrowable copies, and record the time of returning the books.For overdue returns, the system shall automatically calculate the fine amount according to the library's overdue fine rules, generate the corresponding fine record, and notify the user to pay the fine.In the process of returning books, if the books are found to be damaged, the administrator should be able to record the damage information through the system and determine the amount of compensation according to the library's damage compensation system, and the user should handle the compensation accordingly.

II.Non-functional requirement

(i) Performance requirements

The system should have a rapid response capability to ensure that the administrator's operations can be completed within a short period of time to ensure work efficiency.

In the case of highly concurrent user queries, the system can run stably to ensure the accuracy and timeliness of the query results, and at the same time ensure the reasonable use of system resources to avoid the phenomenon of lag or crash.

(ii) Security requirements

The system should set up a strict user rights management mechanism, administrators and users have different login accounts and passwords, and passwords should be encrypted storage to prevent information leakage.The administrator has all the operation rights of the library management system, and the user only has the authority to find books, to ensure the security and integrity of the system data.

The system should be equipped with data backup and recovery functions, regular backup of book information and other important data, and stored in a safe and reliable location.In the event of data loss or damage, the data can be quickly recovered to ensure the normal operation of the system and data availability.

(iii) Ease of use requirements

The system interface design should be simple, intuitive and friendly, and the operation process should be simple and easy to understand, so as to facilitate administrators and users to quickly get started with the use.Whether it is management operations such as book entry, deletion, updating, or user book search operations, the user's operating steps and input workload should be minimised to improve the user experience.

Part2: ER Diagram/Nomalization and Convert to Relations

Database Design:

Our system entities are: users, user_profiles, books, borrowing_records, book categories, publishers, authors, fines, reservations, book reviews, and administrator operation logs. There is also a book-author association table for linking a book to multiple authors.

Entity Relationship:

One-to-one relationship:

i.users and user_profiles

A user corresponds to a user profile, which is a one-to-one relationship.

One-to-many and many-to-one relationships:

i.books and borrowing_records

A book can have multiple borrowing records, and a borrowing record corresponds to a book.A book can have zero to many borrowing records.

ii.books and book_reviews

A book can have multiple reviews, and a review corresponds to a book.A book can have from zero to many reviews.

iii.books and reservations

A book can have multiple reservations, and a reservation corresponds to a book.A book can have zero to many reservations.

iv.books and fines

A book can result in multiple fines (e.g., multiple overdue), and a single fine corresponds to a book.A book can have zero to multiple fines.

v.authors and book_authors

An author can be associated with multiple books.An author can be associated with zero to many books.

vi.publishers and books

A publisher can publish more than one book, and a book is published by a single publisher.A publisher can publish zero to many books.

vii.book_categories and books

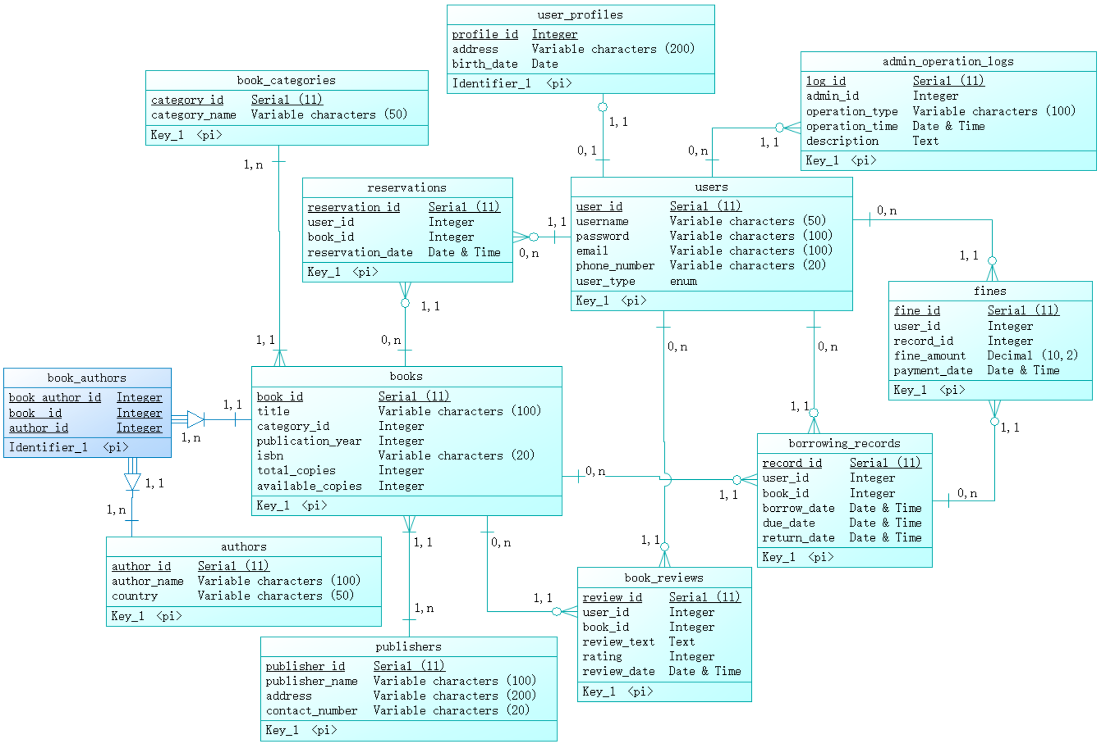
A book category can contain multiple books, and a book belongs to a single category.A book category can contain zero to many books.

Many-to-many relationship:

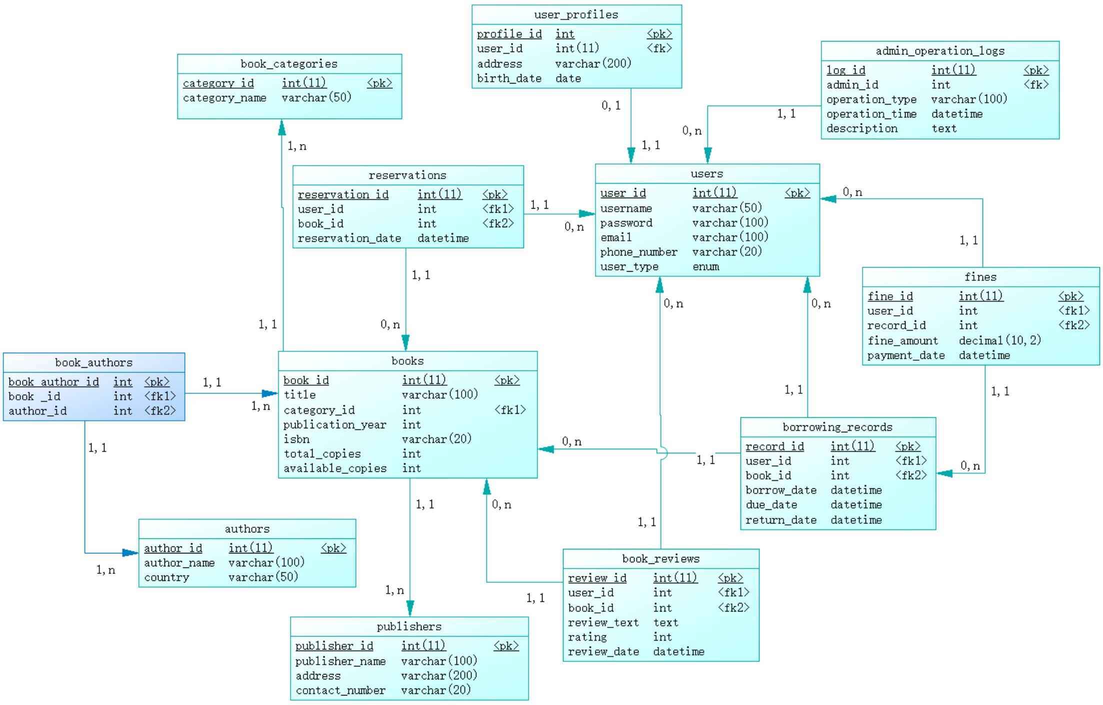
i.books and book_authors

A book can have multiple authors, and an author can write multiple books.This relationship is implemented through the book_authors association table.

