|  |  |
| --- | --- |
| Library Management System | sUBMITTED BY:  Zainab bilal (2022635)  rukh-e-zahra (2022508)  MARIAM (2022282)    SUBMITTED TO: MISS ABINTA MEHMOOD |

Contents

[**LIBRARY MANAGEMNENT SYSTEM** 1](#_Toc135444845)

LIST OF CONTENT…………………………………………………………………………...................................................... 2

[1. Introduction: 3](#_Toc135444848)

[2. OVERVIEW OF DATABASE 3](#_Toc135444849)

3. BUSINESS SCENARIO:……………………………………………………………………………………………………………………..4

[4. ENTITY RELATION DESIGN: 4](#_Toc135444851)

5. RELATIONSHIP SCHEMA…………………………………………………………………………………………………………………5

6.NORMALIZATION:…………………………………………………………………………………………………………………………..5

7. NORMALIZATION:………………………………………………………………………………………………………………………….6

8. DATA POPULATION:……………………………………………………………………………………………………………………..6

9. DATA POPULATION……………………………………………………………………………………………………………………….7

10.INDEXATION IN THE DATABASE……………………………………………………………………………………………………7

11.EXECUTIVE SUMMARY…………………………………………………………………………………………………………………7

12 CONCLUSION:………………………………………………………………………………………………………………………………8

**Introduction:**

Introduction to the Library Management System Database Setup:  
  
This SQL script creates a comprehensive database for a library management system. The database's design includes tables for publishers, books, library branches, borrowers, librarians, members, book requests, and more. By designing the database in this manner, we provide the groundwork for effectively managing the library's resources, recording borrowing activities, processing membership information, and facilitating administrative chores. This arrangement allows for the smooth organisation and retrieval of library data, thus improving the overall efficiency and efficacy of the library management system.

**Overview of the Database:**  
  
To create a solid database structure for a library management system, this SQL script was created. It includes various essential elements required for the efficient management of the library and the provision of services to users. An outline of the features and parts is provided below:  
  
  
1. Tables Creation: Tables are made for all the important entities: publishers, books, branches of libraries, borrowers, librarians, members, authors of books, book issues, requests for books that are still waiting, and registrations that are not yet complete.   
  
2. Data Insertion: To provide a starting point for testing and development, sample data is inserted into each table to fill it with records.   
  
3 . Data Integrity Enforcement: Rules about data integrity are enforced through the use of triggers. Examples of these triggers include updating book copies upon the issue and verifying member balances before approving book requests.

4. Authentication and Authorization: Authentication procedures are in place for both librarians and members, ensuring secure access and protecting sensitive information.  
  
5. Search Functionality: Users may easily find books by title, author, category, or ISBN, improving their experience and accessibility.  
  
6. Data Modification: Administrators can use update queries to keep book information accurate and up-to-date.  
  
7. Pending Requests Handling: Tables for pending book requests and registrations help manage user actions and streamline administrative work.  
  
8. Overall Functionality: This script's database structure and functionalities offer a comprehensive foundation for managing library resources, recording borrowing operations, handling user interactions, and maintaining data integrity.

By implementing this database setup, library administrators can streamline their operations, improve user experience, and effectively manage library resources to better serve their patrons.

