

CS5200 - Database Management Systems

Project Final Report

MyBooks Library Management Application

Group Name:

LinZZhouWParkJ

Group Members:

Wanning Zhou

Ziqing lin

Jin Park

Introduction

The primary objective of the “Library Books Management Application” is to develop a structured relational database tailored for the efficient management of data concerning books, authors, nationalities of authors, publishers, genres of books, languages of books, formats of books, users, and their respective book ratings. Designed to support administrative tasks within a library, this system provides a comprehensive toolset for the seamless organization, cataloging, and oversight of library resources.

While users can search for books, learn about authors, publishers, languages and submit ratings for books, the system's primary focus is to cater to the administrative needs of libraries. From inventory management to analytics about book circulation and ratings, it's built with the librarian or administrator in mind.

Future developments, depending on resources and feedback, might see the addition of features like a recommendation engine for book procurement, more in-depth analytics for understanding borrowing patterns, and perhaps an interface for users to suggest books or provide feedback on existing resources.

Technical description

The primary language used to develop the application is Python3. Mysql database will be used for efficient data query. Since the database is well-structured, it is more efficient to use a structured database like Mysql. For the Graphical interface, the tkinter library will be used for GUI components and the tkinter.messagebox will be used to display message boxes to the user. The pymysql will be used to connect the python application to the Mysql database for CRUD operations.

Database Description

The "Library Books Management Application" database is meticulously structured to manage detailed information about books, authors, nationalities of authors, publishers, genres of books, languages of books, formats of books, users, and the ratings they provide to the books. It is primed to offer a reliable framework to library administrators, ensuring that the intricacies of book management are handled efficiently. There are 8 strong entities in the database:

Book forms the nucleus of our database. Every book entry holds a `book_id` as a unique identifier, an ISBN-10 number, a book title, the edition of this book, number of pages, its first published date and an average rating based on users' ratings. The comprehensive nature of the database allows for every book to be linked to its publisher, author, language, format, and the genre it belongs to. In addition, every book can be rated by different users.

Publisher each has a unique name, allowing administrators to have clarity about the origin and background of the books they house. A published date is also tracked when a publisher published a specific book.

Author is identified by a unique `author_id` and come with their respective names, their born year, died year (if applicable) and a short description. This ensures that books written by the same author can be easily clustered together or searched for.

Genre plays a vital role in classification. Each genre is tagged with a unique type name and a short description, helping categorize books and streamline searches for particular thematic content.

Language has a unique language name and a short description, which helps track the language each book was written in.

Format is identified by a unique format name, and comes with a short description. This helps track the binding format of each book.

Nationality has a unique nationality name, representing the nation that each user belongs to. Each user can belong to up to 2 nationalities.

User mainly comprises library members or administrators. They have a unique ID and a username. This assists in tracking the books rating values provided by the user.

4 relationship tables are also implemented:

Publish represents a publisher published a book in a specific published date. One publisher can publish 0 to many books, while one specific book (given the edition and format) can only be published by one publisher.

Author_nationality represents one author belongs to up to two nationalities, while one nationality can be related to 0 to many authors.

Author_write_book represents the relationship between authors and the books they have written. One author can write 0 to many books, while one book can be written by 1 to many authors.

User_rate_book tracks the book rating values that users posted for the books. One user is allowed to rate 0 to many books, while each book can be rated by 0 to many users. However,

after users posted a rating value to a book, posting a new rating value for the same book is not allowed, while deleting and updating the existed rating values are allowed.

