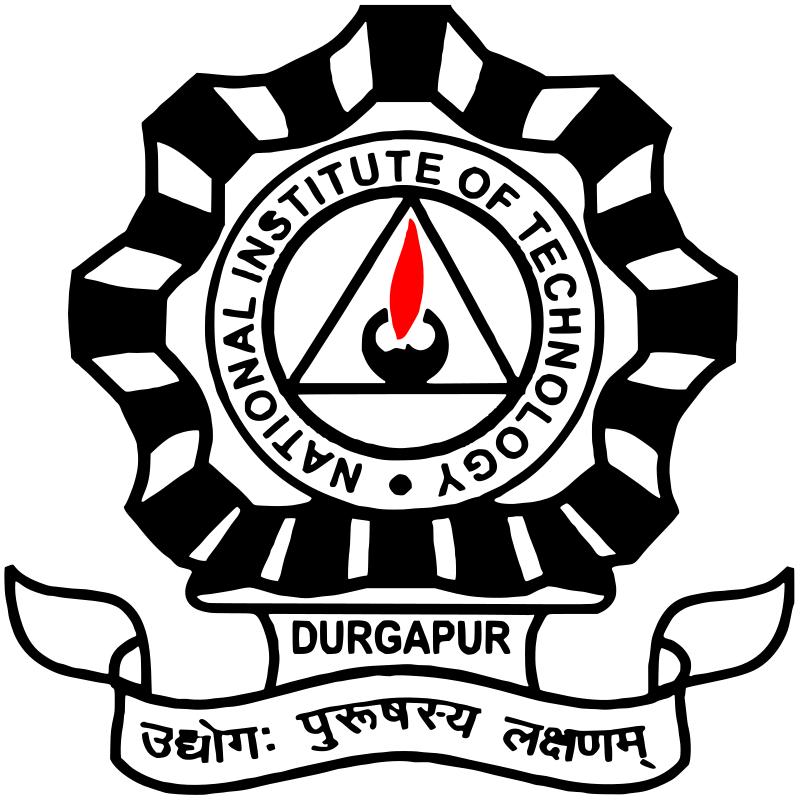
National Institute of Technology Durgapur



Department of Computer Science and Engineering

Mini Project

Database Management Systems Laboratory Report

[CSS-453]

Session: 2024-25

LIBRARY MEMBERSHIP SYSTEM

[Under the advisory of Prof. Prasenjit Choudhury]

**Group 13**

Menuka Hansda 23CS8061

Soubhik Ghosh 23CS8062

Sanket Pathak 23CS8063

Shayantan Biswas 23CS8064

C. Harishwar Reddy 23CS8065

Acknowledgment

We would like to express our heartfelt gratitude to all those who supported and guided us throughout the successful completion of this project.

First and foremost, we extend our sincere thanks to **Dr. Prasenjit Choudhury** of the Department of Computer Science & Engineering for his consistent mentorship and thoughtful insights. His encouragement and deep subject knowledge played a pivotal role in shaping our understanding and approach.

We are equally grateful to **Mr. Shaukat Ali**, Sir’s lab assistant, for his timely technical assistance and steady support during development.

Our heartfelt thanks go to our teammates for their dedication, cooperation, and collaborative spirit, which were essential to the successful execution of this project.

Additionally, we appreciate the tools and resources—particularly MySQL and its extensive documentation—that enabled us to design, build, and test our virtual database system with efficiency and precision.

Overall, this project has been an enriching learning experience, and we are truly thankful for the opportunity to undertake and complete it.

Contents

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Topic** | **Page no.** |
| 1. | Problem statement | 01 |
| 2. | Introduction | 02 |
| 3. | Objective | 02 |
| 4. | Technology used | 02 |
| 5. | Scope of this project | 03 |
| 6. | Overview of the project | 03 |
| 7. | GitHub Repository Link | 03 |
| 8. | Database Design | 04 |
| 9. | ER diagram representing the schema | 05 |
| 10. | Creation of tables and inserting values | 06 |
| 11. | Solutions for each query | 09 |
| 12. | Conclusion | 13 |

Problem statement:

Build a Library Membership System to manage book loans and members with:

