**B.Tech Program Second Year**

**Course: Relational Database**

**Management System Lab**

**Course Code: CC2231**

**LIBRARY DATABASE MANAGEMENT SYSTEM**

by

**Akash Aggarwal**

(Reg. No: 219303172 )

Submitted to

**Mr. Monu Bhagat Assistant Professor**

Department of CCE, SCIT, Manipal University Jaipur, Jaipur (Raj.).

**Department of CCE**

**School of Computing and Information Technology Faculty of Engineering**

**Manipal University Jaipur, India**

April, 2023

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | TITLE | DATE | SIGNATURE |
| 1 | Introduction |  |  |
| 2 | Aim and Objective |  |  |
| 3 | Technical Details |  |  |
| 4 | ER Diagram |  |  |
| 5 | Functional Dependencies |  |  |
| 6 | Normalization |  |  |
| 7 | Front End |  |  |
| 8 | Future Prospects |  |  |
| 9 | Bibliography and References |  |  |

***INTRODUCTION***

Maintaining the records of students, books, accounts & staff manually is a tedious task. There are various other problems too, faced by the students in a library, such as finding any book, information about a particular books’ availability, searching of books using ISBN number, etc. To eliminate this manual system, this library database management system has been developed. It will handle all the current issues faced by the students and by its admin personnel.

To store all the information in the database from where user will place their query and get the results based on their query. Only valid users will be able to access this Library Database Management System. Through this Library Management System, it will be easy to manage accounts and various details of student and employees working under library along with the records of the books.

***AIM AND OBJECTIVE***

The plan is to implement the following features:

* Searching of books
* Issuing and returning books
* Paying fine (if any) online
* Librarian can read information about any member
* Librarian can track the books issued by a particular student

***TECHNICAL DETAILS***

***Software and Languages used:***

* Front End:

1. HTML
2. CSS
3. JavaScript

* Back End:

1. MySQL

***Functional Requirements:***

* The system must only allow user with valid id and password to enter the system.
* The user must be able to logout after they have finished using system.
* System must be able to not allow two books having same book id.
* System must be able to search if a book is available or not before issuing books.
* Librarian (Admin) must be able to see the availability of a particular book & the data/status of each user.

### *Data Requirements:*

* There will be the data of each book in the database.
* There will be the username and password of each member & faculty in the database.
* The record of the issued books will be in the database.
* The record of all the staff in the library will be in the database.
* The record of all the staff in the library will be there in the database.

### *Detailed Data Requirements:*

* Library will be managed by the admin. Each admin will have its unique login id and password. The library will be managing the books, staffs, and members (both students and faculties).
* Each staff will be having its unique id, a name, designation, salary, date\_of\_joining, address and dob.
* Each of the books will have its unique book\_code, a name, a subject\_code, no\_of\_books, rack\_no, cost, date\_of\_purchase and name of the author.
* The members are the ones who would be accessing the library system for issuing and returning books and paying fine when not returned in time. The library consists of two members, Faculties and Students.
* Faculty (member) will be having its unique faculty id, a name, an address, a contact number, and a department. When the faculty issues books, it will contain attributes faculty id and book\_id which in turn will contain issue and return date.
* Student (member) will be having its unique id, a name, an address, a contact number, and a branch. When the student issues books, it will contain attributes id and book\_id which in turn will contain issue and return date.

***Entity types:***

* Book

Strong entity set: book\_code is used to identify each entity uniquely

Book\_code, book\_name, Author, date\_of\_purchase, price, subject\_code, rack\_no, no\_of\_books

* Staff

Strong entity set: staff\_id is used to identify each entity uniquely

staff\_id, staff\_name, dob, address, designation, salary, date\_of\_joining

* Student

Strong entity type: student\_id is used to identify each entity uniquely

student\_id, name, branch, fine, address, phone\_no, issue\_date, expiry\_date

* Faculty

Strong entity type: f\_id is used to identify each entity uniquely

f\_id, name, address, phone\_no, department

* Librarian

Strong entity type: admin\_login is used to identify each entity uniquely admin\_login, admin\_password