Datasets

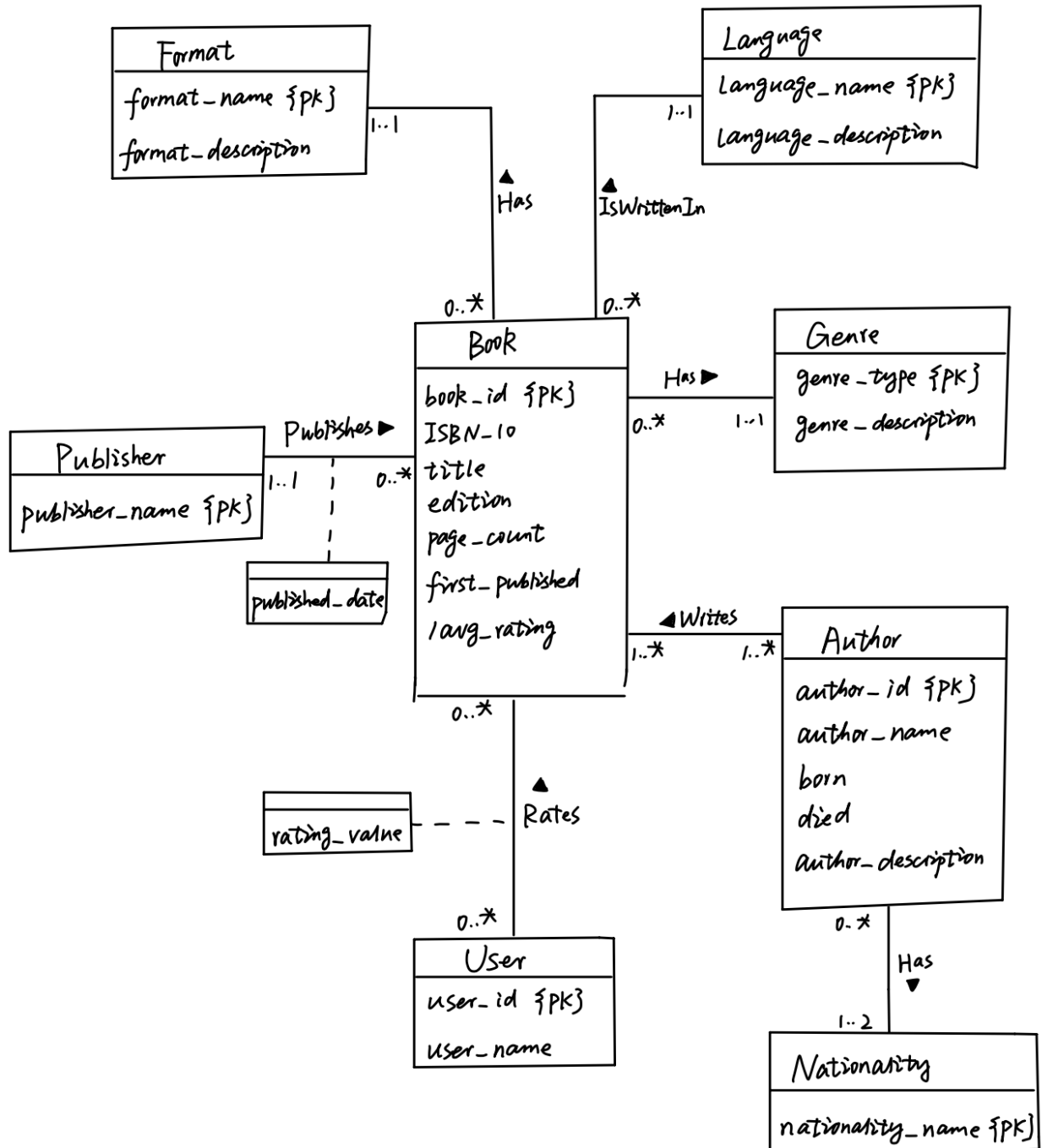
The datasets for this database are collected manually by ourselves. The information for each book in Book table and genre description in Genre table are collected from website *goodreads*: <https://www.goodreads.com/> . The information for each author in Author table as well as description in Language and Format tables are collected from *Wikipedia*.

Since the collected data are not in third normal form, we also applied Python pandas library using Jupyter Notebook to manipulate the data and create the relationship tables. A HTML file 'data_cleaning.html' has been attached to the submission folder for this part.