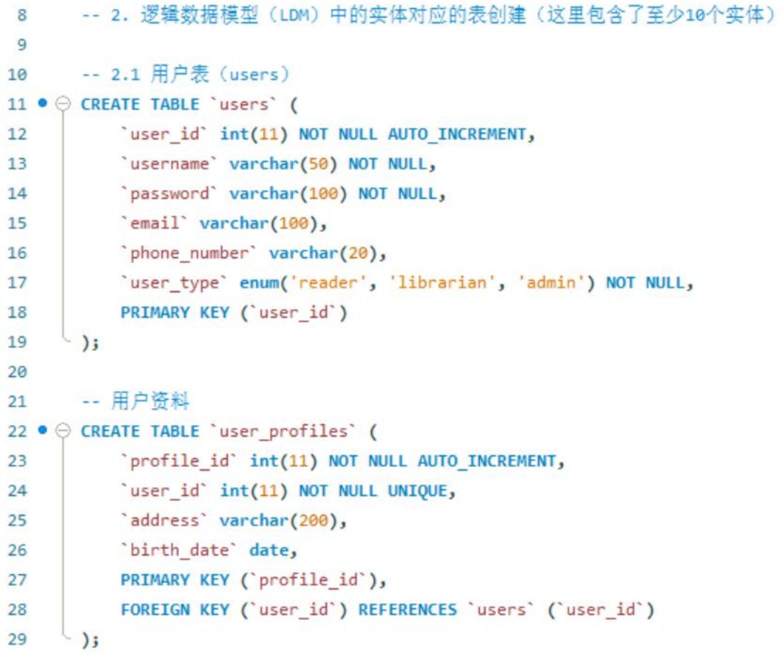
♦Logical Data Model (LDM)



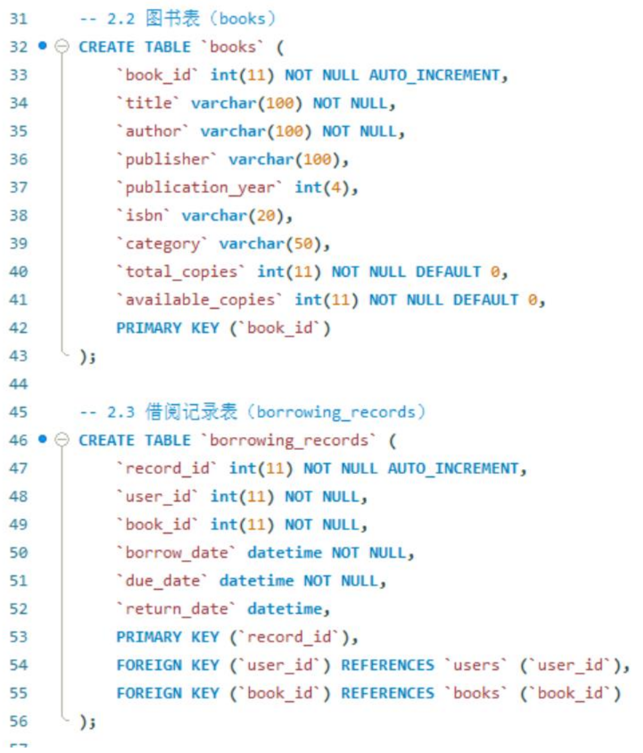
♦Physical Data Model (PDM)



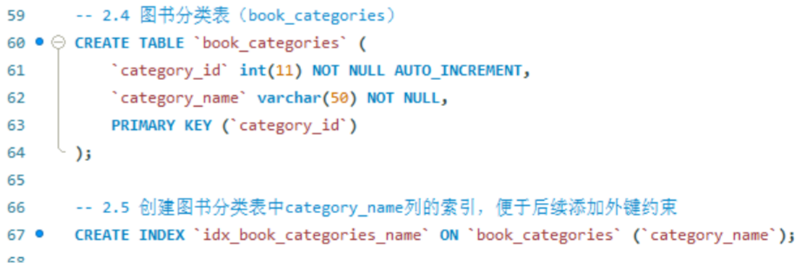
♦MySQL Script Generation:



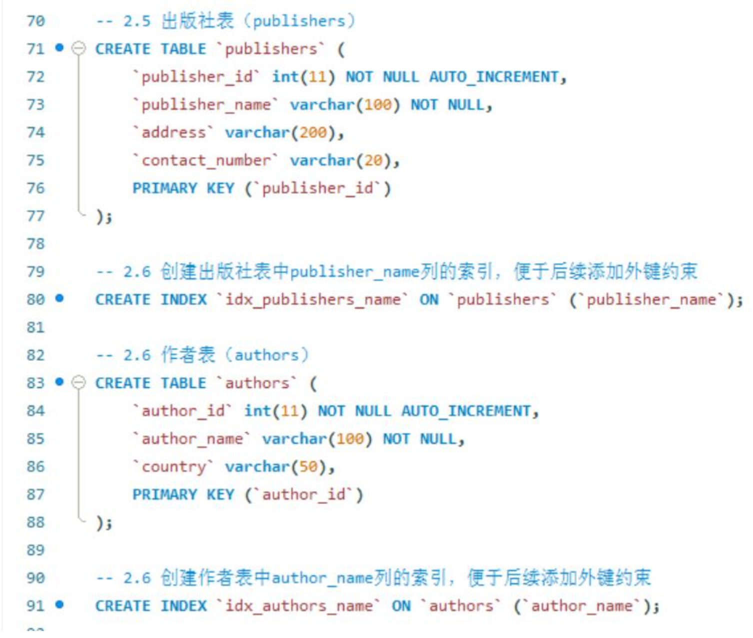
```
-- 2. 逻辑数据模型（LDM）中的实体对应的表创建（这里包含了至少10个实体）

-- 2.1 用户表（users）
CREATE TABLE `users` (
    `user_id` int(11) NOT NULL AUTO_INCREMENT,
    `username` varchar(50) NOT NULL,
    `password` varchar(100) NOT NULL,
    `email` varchar(100),
    `phone_number` varchar(20),
    `user_type` enum('reader', 'librarian', 'admin') NOT NULL,
    PRIMARY KEY (`user_id`)
);

-- 用户资料
CREATE TABLE `user_profiles` (
    `profile_id` int(11) NOT NULL AUTO_INCREMENT,
    `user_id` int(11) NOT NULL UNIQUE,
    `address` varchar(200),
    `birth_date` date,
    PRIMARY KEY (`profile_id`),
    FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`)
);
```



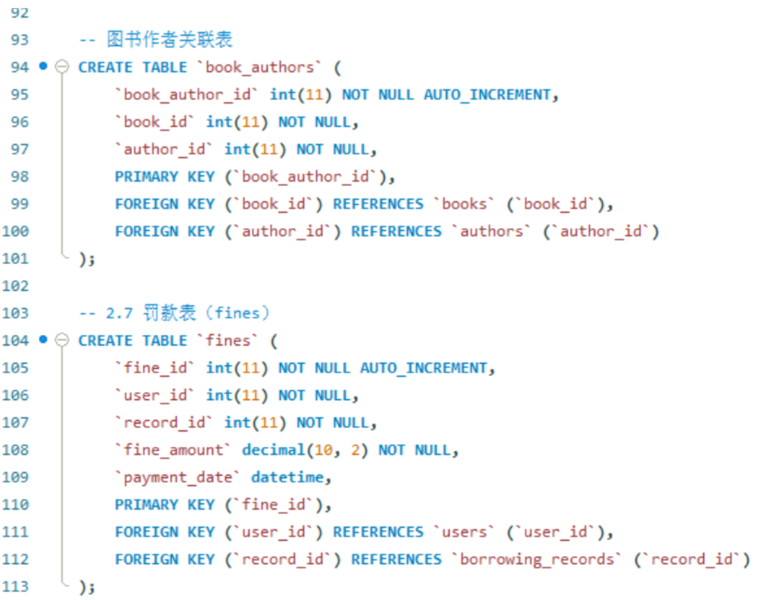
```sql
-- 2.2 图书表（books）
CREATE TABLE `books` (
    `book_id` int(11) NOT NULL AUTO_INCREMENT,
    `title` varchar(100) NOT NULL,
    `author` varchar(100) NOT NULL,
    `publisher` varchar(100),
    `publication_year` int(4),
    `isbn` varchar(20),
    `category` varchar(50),
    `total_copies` int(11) NOT NULL DEFAULT 0,
    `available_copies` int(11) NOT NULL DEFAULT 0,
    PRIMARY KEY (`book_id`)
);

-- 2.3 借阅记录表（borrowing_records）
CREATE TABLE `borrowing_records` (
    `record_id` int(11) NOT NULL AUTO_INCREMENT,
    `user_id` int(11) NOT NULL,
    `book_id` int(11) NOT NULL,
    `borrow_date` datetime NOT NULL,
    `due_date` datetime NOT NULL,
    `return_date` datetime,
    PRIMARY KEY (`record_id`),
    FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`),
    FOREIGN KEY (`book_id`) REFERENCES `books` (`book_id`)
);
```