* Author

Strong entity type: author\_id is used to identify each entity uniquely

author\_id, author\_name, dob, address, experience

* Periodicals

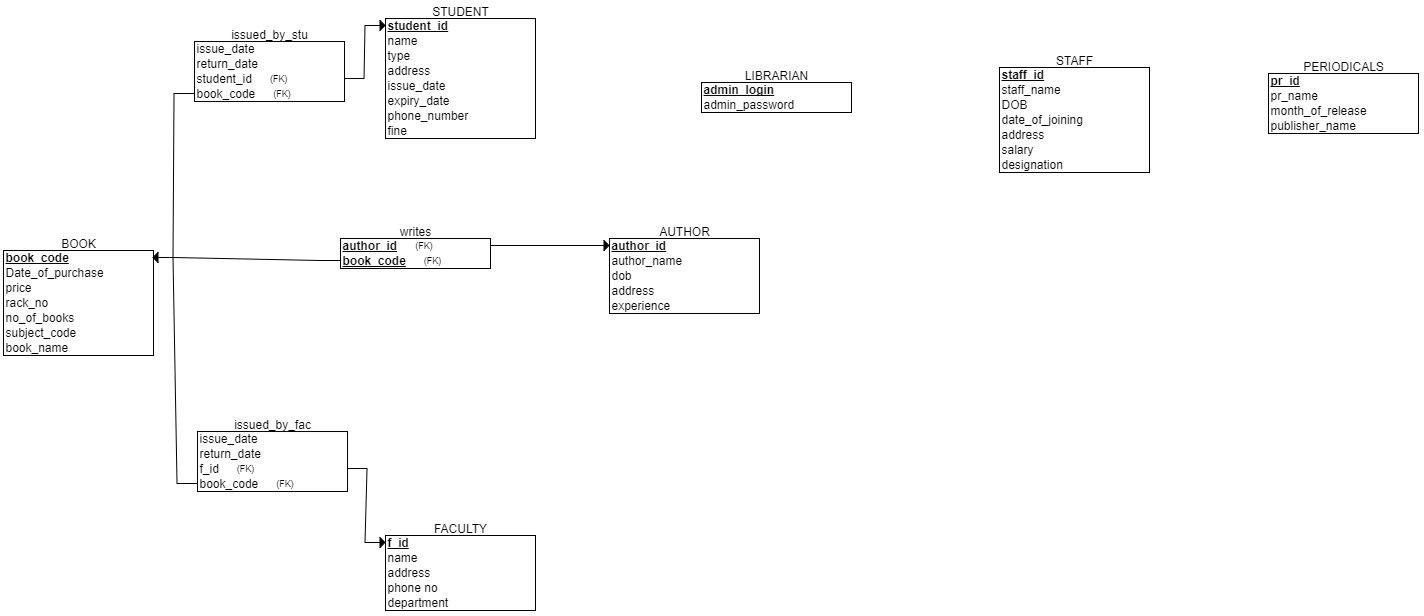
Strong entity type: pr\_id is used to identify each entity uniquely

pr\_id, pr\_name, month\_of\_release, publisher\_name

***ER DIAGRAM***

### 

***RELATIONAL SCHEMA***



### Functional Dependencies:

* {book\_code} → {date\_of\_purchase, book\_name, price, rack\_no, no\_of\_books, subject\_code}
* {staff\_id} → {staff\_name, DOB, date\_of\_joining, address, salary, designation}
* {student\_id} → {name, type, address, issue\_date, expiry\_date, phone\_no, fine}
* {faculty} → {name, address, phone\_no, department}
* {author\_id} → {author\_name, dob, address, experience}
* {book\_code, student\_id} → {issue\_date, return\_date}
* {book\_code, f\_id} → {issue\_date, return\_date}

***NORMALIZATION***

* Student (student\_id, name, type, address, issue\_date, expiry\_date, phone\_number, fine) The is already in First Normal Form, since all the attributes are single valued. All the attributes of the Student table are atomic.

Since it is already in 1NF & there is no chance of partial dependency on the key attribute because it has only one key attribute, the table is in Second Normal Form.