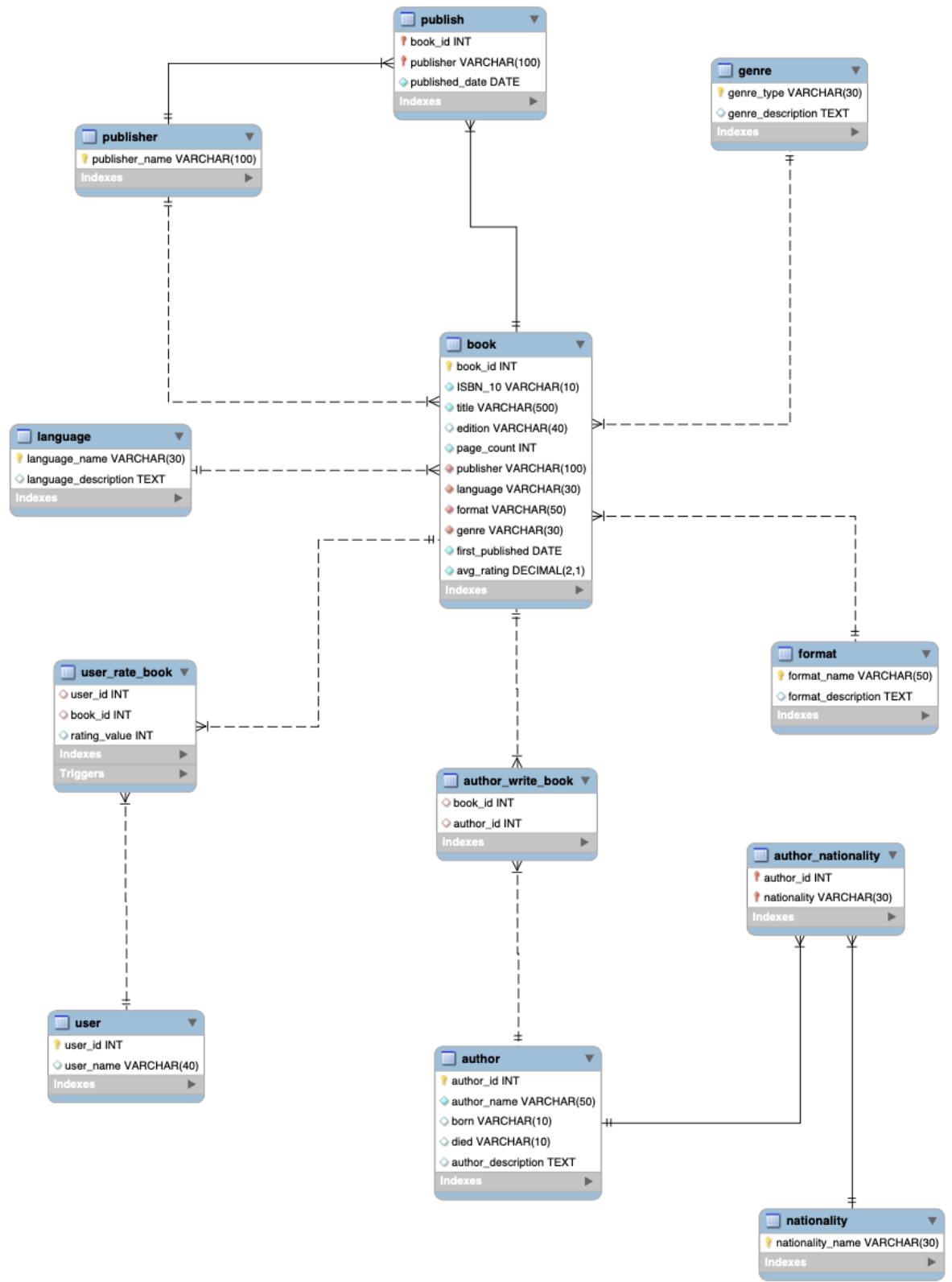
Video Presentation

The video presentation was uploaded to Google Drive, please follow the link to watch the video: https://drive.google.com/file/d/1TJCPeuOI_eLMO3NaxJlf8wiD6W70dvWn/view?usp=sharing