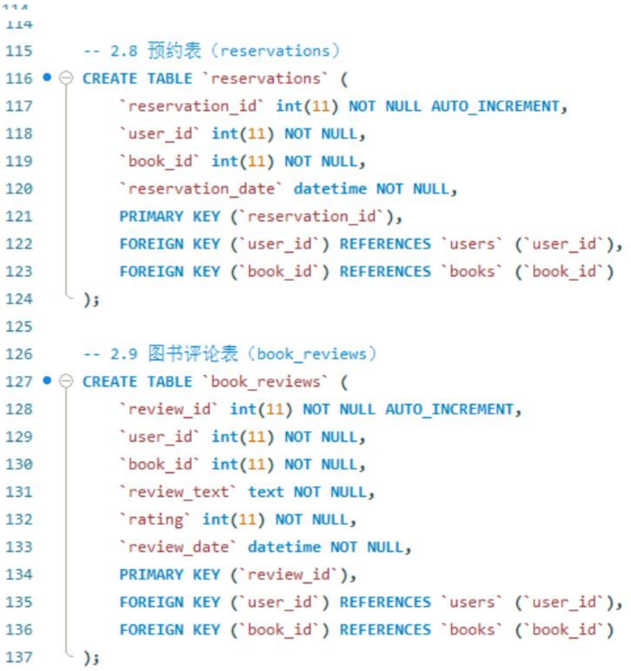
```sql
-- 2.4 图书分类表（book_categories）
CREATE TABLE `book_categories` (
    `category_id` int(11) NOT NULL AUTO_INCREMENT,
    `category_name` varchar(50) NOT NULL,
    PRIMARY KEY (`category_id`)
);

-- 2.5 创建图书分类表中category_name列的索引，便于后续添加外键约束
CREATE INDEX `idx_book_categories_name` ON `book_categories` (`category_name`);
```



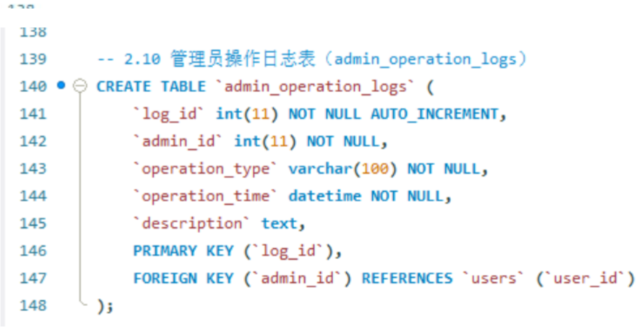
```sql
-- 2.5 出版社表（publishers）
CREATE TABLE `publishers` (
    `publisher_id` int(11) NOT NULL AUTO_INCREMENT,
    `publisher_name` varchar(100) NOT NULL,
    `address` varchar(200),
    `contact_number` varchar(20),
    PRIMARY KEY (`publisher_id`)
);

-- 2.6 创建出版社表中publisher_name列的索引，便于后续添加外键约束
CREATE INDEX `idx_publishers_name` ON `publishers` (`publisher_name`);

-- 2.6 作者表（authors）
CREATE TABLE `authors` (
    `author_id` int(11) NOT NULL AUTO_INCREMENT,
    `author_name` varchar(100) NOT NULL,
    `country` varchar(50),
    PRIMARY KEY (`author_id`)
);

-- 2.6 创建作者表中author_name列的索引，便于后续添加外键约束
CREATE INDEX `idx_authors_name` ON `authors` (`author_name`);
```



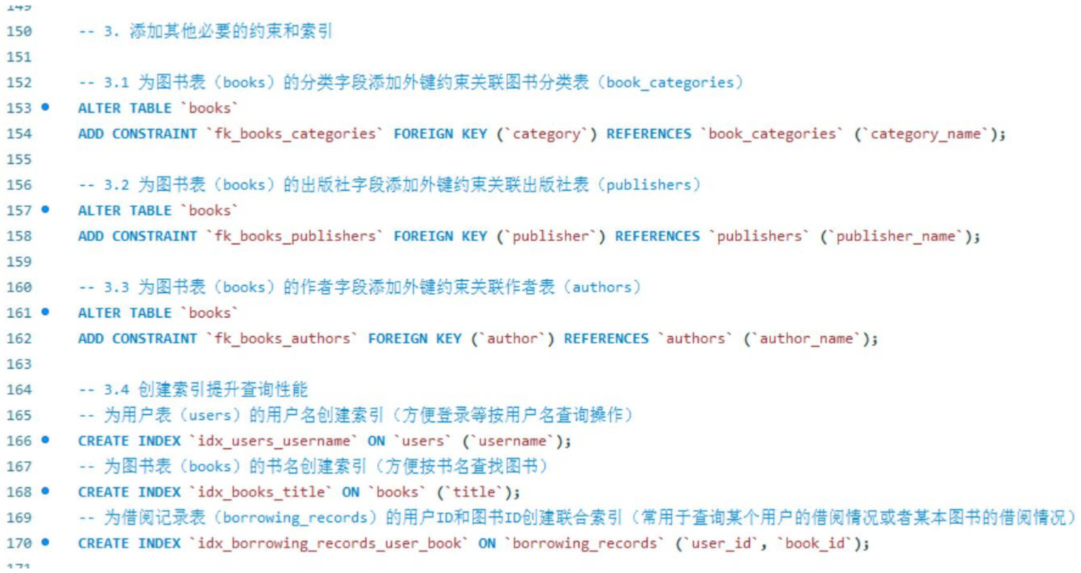
```sql

-- 图书作者关联表
CREATE TABLE `book_authors` (
    `book_author_id` int(11) NOT NULL AUTO_INCREMENT,
    `book_id` int(11) NOT NULL,
    `author_id` int(11) NOT NULL,
    PRIMARY KEY (`book_author_id`),
    FOREIGN KEY (`book_id`) REFERENCES `books` (`book_id`),
    FOREIGN KEY (`author_id`) REFERENCES `authors` (`author_id`)
);

-- 2.7 罚款表（fines）
CREATE TABLE `fines` (
    `fine_id` int(11) NOT NULL AUTO_INCREMENT,
    `user_id` int(11) NOT NULL,
    `record_id` int(11) NOT NULL,
    `fine_amount` decimal(10, 2) NOT NULL,
    `payment_date` datetime,
    PRIMARY KEY (`fine_id`),
    FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`),
    FOREIGN KEY (`record_id`) REFERENCES `borrowing_records` (`record_id`)
);
```



```sql

-- 2.8 预约表（reservations）
CREATE TABLE `reservations` (
    `reservation_id` int(11) NOT NULL AUTO_INCREMENT,
    `user_id` int(11) NOT NULL,
    `book_id` int(11) NOT NULL,
    `reservation_date` datetime NOT NULL,
    PRIMARY KEY (`reservation_id`),
    FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`),
    FOREIGN KEY (`book_id`) REFERENCES `books` (`book_id`)
);

-- 2.9 图书评论表（book_reviews）
CREATE TABLE `book_reviews` (
    `review_id` int(11) NOT NULL AUTO_INCREMENT,
    `user_id` int(11) NOT NULL,
    `book_id` int(11) NOT NULL,
    `review_text` text NOT NULL,
    `rating` int(11) NOT NULL,
    `review_date` datetime NOT NULL,
    PRIMARY KEY (`review_id`),
    FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`),
    FOREIGN KEY (`book_id`) REFERENCES `books` (`book_id`)
);
```



```sql

-- 2.10 管理员操作日志表（admin_operation_logs）
CREATE TABLE `admin_operation_logs` (
    `log_id` int(11) NOT NULL AUTO_INCREMENT,
    `admin_id` int(11) NOT NULL,
    `operation_type` varchar(100) NOT NULL,
    `operation_time` datetime NOT NULL,
    `description` text,
    PRIMARY KEY (`log_id`),
    FOREIGN KEY (`admin_id`) REFERENCES `users` (`user_id`)
);
```

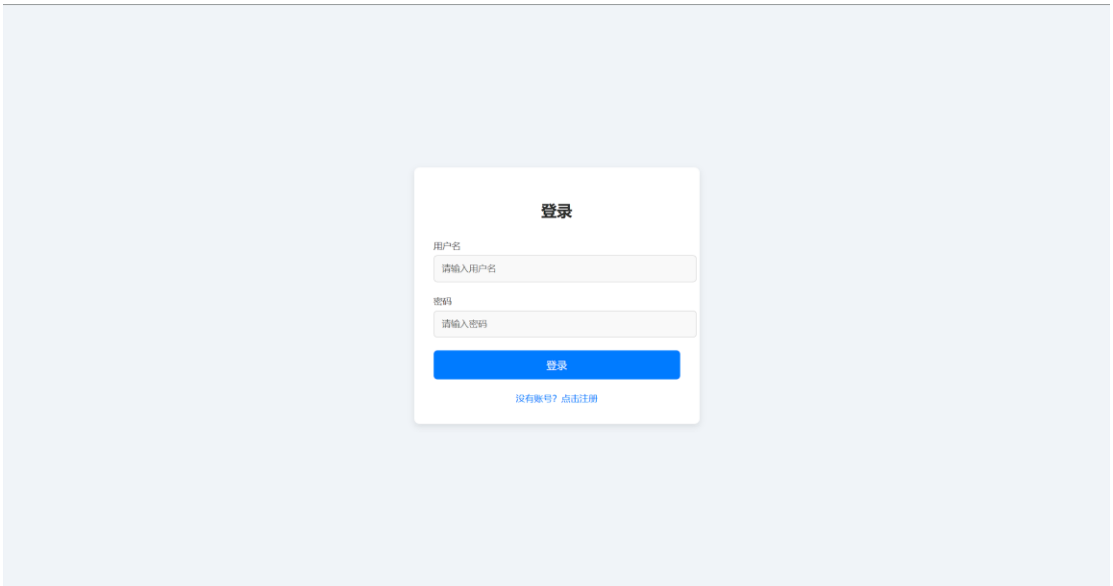


```sql
-- 3. 添加其他必要的约束和索引

-- 3.1 为图书表（books）的分类字段添加外键约束关联图书分类表（book_categories）
ALTER TABLE `books`
ADD CONSTRAINT `fk_books_categories` FOREIGN KEY (`category`) REFERENCES `book_categories` (`category_name`);

-- 3.2 为图书表（books）的出版社字段添加外键约束关联出版社表（publishers）
ALTER TABLE `books`
ADD CONSTRAINT `fk_books_publishers` FOREIGN KEY (`publisher`) REFERENCES `publishers` (`publisher_name`);

-- 3.3 为图书表（books）的作者字段添加外键约束关联作者表（authors）
ALTER TABLE `books`
ADD CONSTRAINT `fk_books_authors` FOREIGN KEY (`author`) REFERENCES `authors` (`author_name`);

-- 3.4 创建索引提升查询性能
-- 为用户表（users）的用户名创建索引（方便登录等按用户名查询操作）
CREATE INDEX `idx_users_username` ON `users` (`username`);
-- 为图书表（books）的书名创建索引（方便按书名查找图书）
CREATE INDEX `idx_books_title` ON `books` (`title`);
-- 为借阅记录表（borrowing_records）的用户ID和图书ID创建联合索引（常用于查询某个用户的借阅情况或者某本图书的借阅情况）
CREATE INDEX `idx_borrowing_records_user_book` ON `borrowing_records` (`user_id`, `book_id`);
```