Closure of the attribute set:

F = {student\_id → name, student\_id → type, student\_id → address, student\_id → fine, student\_id→ phone\_number, student\_id → issue\_date, student\_id → expiry\_date} Student\_id+ = {name, type, address, fine, phone\_number, issue\_date, expiry\_date}

A transitive dependency can occur only in a relation that has three or more attributes. Since there is no transitive dependency, the table is in Third Normal Form.

Since there is only one key attribute & it is not dependent on the non key attribute(s), we can say that the above table is in BCNF.

* Faculty (f\_id, name, phone\_no, department, address)

The table is in BCNF, as f\_id is the only candidate key, and all others are non-key attributes, thus table is in 2NF. There are no non-key dependencies, hence it is in 3NF.

* Book (book\_code, book\_name, subject\_code, no\_of books, rack\_no, price, date\_of\_purchase)

The table is in BCNF, as book\_code is the only candidate

key, and all others are non-key attributes, thus table is in 2NF. There are no non-key dependencies, hence it is in 3NF.

* Staff (staff\_id, staff\_name, staff\_name, address, date\_of\_joining, salary, designation) The table is in BCNF, as staff\_id is the only candidate key, and all others are non-key attributes, thus table is in 2NF. There are no non-key dependencies, hence it is in 3NF.
* Periodicals (pr\_id, pr\_name, month\_of\_release, publisher\_name)

The table is in BCNF, as pr\_id is the only candidate key, and all others are non-key attributes, thus table is in 2NF. There are no non-key dependencies, hence it is in 3NF.

* Librarian (admin\_login, admin\_password)

The table is in BCNF, as admin\_login is the only candidate key, and all others are non-key attributes, thus table is in 2NF. There are no non-key dependencies, hence it is in 3NF.

* Author (author\_id, author\_name, dob, address, experience)

The table is in BCNF, as author\_id is the only candidate key, and all others are non-key attributes, thus table is in 2NF. There are no non-key dependencies, hence it is in 3NF.

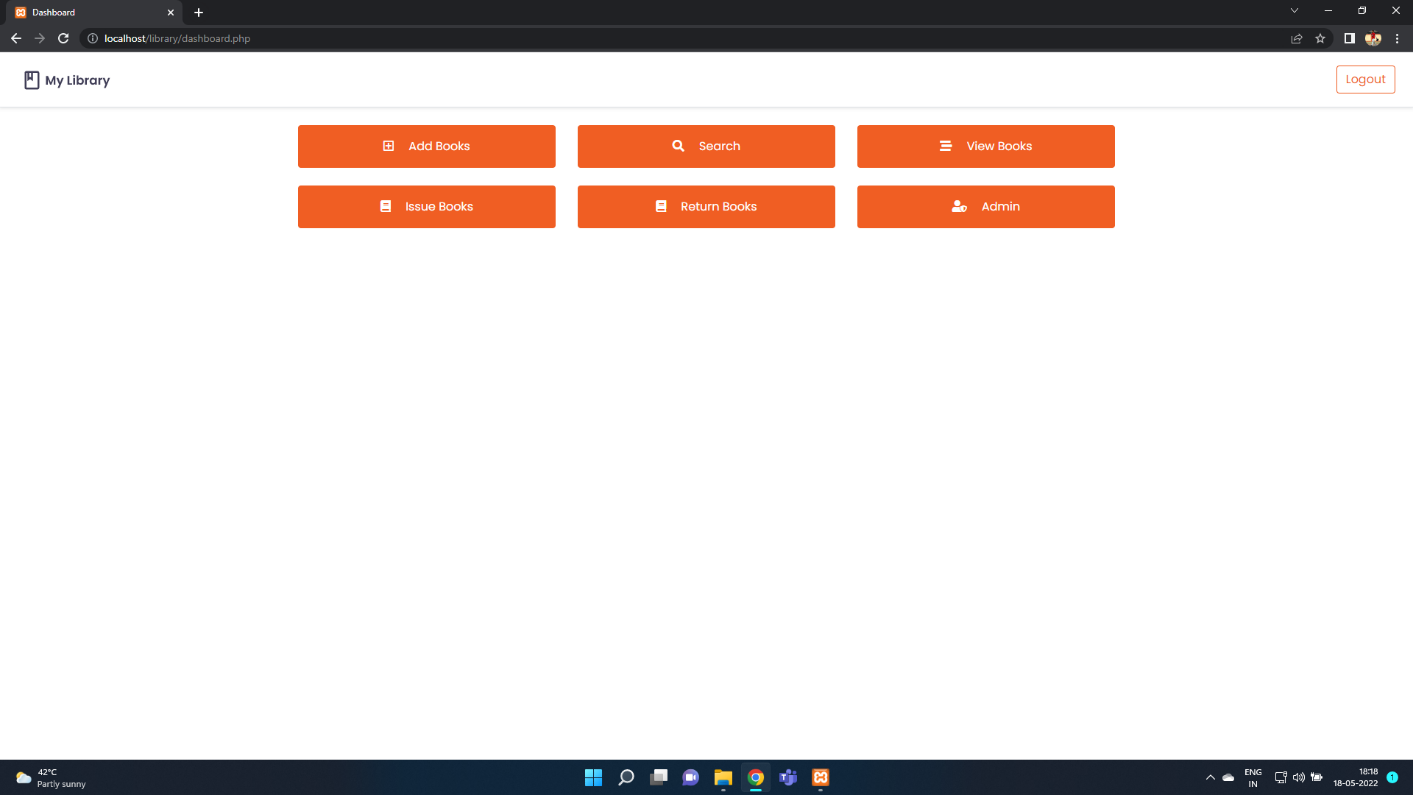
* issued\_by\_stu (student\_id, book\_code, issue\_date, return\_date)