# Tables:

* Members (MemberID, Name, Contact, MembershipType)
* Books (BookID, Title, Author, Availability)
* Loans (LoanID, MemberID, BookID, BorrowDate, DueDate, ReturnDate)

# Queries:

* List all overdue books.
* Find the most borrowed book.
* Get member details who borrowed a specific book.
* Retrieve books currently available in the library.
* Count the number of active members.
* Retrieve books borrowed by members with premium membership.
* Identify members who have never returned a book late.
* Find the most borrowed author.
* Calculate the average number of books borrowed per member.
* List books that have never been borrowed

Introduction

This Library Management System uses a relational database to manage books, members, and book loans. It helps ease tasks like adding new records, tracking availability and due dates, issuing books and other records which is easier to read and access in a more organized format.

**Objective:**

* To manage library members and their membership types.
* To maintain book records and their availability status.
* To track book borrow and return details.
* To ensure data consistency using primary and foreign keys.
* To retrieve useful information through SQL queries (e.g., overdue books, most borrowed book).
* To support efficient and organized library operations.

**Technology used:**

1. Denodo: Denodo is a data virtualization platform that allows real-time access to diverse data sources without physical data movement. It simplifies data integration, enabling faster insights and centralized data views.
2. MySQL: MySQL is an open-source relational database management system known for its speed, reliability, and ease of use. It is widely used for managing structured data in web and enterprise applications.
3. SQL Language: It is the standard language for querying and manipulating relational databases. It enables users to retrieve, insert, update, and manage data efficiently through simple, declarative commands

**Scope of this project:**

* Manage member details and membership types.
* Maintain book records and availability.
* Track book loans, returns, and due dates.
* Retrieve useful data through SQL queries.
* Ensure data integrity with relational constraints.
* Can be extended with features like fines or reservations.

# Task:

To build a **Library Membership System** that effectively manages:

* Library Members
* Book Inventory
* Book Loans
* Loan Tracking and Due Dates

**Overview:**

This management system simplifies the day-to-day operations of a library by using a relational database. It helps librarians manage different records and helps user access records easily. The system allows for smooth tracking of available books, overdue returns, and borrowing history, making library data easy to maintain and access. With features like member categorization (e.g., premium members), most borrowed books, and real-time availability, the system improves organization and decision-making within the library.

**GitHub Repository:**

[**https://github.com/Soubhik-coder/CSS403\_MiniProject**](https://github.com/Soubhik-coder/CSS403_MiniProject)

Database Design

**Table MEMBERS :**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| MemberID | |  | | --- | |  |   INT | |  | | --- | |  |   Primary Key |
| Name | |  | | --- | |  |   VARCHAR | Name of the member |
| Contact | VARCHAR | Contact information |
| MembershipType | VARCHAR | Type of membership (e.g., Premium, Regular) |

**Table BOOKS :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | |  | | --- | |  |   **Description** |
| BookID | |  | | --- | |  |  |  | | --- | | INT | | |  | | --- | |  |   Primary Key |
| Title | |  | | --- | |  |   VARCHAR | Book title |
| Author | VARCHAR | Book Author |
| Availabilty | BOOLEAN | TRUE if book is available |

**Table LOANS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | |  | | --- | |  |   **Description** |
| LoanID | |  | | --- | |  |   INT | |  | | --- | |  |   Primary Key |
| MemberID | |  | | --- | |  |   INT | Foreign Key - Members |
| BookID | INT | Foreign Key - Books |
| BorrowDate | DATE | Date of borrowing |
| DueDate | DATE | Expected return date |
| ReturnDate | DATE | Actual return date |

**ER Diagram :**

An **Entity-Relationship (ER) Diagram** visually represents the entities in the system and the relationships among them.