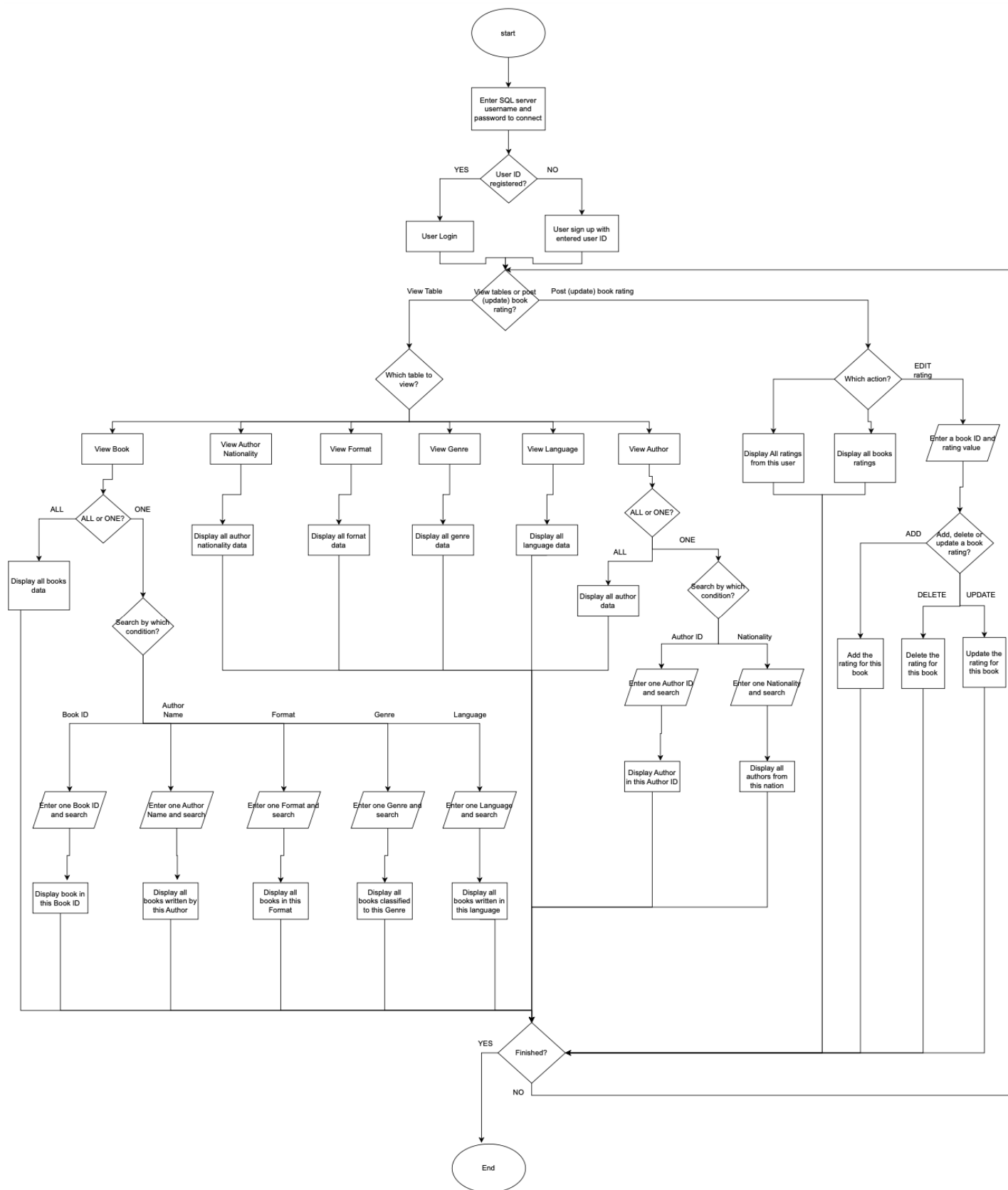
UML Diagram:



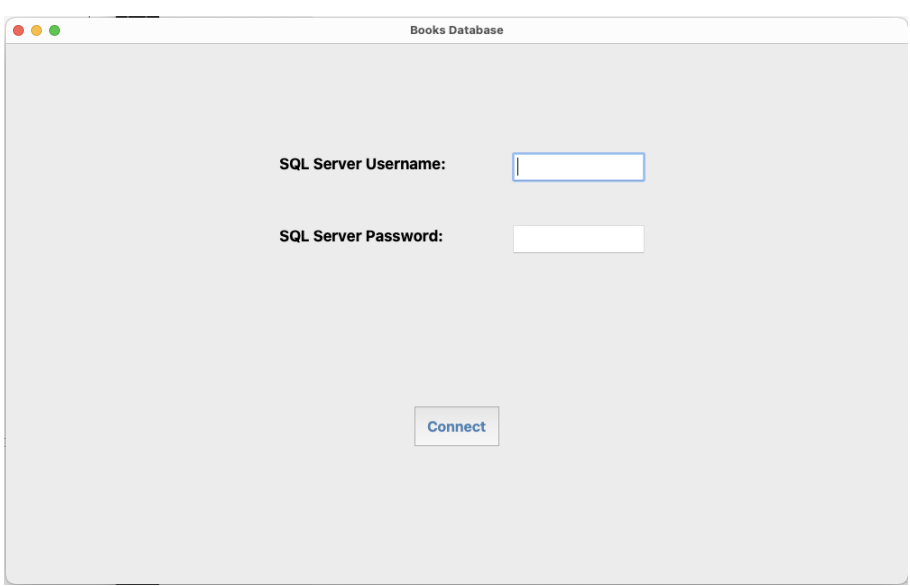
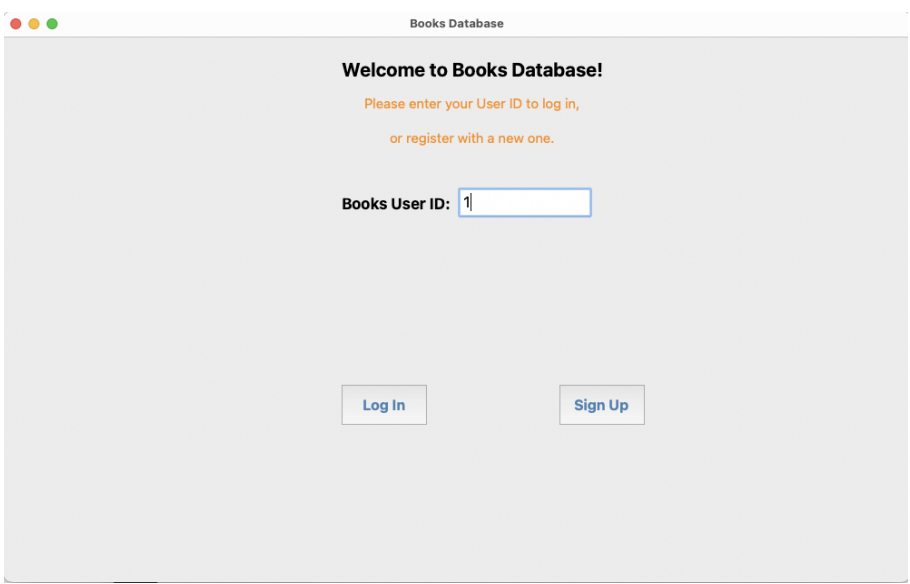
Reverse Engineer EER Diagram:



Activity diagram



Application User Flow Example

Connect to db	
Enter user ID	

Options 1:
Select Query on
(Book, Author,
Format, Genre,
Language, Author
Nationality)

Options 2:
Post or Update
Book Ratings

Books Database

Welcome to Books Database!

Check tables or leave your rating to the books!

Check One Table:

Post or Update Your Book Ratings:

Selected Options 1
Author query example

- Perform query by Author ID
- Perform query by Author ID
- Check All Authors

Books Database

←

Welcome to Author Table!