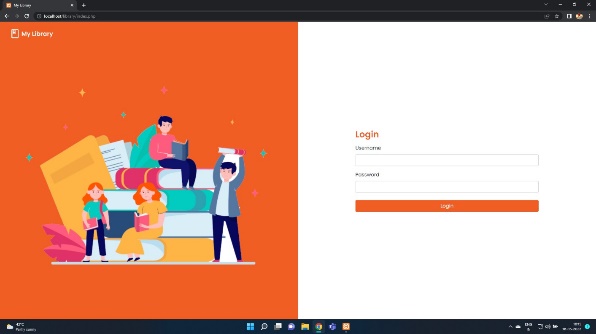
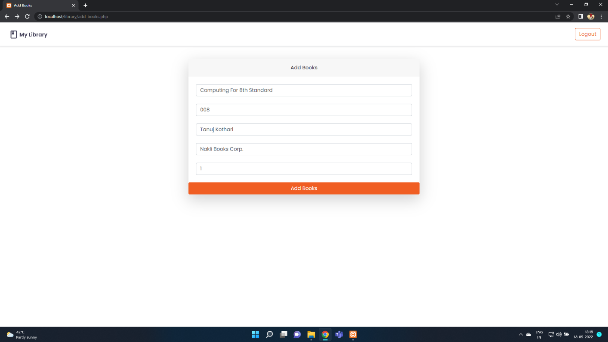
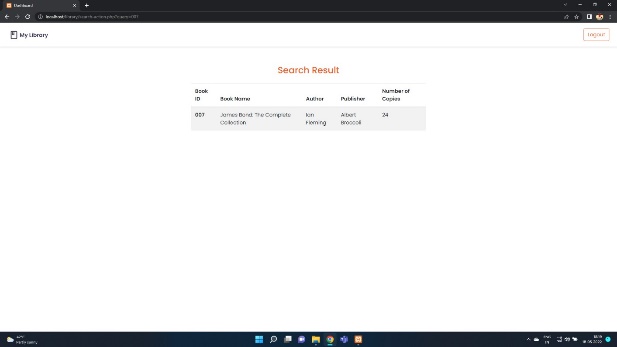
The table is in BCNF, as each attribute is atomic (1NF). issue\_date & return\_date is dependent on student\_id & book\_code, and no partial dependency exists (2NF). No transitive dependency exists in this table (3NF). None of the key attributes are dependent on the key attribute (BCNF).