MEMBERS

LOANS

BOOKS

E-R Diagram of Library Management System

Tables

Table 1: **Members (MemberID, Name, Contact, MembershipType)**

## Code:

**CREATE** **TABLE** Members **(**

MemberID INT **PRIMARY** **KEY,**

Name VARCHAR**(**100**)** **NOT** **NULL,**

Contact VARCHAR**(**50**),**

MembershipType VARCHAR**(**20**)**

**);**

**INSERT** **INTO** Members **(**MemberID**,** Name**,** Contact**,** MembershipType**)** **VALUES**

**(**1**,** 'Raj Patel'**,** 'raj@gmail.com'**,** 'Premium'**),**

**(**2**,** 'Priya Sharma'**,** 'priya@yahoo.com'**,** 'Regular'**),**

**(**3**,** 'Amit Kumar'**,** 'amit@gmail.com'**,** 'Premium'**),**

**(**4**,** 'Suresh Reddy'**,** 'suresh@gmail.com'**,** 'Regular'**),**

**(**5**,** 'John Doe'**,** 'john@hotmail.com'**,** 'Premium'**),**

**(**6**,** 'Maria Fernandes'**,** 'maria@yahoo.com'**,** 'Regular'**),**

**(**7**,** 'Ananya Bhattacharya'**,** 'ananya@gmail.com'**,** 'Premium'**),**

**(**8**,** 'David Smith'**,** 'david@hotmail.com'**,** 'Regular'**),**

**(**9**,** 'Neha Gupta'**,** 'neha@gmail.com'**,** 'Regular'**),**

**(**10**,** 'Alex Johnson'**,** 'alex@gmail.com'**,** 'Premium'**);**

## Result:

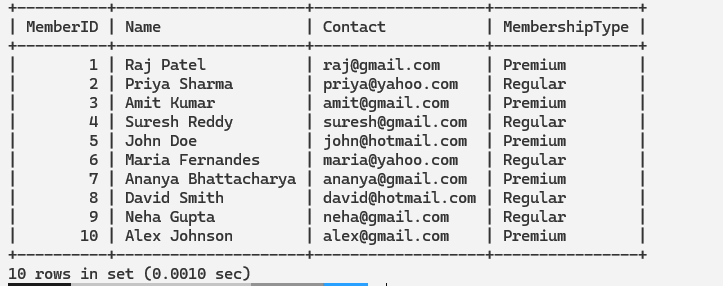


Table 2: **Books (BookID, Title, Author, Availability)**

## Code:

**CREATE** **TABLE** Books **(**

BookID INT **PRIMARY** **KEY,**

Title VARCHAR**(**200**)** **NOT** **NULL,**

Author VARCHAR**(**100**)** **NOT** **NULL,**

Availability VARCHAR**(**10**)** **NOT** **NULL**

**);**

**INSERT** **INTO** Books **(**BookID**,** Title**,** Author**,** Availability**)** **VALUES**

**(**1**,** 'The Great Indian Novel'**,** 'Shashi Tharoor'**,** 'Yes'**),**

**(**2**,** 'Midnight''s Children'**,** 'Salman Rushdie'**,** 'Yes'**),**

**(**3**,** 'The Alchemist'**,** 'Paulo Coelho'**,** 'Yes'**),**

**(**4**,** 'The God of Small Things'**,** 'Arundhati Roy'**,** 'Yes'**),**

**(**5**,** 'A Brief History of Time'**,** 'Stephen Hawking'**,** 'Yes'**),**

**(**6**,** 'To Kill a Mockingbird'**,** 'Harper Lee'**,** 'Yes'**),**

**(**7**,** '1984'**,** 'George Orwell'**,** 'Yes'**),**

**(**8**,** 'Pride