Top of Form

**Business Scenario: Library Operations Optimization.**

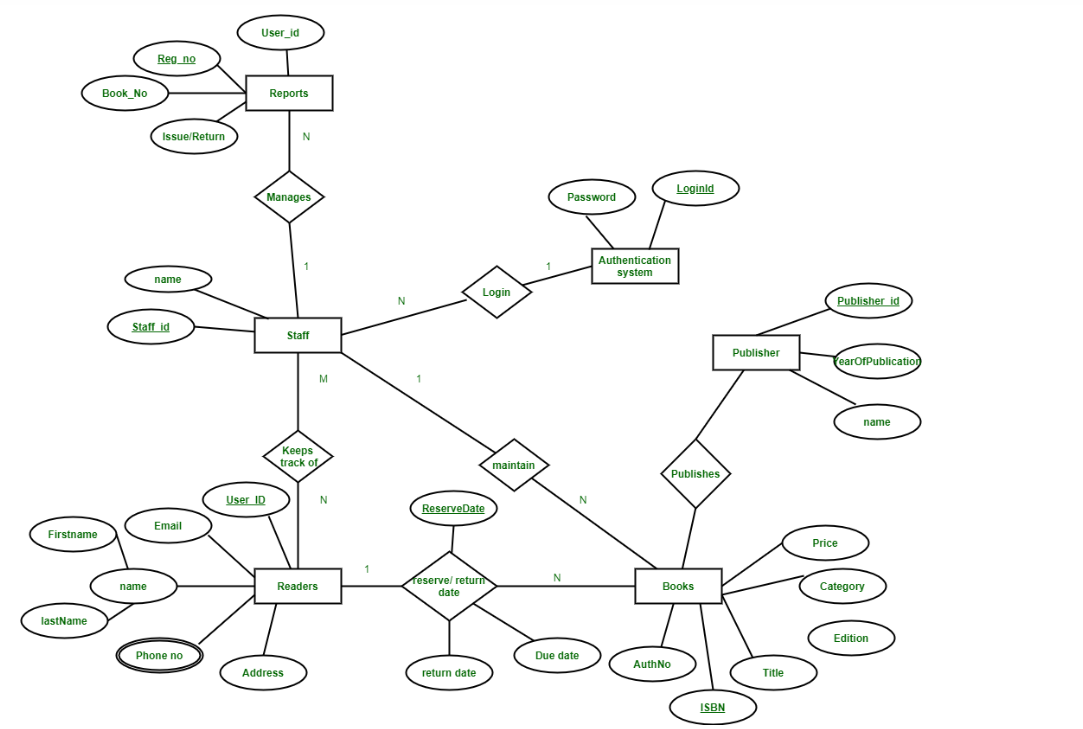
In a typical library, managing resources and serving patrons efficiently are critical. This database system helps with these efforts by efficiently arranging data. It allows librarians to organise books methodically by title, author, and category. With automated book borrowing and return systems, librarians can handle transactions quickly, reducing patron queues and wait times.

The system's basic search functionality enables customers to easily find books based on titles, authors, or categories, thereby improving their experience. Furthermore, triggers built into the system maintain data integrity, ensuring that rules like checking member balances before accepting book requests are constantly followed.

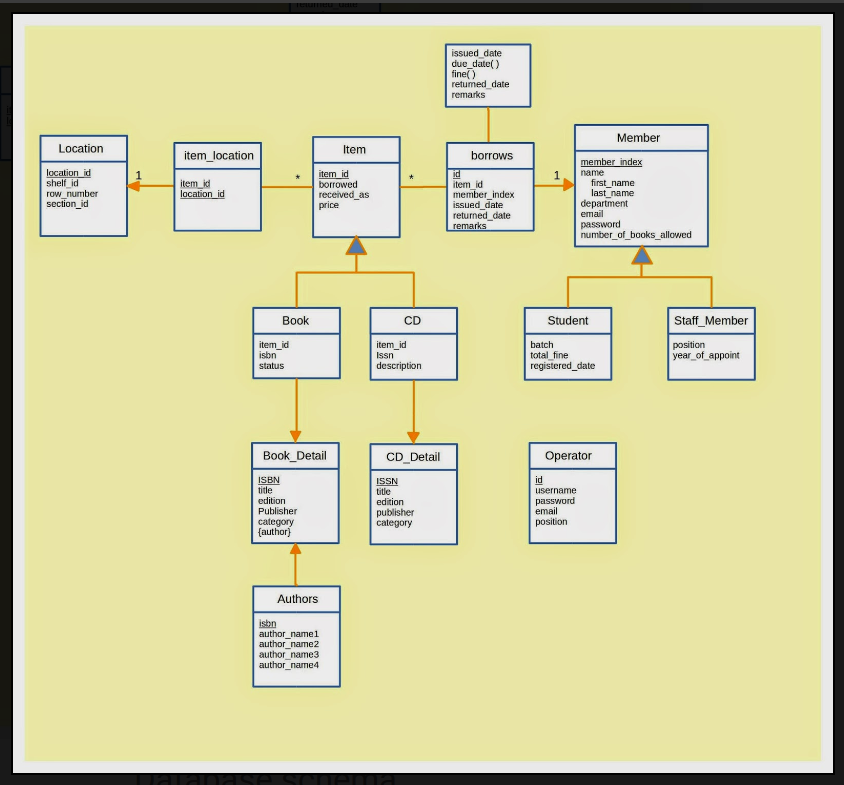
Administratively, the system centralises responsibilities such as member registration and book requests, making management easier for librarians.

.