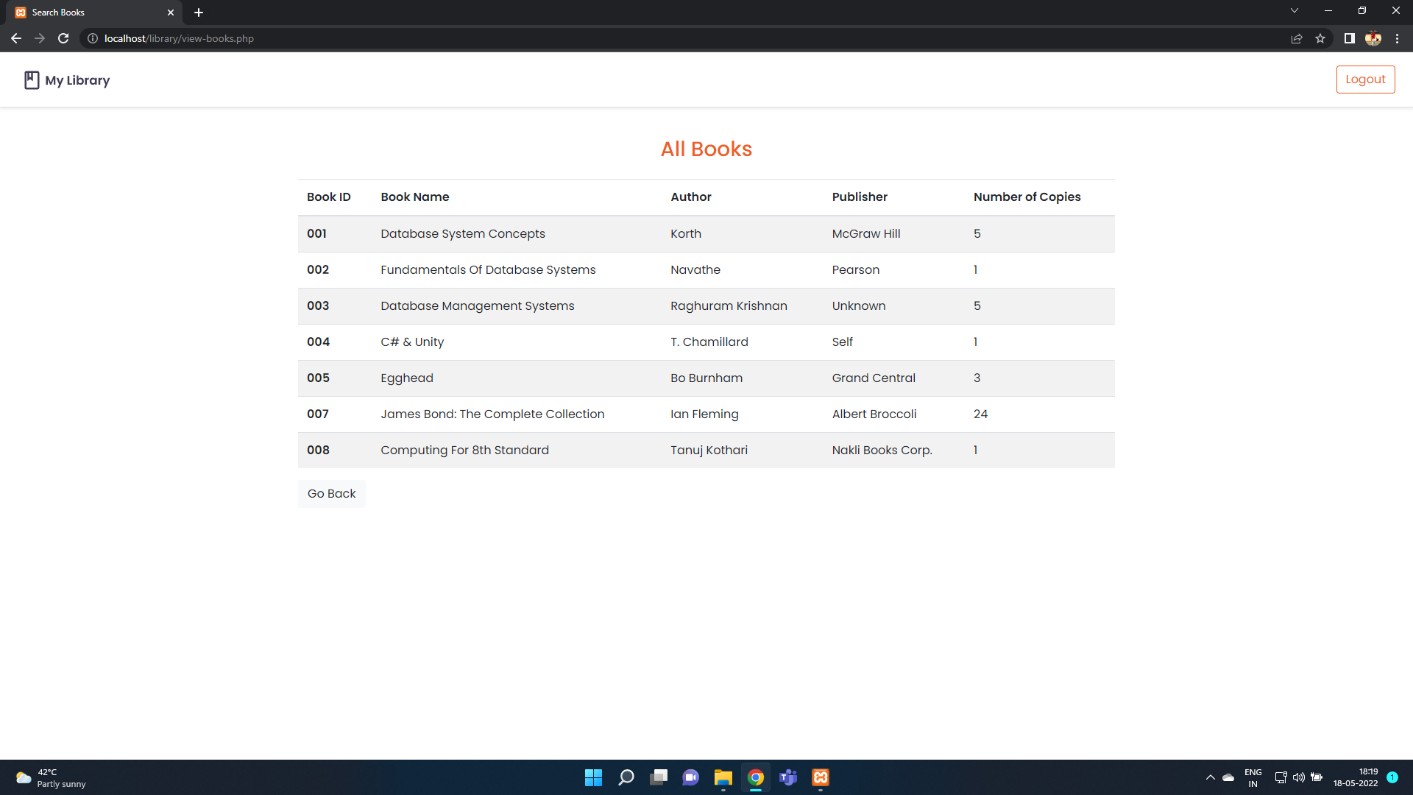
* issued\_by\_fac (f\_id, book\_code, issue\_date, return\_date)