Part3:Functional Requirements and Technical Requirements

I. User management

a) login



登录

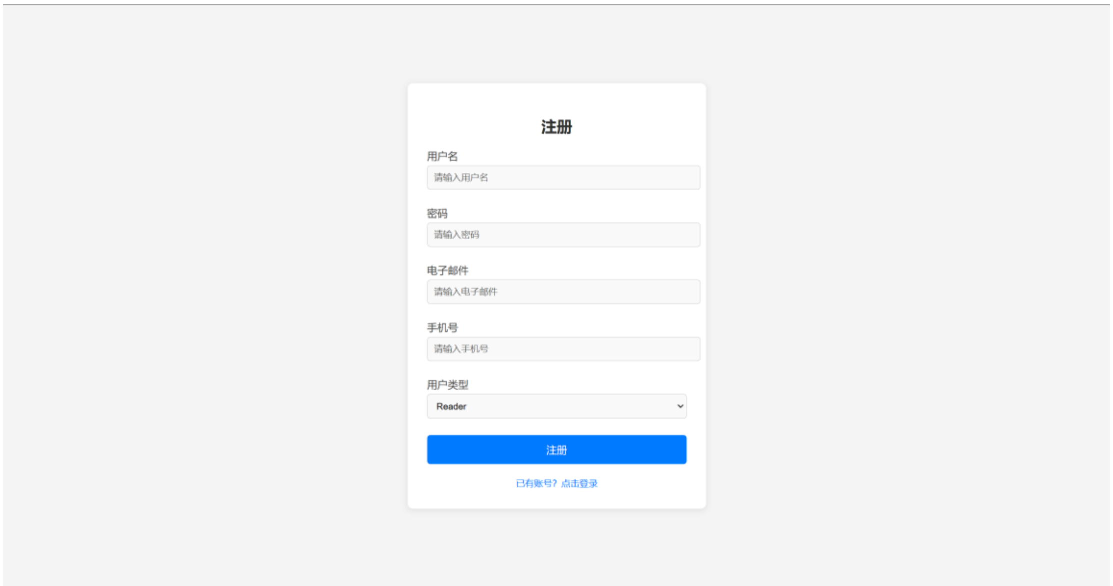
用户名

请输入用户名

密码

请输入密码

b) registration



注册

用户名

请输入用户名

密码

请输入密码

电子邮件

请输入电子邮件

手机号

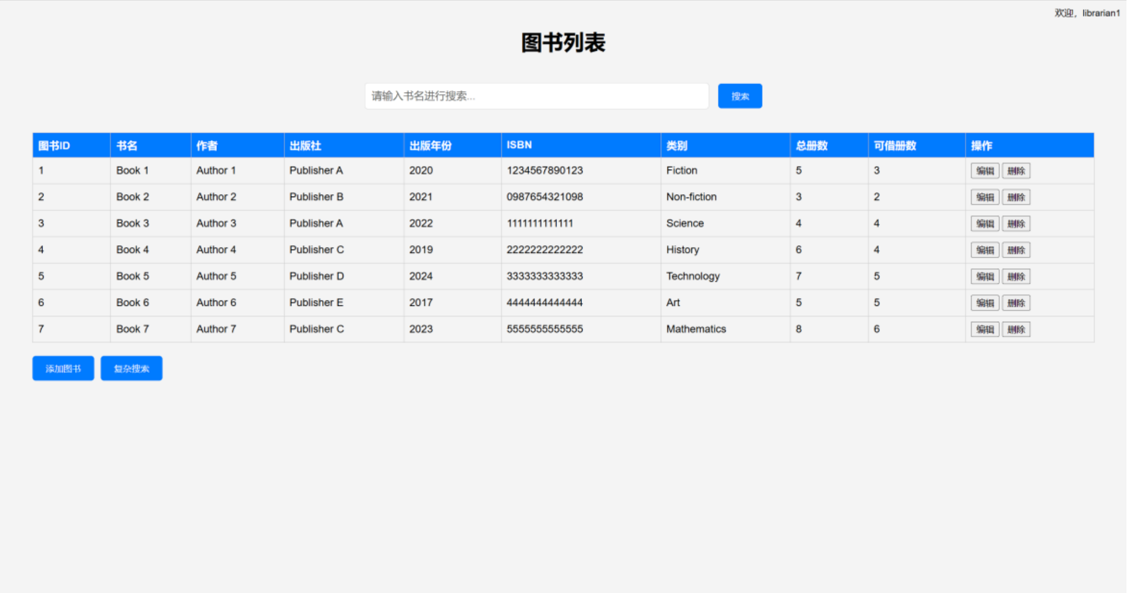
请输入手机号

用户类型

Reader

II. Data management

a) Read



欢迎，librarian1

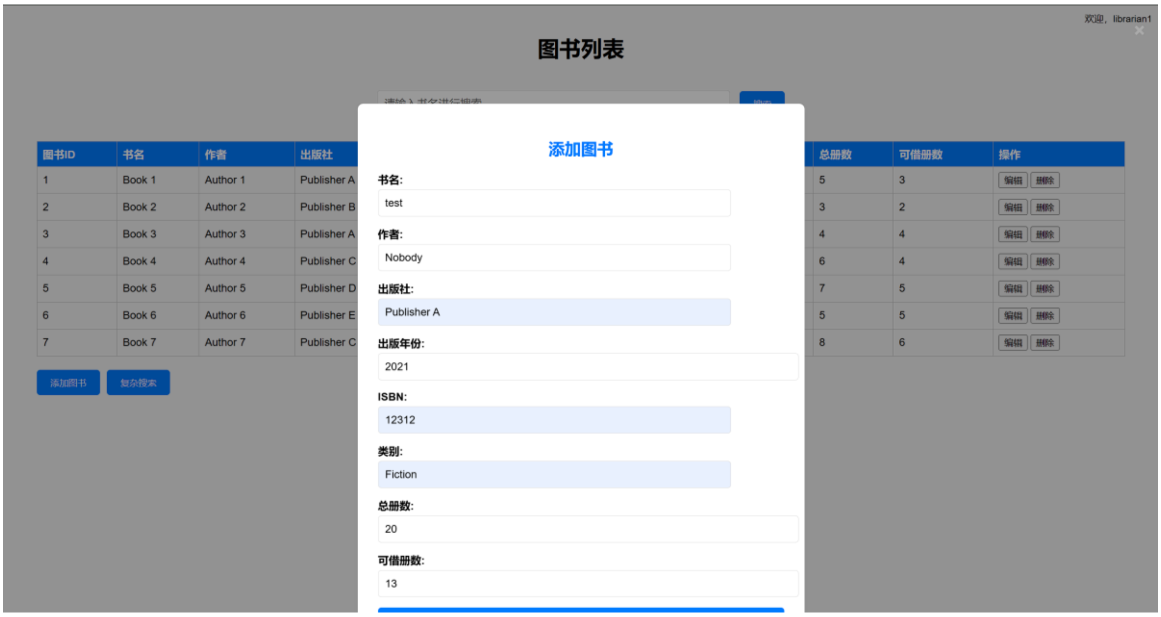
图书列表

请输入书名进行搜索... 　搜索

图书ID	书名	作者	出版社	出版年份	ISBN	类别	总册数	可借册数	操作
1	Book 1	Author 1	Publisher A	2020	1234567890123	Fiction	5	3	编辑 删除
2	Book 2	Author 2	Publisher B	2021	0987654321098	Non-fiction	3	2	编辑 删除
3	Book 3	Author 3	Publisher A	2022	1111111111111	Science	4	4	编辑 删除
4	Book 4	Author 4	Publisher C	2019	2222222222222	History	6	4	编辑 删除
5	Book 5	Author 5	Publisher D	2024	3333333333333	Technology	7	5	编辑 删除
6	Book 6	Author 6	Publisher E	2017	4444444444444	Art	5	5	编辑 删除
7	Book 7	Author 7	Publisher C	2023	5555555555555	Mathematics	8	6	编辑 删除