and Prejudice'**,** 'Jane Austen'**,** 'Yes'**),**

**(**9**,** 'The Kite Runner'**,** 'Khaled Hosseini'**,** 'Yes'**),**

**(**10**,** 'The Catcher in the Rye'**,** 'J.D. Salinger'**,** 'Yes'**),**

**(**11**,** 'Malgudi Days'**,** 'R.K. Narayan'**,** 'Yes'**),**

**(**12**,** 'The Immortals of Meluha'**,** 'Amish Tripathi'**,** 'Yes'**),**

**(**13**,** 'The White Tiger'**,** 'Aravind Adiga'**,** 'Yes'**),**

**(**14**,** 'The Fault in Our Stars'**,** 'John Green'**,** 'Yes'**),**

**(**15**,** 'Life of Pi'**,** 'Yann Martel'**,** 'Yes'**);**

## Result:

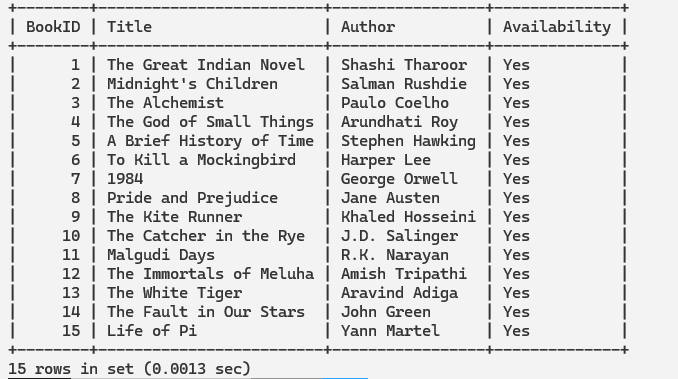


Table 3: **Loans (LoanID, MemberID, BookID, BorrowDate, DueDate, ReturnDate)**

## Code:

**CREATE** **TABLE** Loans **(**

LoanID INT **PRIMARY** **KEY,**

MemberID INT**,**

BookID INT**,**

BorrowDate DATE**,**

DueDate DATE**,**

ReturnDate DATE**,**

**CONSTRAINT** FK\_Member **FOREIGN** **KEY** **(**MemberID**)** **REFERENCES** Members**(**MemberID**),**

**CONSTRAINT** FK\_Book **FOREIGN** **KEY** **(**BookID**)** **REFERENCES** Books**(**BookID**)**

**);**

**INSERT** **INTO** Loans **(**LoanID**,** MemberID**,** BookID**,** BorrowDate**,** DueDate**,** ReturnDate**)** **VALUES**

**(**1**,** 1**,** 1**,** '2023-03-01'**,** '2023-03-15'**,** '2023-03-10'**),**

**(**2**,** 2**,** 2**,** '2023-03-05'**,** '2023-03-20'**,** **NULL),** -- Overdue loan

**(**3**,** 3**,** 3**,** '2023-03-10'**,** '2023-03-25'**,** '2023-03-28'**),** -- Returned late

**(**4**,** 4**,** 4**,** '2023-03-15'**,** '2023-03-30'**,** '2023-03-29'**),**

**(**5**,** 5**,** 5**,** '2023-03-20'**,** '2023-04-04'**,** **NULL),** -- Overdue loan

**(**6**,** 6**,** 6**,** '2023-04-01'**,** '2023-04-15'**,** '2023-04-10'**),**

**(**7**,** 7**,** 7**,** '2023-04-05'**,** '2023-04-20'**,** **NULL),** -- Overdue loan

**(**8**,** 8**,** 8**,** '2023-04-07'**,** '2023-04-22'**,** '2023-04-21'**),**

**(**11**,** 1**,** 11**,** '2023-04-15'**,** '2023-04-30'**,** **NULL),** -- Active loan

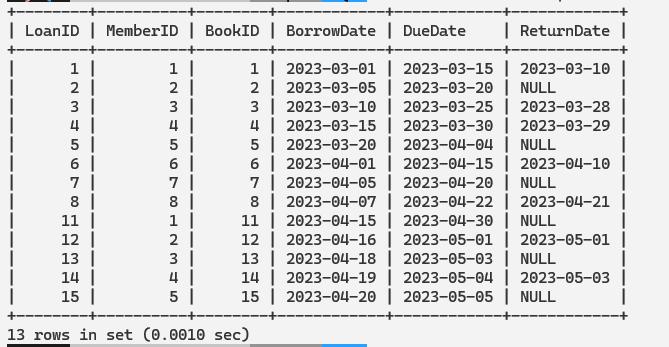
**(**12**,** 2**,** 12**,** '2023-04-16'**,** '2023-05-01'**,** '2023-05-01'**),**