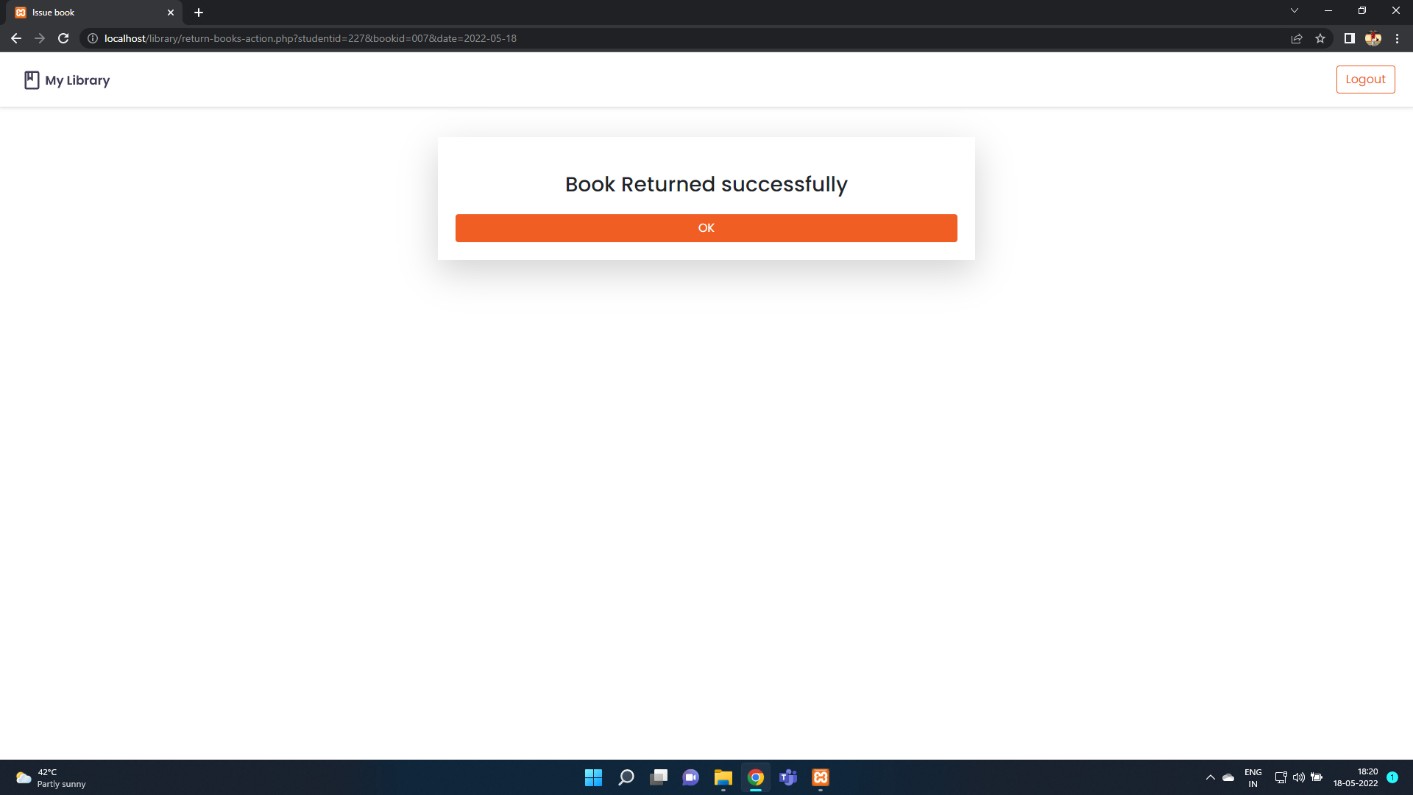
The table is in BCNF, as each attribute is atomic (1NF). issue\_date & return\_date is dependent on student\_id & book\_code, and no partial dependency exists (2NF). No transitive dependency exists in this table (3NF). None of the key attributes are dependent on the key attribute (BCNF).

***Front End***









***Future Prospects***

* Adding more entity sets to have different types of customers.
* Viewing the history of books borrowed by a customer.
* Automate online payment of fine

***Bibliography and References***

* [www.timeto](http://www.timeto)program.com
* www.w3school.com
* www.academy.oracle.com