添加图书　复杂搜索

b) Create



欢迎，librarian1

图书列表

图书ID	书名	作者	出版社	总册数	可借册数	操作
1	Book 1	Author 1	Publisher A	5	3	编辑 删除
2	Book 2	Author 2	Publisher B	3	2	编辑 删除
3	Book 3	Author 3	Publisher A	4	4	编辑 删除
4	Book 4	Author 4	Publisher C	6	4	编辑 删除
5	Book 5	Author 5	Publisher D	7	5	编辑 删除
6	Book 6	Author 6	Publisher E	5	5	编辑 删除
7	Book 7	Author 7	Publisher C	8	6	编辑 删除

添加图书　复杂搜索

添加图书

书名: test

作者: Nobody

出版社: Publisher A

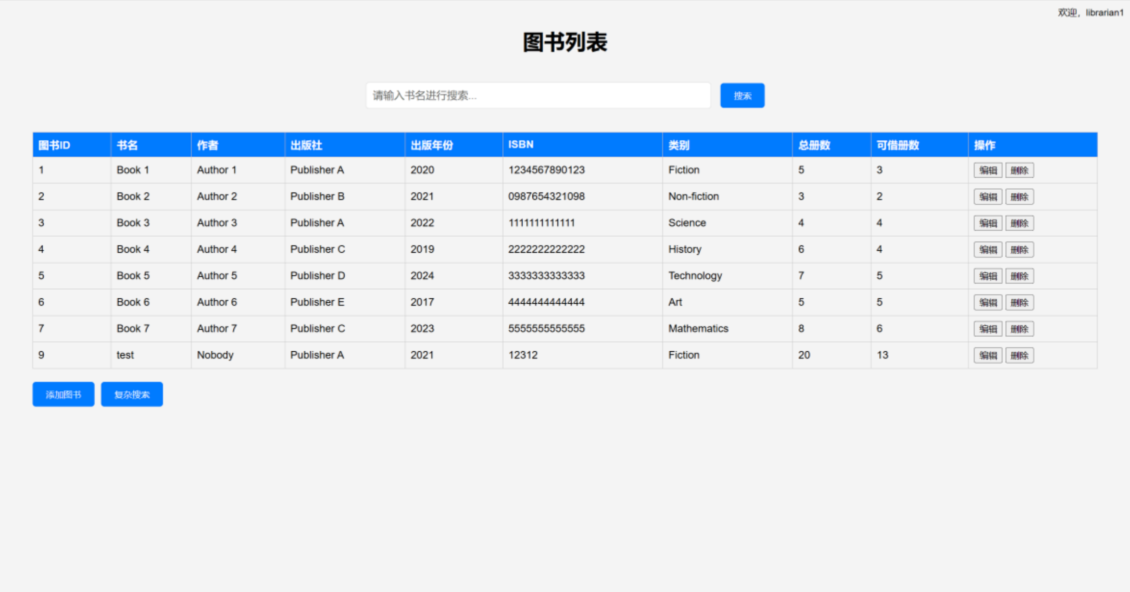
出版年份: 2021

ISBN: 12312

类别: Fiction

总册数: 20

可借册数: 13



欢迎，librarian1

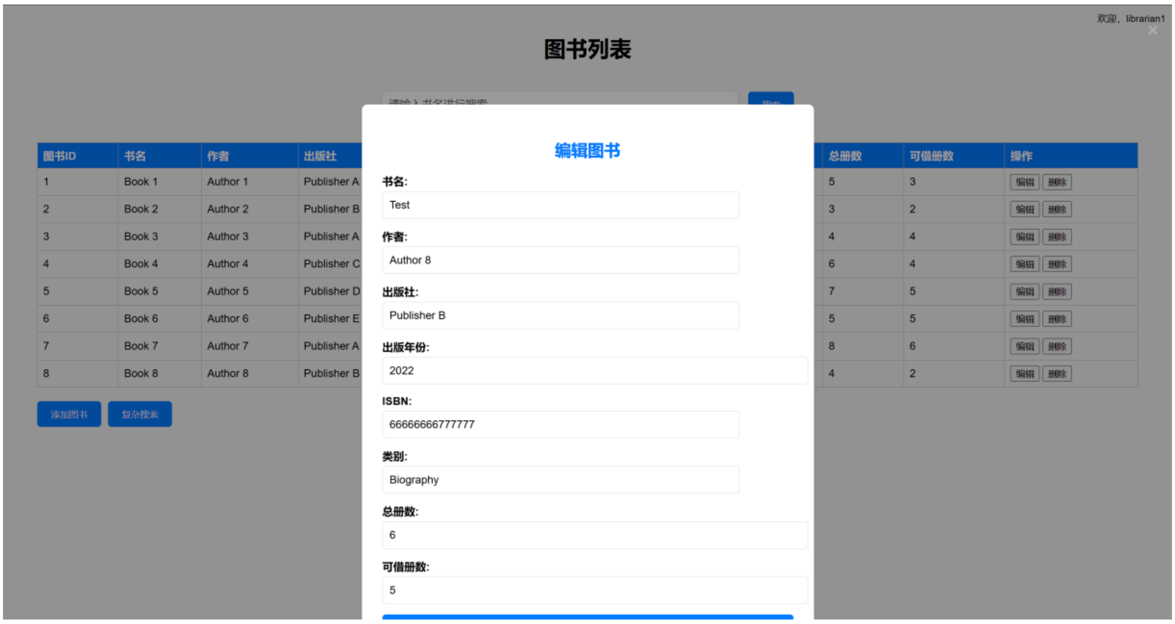
图书列表

请输入书名进行搜索... 　搜索

图书ID	书名	作者	出版社	出版年份	ISBN	类别	总册数	可借册数	操作
1	Book 1	Author 1	Publisher A	2020	1234567890123	Fiction	5	3	编辑 删除
2	Book 2	Author 2	Publisher B	2021	0987654321098	Non-fiction	3	2	编辑 删除
3	Book 3	Author 3	Publisher A	2022	1111111111111	Science	4	4	编辑 删除
4	Book 4	Author 4	Publisher C	2019	2222222222222	History	6	4	编辑 删除
5	Book 5	Author 5	Publisher D	2024	3333333333333	Technology	7	5	编辑 删除
6	Book 6	Author 6	Publisher E	2017	4444444444444	Art	5	5	编辑 删除
7	Book 7	Author 7	Publisher C	2023	5555555555555	Mathematics	8	6	编辑 删除
9	test	Nobody	Publisher A	2021	12312	Fiction	20	13	编辑 删除