**Entity Relationship Diagram (ERD) Design**:



**Relational Database Schema:**



**Normalization:**

Normalization is a crucial database design technique that minimizes redundancy and dependency by organizing tables into well-structured relations. It involves decomposing tables into smaller, related entities to ensure data integrity and reduce anomalies. Let's analyze the database schema in terms of normalization:

**1. First Normal Form (1NF):**

- Each table in the database schema appears to satisfy the requirements of 1NF, as each attribute contains only atomic values, and there are no repeating groups.

**2. Second Normal Form (2NF):**

- A table is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependent on the primary key.

- In the provided schema, the tables seem to be in 2NF as there are no partial dependencies; all attributes depend on the whole primary key.

**3. Third Normal Form (3NF):**

- A table is in 3NF if it is in 2NF and all non-prime attributes are non-transitively dependent on the primary key.

- In the schema:

- `table\_book` and `table\_publisher`: There are no transitive dependencies, so they are in 3NF.

- `table\_book\_authors`: AuthorName is functionally dependent on AuthorID, and AuthorID is functionally dependent on BookID, thus satisfying 3NF.

- `book`: ISBN, Title, Author, Category, Price, and Copies are directly dependent on the primary key (ISBN), and there are no transitive dependencies, so it's in 3NF.

- `book\_issue`: Member, Book\_ISBN, Due\_Date are directly dependent on the primary key (issue\_id), so it's in 3NF.

- `librarian` and `member`: All non-prime attributes are directly dependent on the primary key, so they are in 3NF.

- `pending\_book\_requests` and `pending\_registrations`: All non-prime attributes are directly dependent on the primary key, so they are also in 3NF.

**DATA POPULATION:**

To populate the database with example data consistent with the specified code and business scenario, we must enter records into each table. This is how we do it.   
  
  
1. Publishers (table\_publisher): - Add records with publishers' names, addresses, and phone numbers.   
  
  
2. Books (table\_book):- Provide sample data for books, including IDs, titles, and publishers.   
  
3. Library Branches (table\_library\_branch): - Add information about library branches, including IDs, names, and addresses.   
  
4. Borrowers (table\_borrower): - Add records for borrowers, such as card numbers, names, addresses, and phone numbers.   
  
5. Book Authors (table\_book\_authors): - Add entries for book authors, tying them to books using their IDs and including their names.   
..   
  
6. Book Issues (book\_issue): - Enter records with issue IDs, borrower names, book ISBNs, and due dates to indicate occurrences of book borrowing.   
  
7.Librarians (librarian): - Include information on librarians, such as usernames, hashed passwords for authentication, and IDs.   
  
8. Members (participant): - Enter member details in this table, such as IDs, usernames, passwords, names, email addresses, and account amounts.   
  
9. Pending Book Requests (pending\_book\_requests): - Add records with the request IDs, member names, and book ISBNs to indicate pending book requests.   
  
10. Pending Registrations (pending\_registrations): - Provide sample information, such as usernames, passwords, names, email addresses, and initial balances, for pending member registrations.

**INDEXATION IN THE DATABASE:**

Indexation in this code involves creating indexes on specific columns of the database tables to enhance query performance. Indexes enable faster data retrieval by providing a quick lookup mechanism. They are created on columns frequently used in search conditions, such as ISBN, username, title, and others, to optimize query execution time. This optimization aims to improve overall database efficiency and enhance the user experience by reducing query response times.

**Executive Summary: Library Management System Database Schema and Triggers:**

The Library Management System's database schema and triggers efficiently manage various library operations, including book management and issuance. The system consists of multiple tables, organized by key entities like publishers, books, borrowers, and librarians. Triggers enforce business rules and maintain data consistency, such as preventing unauthorized requests and updating inventory. The system's robust foundation ensures efficient data management, integrity, and security, ensuring optimal performance and functionality in a modern library environment.

**CONCLUSION:**

In conclusion, the provided code snippet demonstrates the creation of a robust database schema for a Library Management System. It encompasses essential entities such as publishers, books, borrowers, library branches, and more, facilitating efficient data organization and management. The schema ensures data integrity through primary and foreign key constraints while optimizing query performance with the creation of indexes on key columns. Triggers are employed to enforce business rules and maintain data consistency, ensuring that operations such as book requests and issuances are processed securely. Overall, this database schema forms a solid foundation for a functional and reliable Library Management System, capable of handling the complexities of library operations effectively.