Check By Author ID:

Check By Nationality:

author_id	author_name	nationality	born	died	author_description
0	Mark P.O. Morford	Sri Lankan	1929	2019	He is best remembered fo
1	Robert J. Lenardon	Canadian	1928		A Canadian classics educ
2	Richard B. Wright	Canadian	1837	2017	He was a Canadian noveli
3	Carlo D'Este	American	1936	2020	He was an American milita
4	Gina Kolata	American	1948		She is an American scienc
5	Elizabeth Wayland Barber	American	1940		She is an American schola
6	Amy Tan	American	1952		She is an American author
7	Robert Cowley	American	1934		He is an American military
8	Paul Genney	English			
9	David Cordingly	English	1938		He is an English naval hist

Selected Options 2

- Check All My Ratings
- Check All Books Ratings
- Add/Update/Delete ratings

The screenshot shows a web application titled "Books Database". At the top, there are buttons for "Check All My Ratings" and "Check All Books Ratings". A welcome message says "Welcome to Books Rating!" with a "My User Id" input field. Below this is a table of books with columns: book_id, ISBN_10, book_title, publisher, language, genre, avg_rating, and your_rating. The table contains five rows of data. At the bottom, there is a form to add, update, or delete ratings, with fields for "Book Id" and "Rating Value(integer 0-5)", and a "Submit" button.

book_id	ISBN_10	book_title	publisher	language	genre	avg_rating	your_rating
1	0060506075	Clara Callan: A Nov	Harper Perennial	English	Fiction	4.5	4
2	0141017619	Decision in Norman	Penguin Books	English	History	4.0	3
5	0143038109	The Kitchen God's	Penguin Books	English	Fiction	3.0	3
6	0425176428	What If?: The Work	Berkley	English	History	4.0	4
222	0385730586	Sisterhood of the T	Ember	English	Fiction	1.5	2

Book Id	Rating Value(integer 0-5)	Action
0	0	<div><div>Add</div><div>✓ Add</div><div>Delete</div><div>Update</div></div>

Lesson Learned

During the development of our library search system, our team gained valuable experience not only in the technical field, but also in time management.

Technical Expertise: We learned how to design and manage a relational database using MySQL from scratch. We gained proficiency in database design, schema creation, and SQL query optimization. We also realized a well-defined database is useful to guide our implementation and can also lead to efficient data retrieval and management. In addition, from a technical point, we also gained valuable experience in GUI and event handling from implementing a graphical user interface (GUI) with Python's tkinter library.

Time Management: We learned the importance of effective time management and setting realistic project timelines and allocating additional time for testing and debugging. Wanning Zhou has been a great team leader in making sure we follow the proposed timeline, and is proactive in pushing the project forward.

Through this experience, we have not only obtained the technical skills to build a library system from scratch using python and relational databases, but also acquired valuable insights and skills that will continue to benefit us in the future.

Future work

In our existing design, the intended users are registered users whose login credentials are stored in the database, i.e. non registered users do not have access to ratings or other library database features. This decision was made with security considerations in mind. However, for future work, we are planning to allow non-registered users to access restricted library features like data

query and view only access to ratings. To accomplish this transition, we will make adjustments across various aspects of our system, including the user interface, database, and the codebase:

User Interface:

To facilitate non-registered user's interaction with our library system, we are planning to create a new user interface specifically designed for those who have not yet registered.

This interface is accessible to any user.

Database Access Control:

The database will undergo an expansion from single user type to two distinct user categories: guests and registered users. For guest users, read-only privileges will be granted. Guests can read and perform queries on library databases but are restricted from write (leave ratings). On the other hand, registered users will have both read and write access, allowing them to not only view but also contribute to the library's ratings and content.

Codebase:

We will introduce a new "guest" role within our system so that non-registered users can access the library system. The guest role has limited permissions, primarily read-only access to our library resources and ratings. It is important to note that guest users will not be granted write access, which includes the ability to leave ratings or contribute in other ways.

README:

To run this application, user would need Python 3, pymysql library and tk library installed on their computer. User would also need to import the database schema dump file 'books_dump.sql' to their locally hosted MySQL server on their computer. The application is run through a command line argument 'python booksApp.py'. The application window will then prompt the user to input their username and password to connect to their local host MySQL server. After this the program will begin and user will be instructed on exploring the application by clicking different buttons and typing inputs.