添加图书　复杂搜索

c) Update



欢迎，librarian1

图书列表

编辑图书

书名:
Test

作者:
Author 8

出版社:
Publisher B

出版年份:
2022

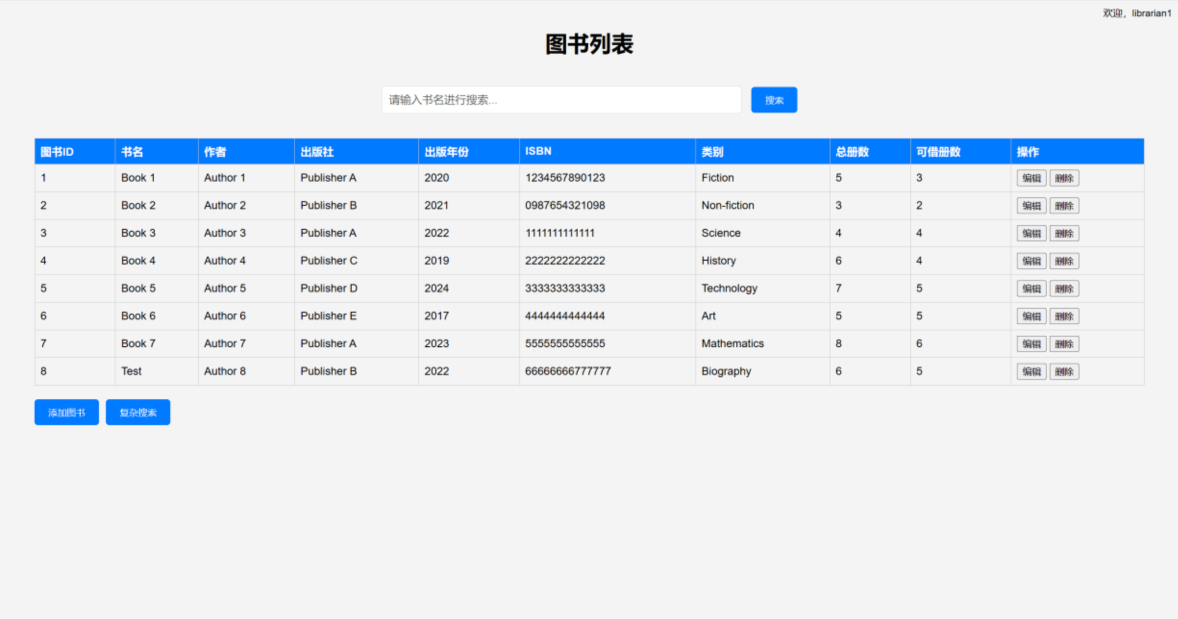
ISBN:
66666666777777

类别:
Biography

总册数:
6

可借册数:
5

图书ID	书名	作者	出版社	总册数	可借册数	操作
1	Book 1	Author 1	Publisher A	5	3	编辑 删除
2	Book 2	Author 2	Publisher B	3	2	编辑 删除
3	Book 3	Author 3	Publisher A	4	4	编辑 删除
4	Book 4	Author 4	Publisher C	6	4	编辑 删除
5	Book 5	Author 5	Publisher D	7	5	编辑 删除
6	Book 6	Author 6	Publisher E	5	5	编辑 删除
7	Book 7	Author 7	Publisher A	8	6	编辑 删除
8	Book 8	Author 8	Publisher B	4	2	编辑 删除

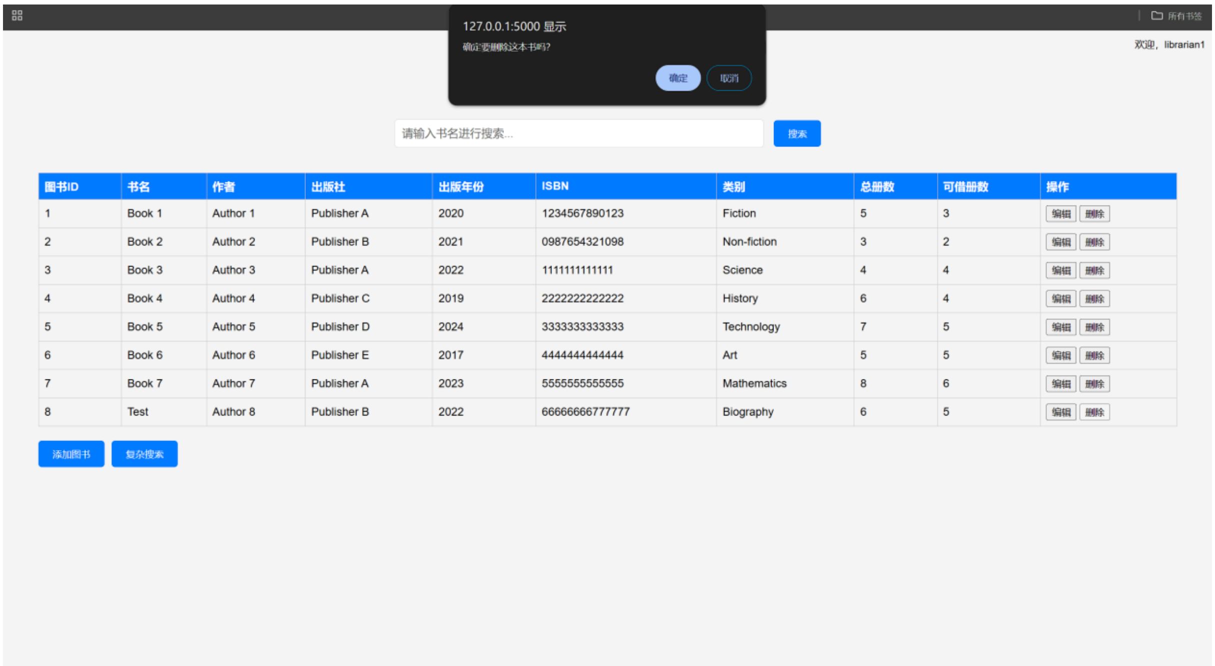


欢迎，librarian1

图书列表

图书ID	书名	作者	出版社	出版年份	ISBN	类别	总册数	可借册数	操作
1	Book 1	Author 1	Publisher A	2020	1234567890123	Fiction	5	3	编辑 删除
2	Book 2	Author 2	Publisher B	2021	0987654321098	Non-fiction	3	2	编辑 删除
3	Book 3	Author 3	Publisher A	2022	1111111111111	Science	4	4	编辑 删除
4	Book 4	Author 4	Publisher C	2019	2222222222222	History	6	4	编辑 删除
5	Book 5	Author 5	Publisher D	2024	3333333333333	Technology	7	5	编辑 删除
6	Book 6	Author 6	Publisher E	2017	4444444444444	Art	5	5	编辑 删除
7	Book 7	Author 7	Publisher A	2023	5555555555555	Mathematics	8	6	编辑 删除
8	Test	Author 8	Publisher B	2022	66666666777777	Biography	6	5	编辑 删除

d) Delete

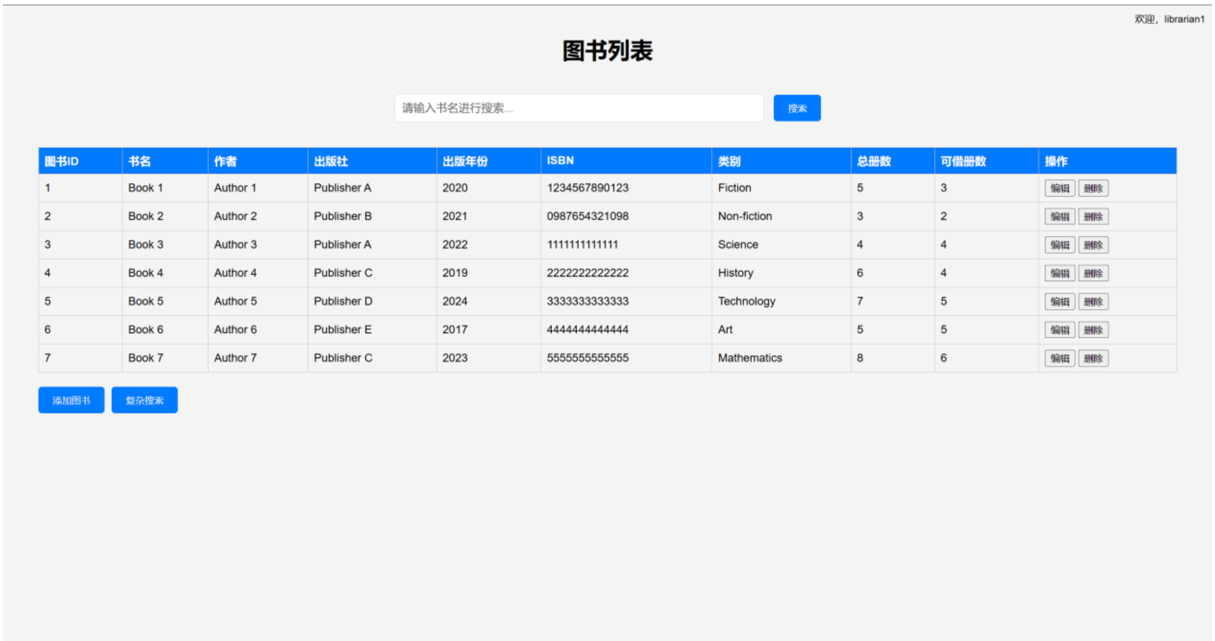


127.0.0.1:5000 显示

确定要删除这本书吗?

欢迎，librarian1

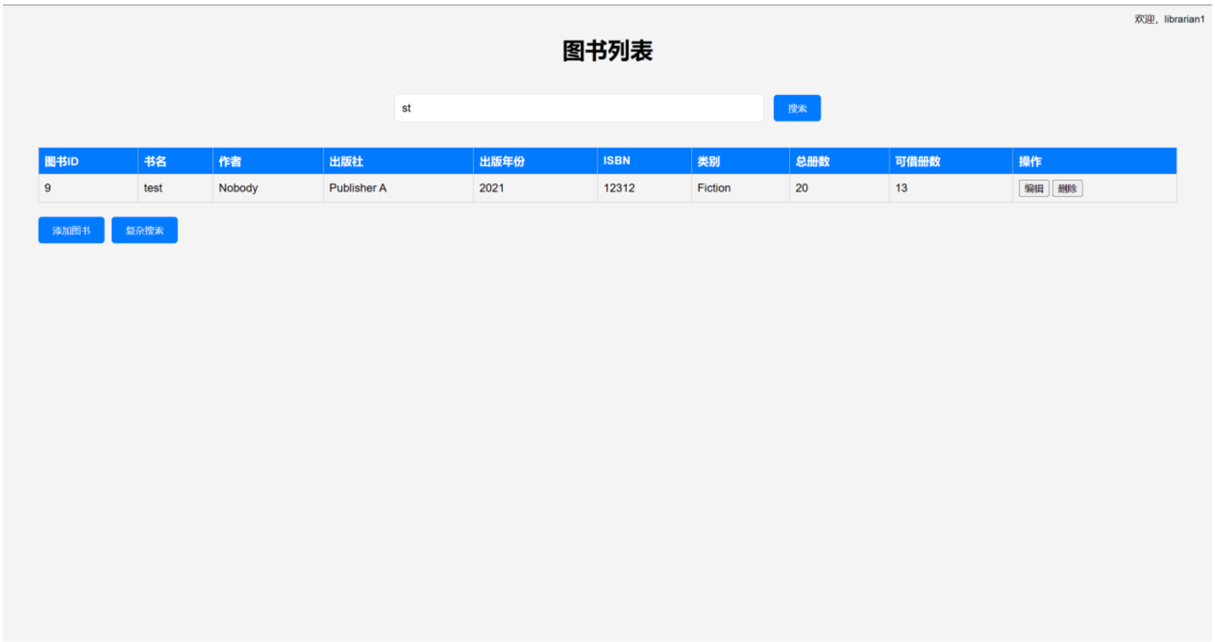
图书ID	书名	作者	出版社	出版年份	ISBN	类别	总册数	可借册数	操作
1	Book 1	Author 1	Publisher A	2020	1234567890123	Fiction	5	3	编辑 删除
2	Book 2	Author 2	Publisher B	2021	0987654321098	Non-fiction	3	2	编辑 删除
3	Book 3	Author 3	Publisher A	2022	1111111111111	Science	4	4	编辑 删除
4	Book 4	Author 4	Publisher C	2019	2222222222222	History	6	4	编辑 删除
5	Book 5	Author 5	Publisher D	2024	3333333333333	Technology	7	5	编辑 删除
6	Book 6	Author 6	Publisher E	2017	4444444444444	Art	5	5	编辑 删除
7	Book 7	Author 7	Publisher A	2023	5555555555555	Mathematics	8	6	编辑 删除
8	Test	Author 8	Publisher B	2022	66666666777777	Biography	6	5	编辑 删除



图书列表

图书ID	书名	作者	出版社	出版年份	ISBN	类别	总册数	可借册数	操作
1	Book 1	Author 1	Publisher A	2020	1234567890123	Fiction	5	3	编辑 删除
2	Book 2	Author 2	Publisher B	2021	0987654321098	Non-fiction	3	2	编辑 删除
3	Book 3	Author 3	Publisher A	2022	1111111111111	Science	4	4	编辑 删除
4	Book 4	Author 4	Publisher C	2019	2222222222222	History	6	4	编辑 删除
5	Book 5	Author 5	Publisher D	2024	3333333333333	Technology	7	5	编辑 删除
6	Book 6	Author 6	Publisher E	2017	4444444444444	Art	5	5	编辑 删除
7	Book 7	Author 7	Publisher C	2023	5555555555555	Mathematics	8	6	编辑 删除

e) Locate



图书列表

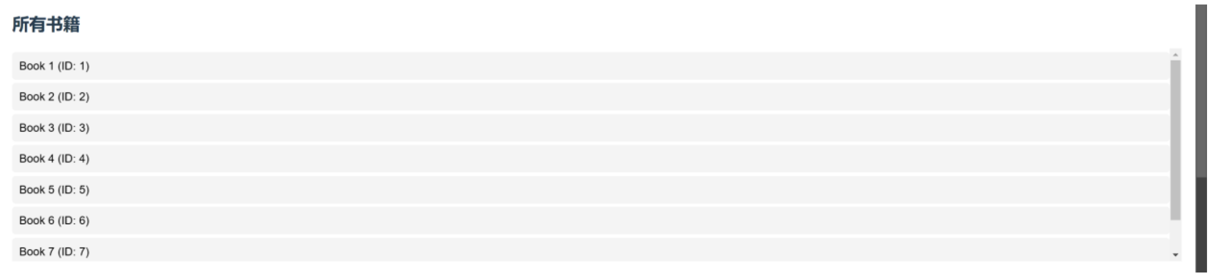
st

图书ID	书名	作者	出版社	出版年份	ISBN	类别	总册数	可借册数	操作
9	test	Nobody	Publisher A	2021	12312	Fiction	20	13	编辑 删除

III. Complex search queries

a) Single-table queries

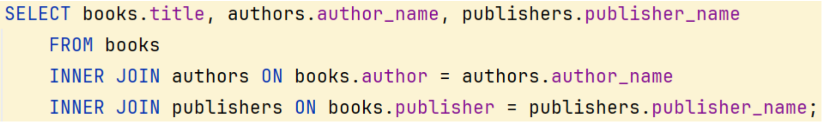
```
SELECT * FROM books
```



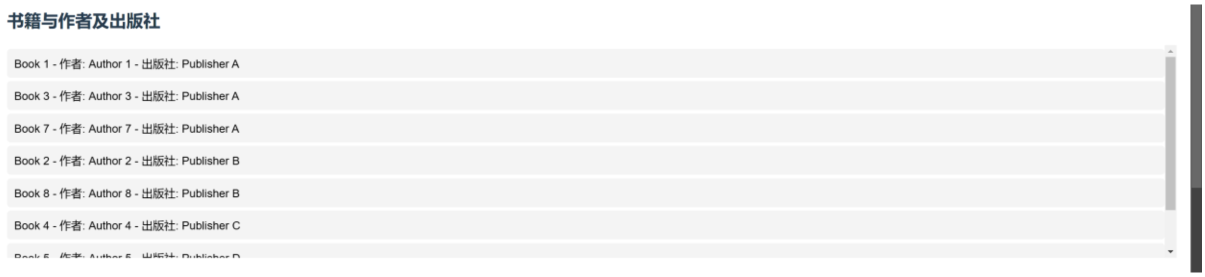
所有书籍

Book 1 (ID: 1)
Book 2 (ID: 2)
Book 3 (ID: 3)
Book 4 (ID: 4)
Book 5 (ID: 5)
Book 6 (ID: 6)
Book 7 (ID: 7)

b) Join queries (inner, outer, self)



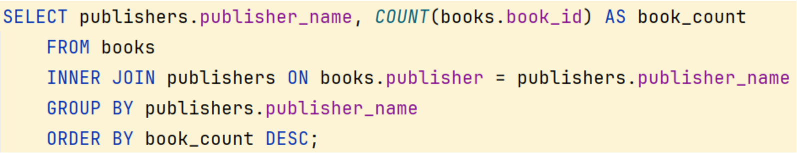
```
SELECT books.title, authors.author_name, publishers.publisher_name
    FROM books
    INNER JOIN authors ON books.author = authors.author_name
    INNER JOIN publishers ON books.publisher = publishers.publisher_name;
```



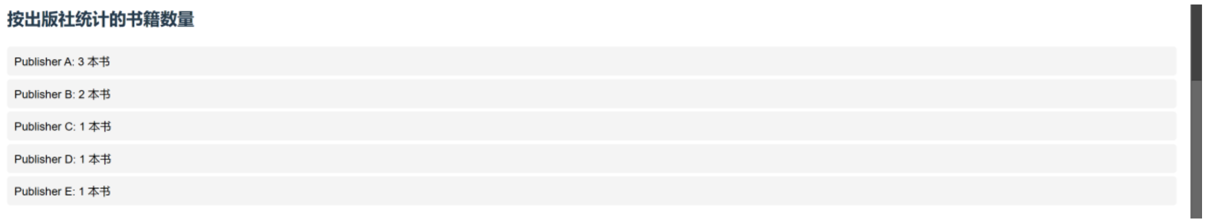
书籍与作者及出版社

```
Book 1 - 作者: Author 1 - 出版社: Publisher A
Book 3 - 作者: Author 3 - 出版社: Publisher A
Book 7 - 作者: Author 7 - 出版社: Publisher A
Book 2 - 作者: Author 2 - 出版社: Publisher B
Book 8 - 作者: Author 8 - 出版社: Publisher B
Book 4 - 作者: Author 4 - 出版社: Publisher C
Book 5 - 作者: Author 5 - 出版社: Publisher D
```

c) Aggregate functions with GROUP BY and ORDER BY clauses



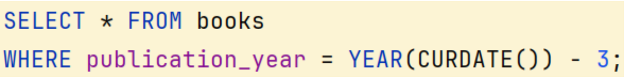
```
SELECT publishers.publisher_name, COUNT(books.book_id) AS book_count
    FROM books
    INNER JOIN publishers ON books.publisher = publishers.publisher_name
    GROUP BY publishers.publisher_name
    ORDER BY book_count DESC;
```



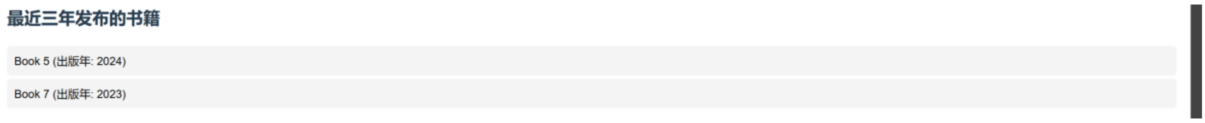
按出版社统计的书籍数量

```
Publisher A: 3 本书
Publisher B: 2 本书
Publisher C: 1 本书
Publisher D: 1 本书
Publisher E: 1 本书
```

d) Date and time functions



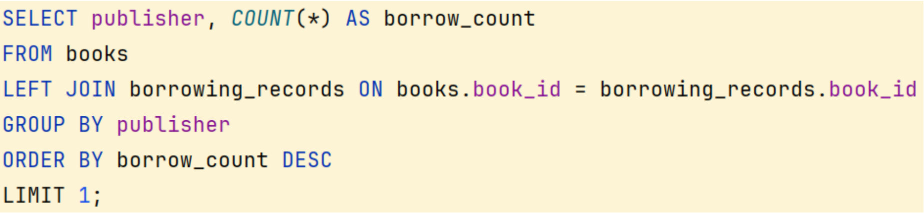
```
SELECT * FROM books
WHERE publication_year = YEAR(CURDATE()) - 3;
```



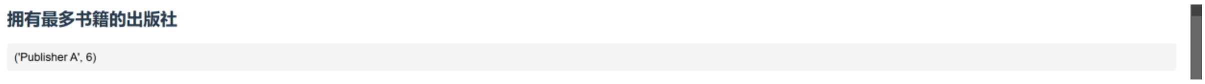
最近三年发布的书籍

```
Book 5 (出版年: 2024)
Book 7 (出版年: 2023)
```

e) Subqueries



```
SELECT publisher, COUNT(*) AS borrow_count
FROM books
LEFT JOIN borrowing_records ON books.book_id = borrowing_records.book_id
GROUP BY publisher
ORDER BY borrow_count DESC
LIMIT 1;
```

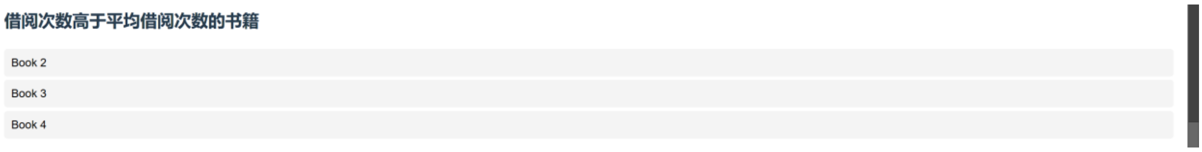


拥有最多书籍的出版社

```
('Publisher A', 6)
```

f) Correlated subqueries

```sql
SELECT title
FROM books
WHERE book_id IN (
    SELECT book_id
    FROM borrowing_records
    GROUP BY book_id
    HAVING COUNT(*) > (
        SELECT AVG(borrow_count)
        FROM (
            SELECT book_id, COUNT(*) AS borrow_count
            FROM borrowing_records
            GROUP BY book_id
        ) AS borrow_stats
    )
);
```



借阅次数高于平均借阅次数的书籍

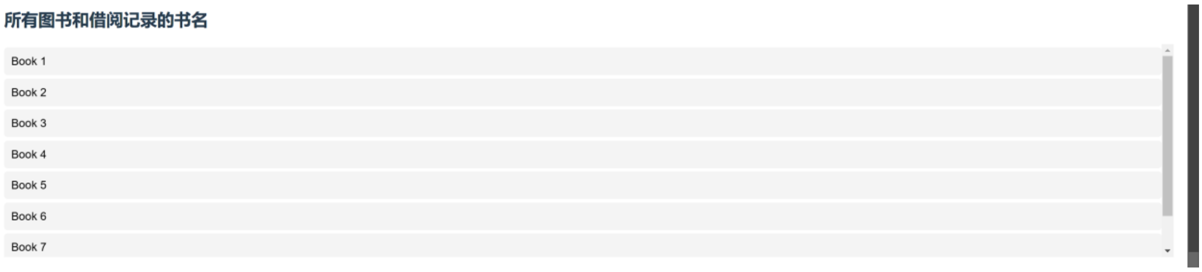
Book 2

Book 3

Book 4

g) Set operations (UNION, INTERSECT, EXCEPT)

```sql
SELECT title FROM books
UNION
SELECT b.title FROM borrowing_records br
JOIN books b ON br.book_id = b.book_id
```



所有图书和借阅记录的书名

Book 1

Book 2

Book 3

Book 4

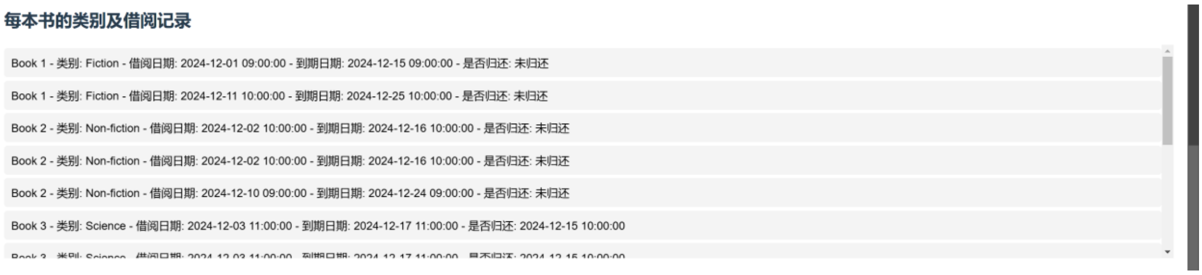
Book 5

Book 6

Book 7

h) Multi-table join queries

```sql
SELECT books.title, book_categories.category_name, COUNT(borrowing_records.book_id) AS borrow_count
FROM books
INNER JOIN book_categories ON books.category = book_categories.category_id
LEFT JOIN borrowing_records ON books.book_id = borrowing_records.book_id
GROUP BY books.book_id, book_categories.category_name
ORDER BY borrow_count DESC;
```



每本书的类别及借阅记录

Book 1 - 类别: Fiction - 借阅日期: 2024-12-01 09:00:00 - 到期日期: 2024-12-15 09:00:00 - 是否归还: 未归还

Book 1 - 类别: Fiction - 借阅日期: 2024-12-11 10:00:00 - 到期日期: 2024-12-25 10:00:00 - 是否归还: 未归还

Book 2 - 类别: Non-fiction - 借阅日期: 2024-12-02 10:00:00 - 到期日期: 2024-12-16 10:00:00 - 是否归还: 未归还

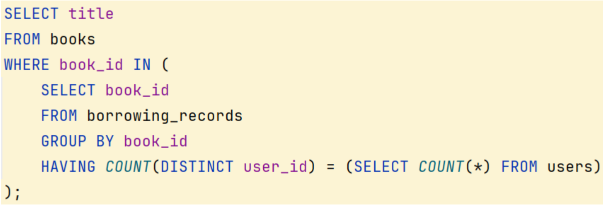
Book 2 - 类别: Non-fiction - 借阅日期: 2024-12-02 10:00:00 - 到期日期: 2024-12-16 10:00:00 - 是否归还: 未归还

Book 2 - 类别: Non-fiction - 借阅日期: 2024-12-10 09:00:00 - 到期日期: 2024-12-24 09:00:00 - 是否归还: 未归还

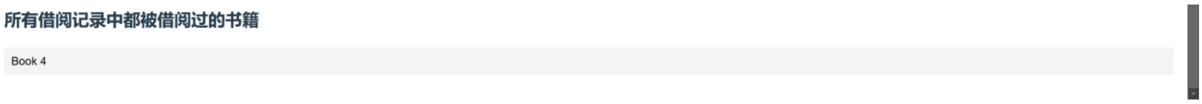
Book 3 - 类别: Science - 借阅日期: 2024-12-03 11:00:00 - 到期日期: 2024-12-17 11:00:00 - 是否归还: 2024-12-15 10:00:00

Book 3 - 类别: Science - 借阅日期: 2024-12-03 11:00:00 - 到期日期: 2024-12-17 11:00:00 - 是否归还: 2024-12-15 10:00:00

i) Division queries



```
SELECT title
FROM books
WHERE book_id IN (
    SELECT book_id
    FROM borrowing_records
    GROUP BY book_id
    HAVING COUNT(DISTINCT user_id) = (SELECT COUNT(*) FROM users)
);
```



所有借阅记录中都被借阅过的书籍

Book 4

IV. Technology used

The front and back ends and databases used in our system are as follows：

1. Front-end: HTML and CSS
2. Back-end: Python
3. Database: MySQL

二、诚信承诺：

本人郑重承诺在该课程论文完成的过程中不发生任何不诚信现象，一切不诚信所导致的后果均由本人承担。

Integrity Commitment:

I/We solemnly promise that I/we have not plagiarized during the completion of this course paper, and all the consequences caused by plagiarism will be borne by me/us.

签名拍照附件:

Name and Digital Signature of all students in the group

Date: December 26, 2024