**(**13**,** 3**,** 13**,** '2023-04-18'**,** '2023-05-03'**,** **NULL),** -- Active loan

**(**14**,** 4**,** 14**,** '2023-04-19'**,** '2023-05-04'**,** '2023-05-03'**),**

**(**15**,** 5**,** 15**,** '2023-04-20'**,** '2023-05-05'**,** **NULL);** -- Active loan

## Result:



Queries

Query 1: **List all overdue books.**

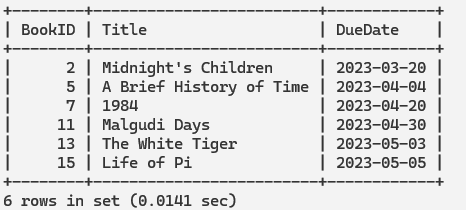
## Code:

**SELECT** B**.**BookID**,** B**.**Title**,** L**.**DueDate

**FROM** Books **AS** B **JOIN** Loans **AS** L **USING(**BookID**)**

**WHERE** L**.**DueDate **<** **CURRENT\_DATE** **AND** L**.**ReturnDate **IS** **NULL;**

## Output:



Query 2: **Find the most borrowed book.**

## Code:

**SELECT** B**.**BookID**,** B**.**Title**,** **COUNT(**L**.**LoanID**)** **AS** BorrowCount

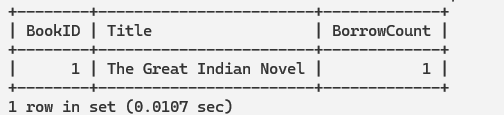
**FROM** Books **AS** B **JOIN** Loans **AS** L **USING(**BookID**)**

**GROUP** **BY** B**.**BookID**,** B**.**Title

**ORDER** **BY** BorrowCount **DESC**

**LIMIT** 1**;**

## Output:



Query 3: Get member details who borrowed a specific book.

## Code:

**DROP** **PROCEDURE** **IF** **EXISTS** GetMembersByBookID**;**

DELIMITER **//**

**CREATE** **PROCEDURE** GetMembersByBookID **(IN** BookID INT**)**

**BEGIN**

**SELECT** m**.**MemberID**,** m**.**Name**,** m**.**Contact**,** m**.**MembershipType**,** l**.**BorrowDate**,**

l**.**DueDate**,** l**.**ReturnDate

**FROM** Loans **AS** l **JOIN** Members **AS** m **USING** **(**MemberID**)**

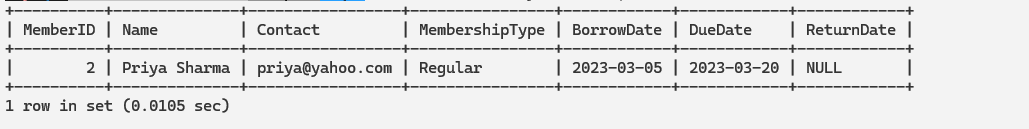
**WHERE** l**.**BookID **=** BookID**;**

**END** **//**

DELIMITER**;**

**CALL** GetMembersByBookID**(**2**);**

## Output:



Query 4: **Retrieve books currently available in the library.**

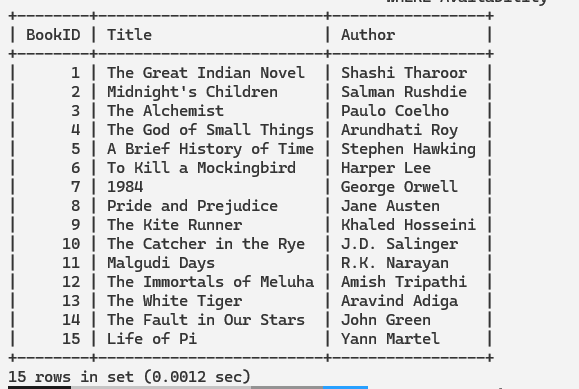
## Code:

**SELECT** BookID**,** Title**,** Author

**FROM** Books

**WHERE** Availability **=** 'Yes'**;**

## Output:



Query 5: **Count the number of active members.**

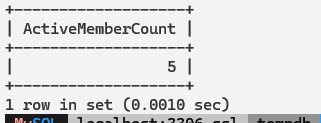
## Code:

**SELECT** **COUNT(DISTINCT** MemberID**)** **AS** ActiveMemberCount

**FROM** Loans

**WHERE** ReturnDate **IS** **NULL;**

## Output:



Query 6: **Retrieve books borrowed by members with premium membership.**

## Code:

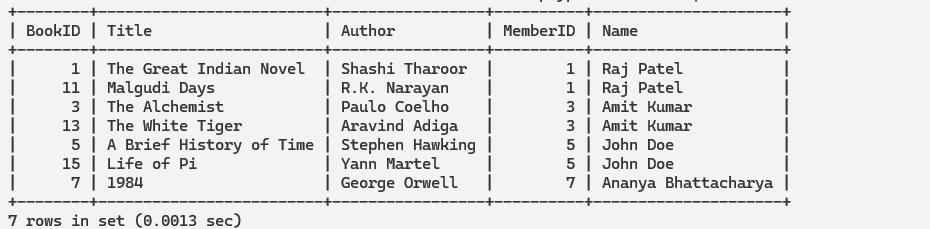
**SELECT** B**.**BookID**,** B**.**Title**,** B**.**Author**,** M**.**MemberID**,** M**.**Name

**FROM** Loans **AS** L **JOIN** Members **AS** M **USING(**MemberID**)**

**JOIN** Books **AS** B **USING(**BookID**)**

**WHERE** M**.**MembershipType **=** 'Premium'**;**

## Output:



Query 7: **Identify members who have never returned a book late.**

## Code:

**SELECT** M**.**MemberID**,** M**.**Name**,** M**.**Contact**,** M**.**MembershipType

**FROM** Members **AS** M

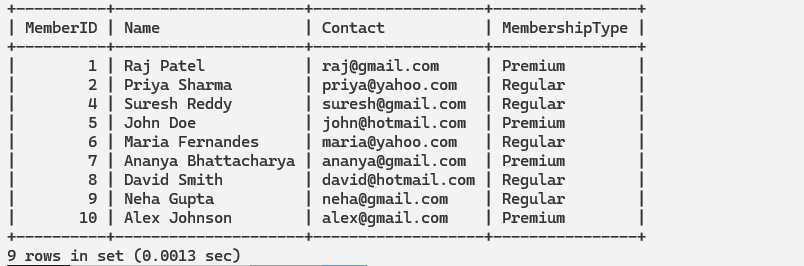
**WHERE** **NOT** **EXISTS** **(**

**SELECT** 1

**FROM** Loans L

**WHERE** L**.**MemberID **=** M**.**MemberID **AND** L**.**ReturnDate **>** L**.**DueDate**);**

## Output:



Query 8: **Find the most borrowed author.**

## Code:

**SELECT** B**.**Author**,** **COUNT(**L**.**LoanID**)** **AS** BorrowCount

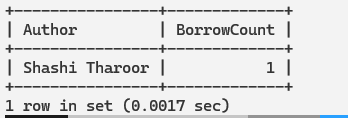
**FROM** Books **AS** B **JOIN** Loans **AS** L **USING(**BookID**)**

**GROUP** **BY** B**.**Author

**ORDER** **BY** BorrowCount **DESC**

**LIMIT** 1**;**

## Output:



Query 9: **Calculate the average number of books borrowed per member**.

## Code:

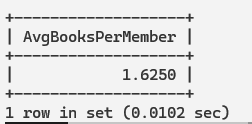
**SELECT** **AVG(**LoanCount**)** **AS** AvgBooksPerMember

**FROM** **(SELECT** MemberID**,** **COUNT(**LoanID**)** **AS** LoanCount

**FROM** Loans

**GROUP** **BY** MemberID**)** **AS** MemberLoans**;**

## Output:



Query 10: **List books that have never been borrowed.**

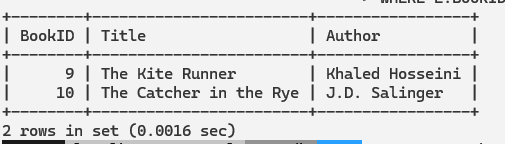
## Code:

**SELECT** B**.**BookID**,** B**.**Title**,** B**.**Author

**FROM** Books **AS** B **LEFT** **JOIN** Loans **AS** L **USING(**BookID**)**

**WHERE** L**.**BookID **IS** **NULL;**

## Output:



Conclusion

The **Library Membership System** project showcases a practical application of database management concepts to address real-life challenges faced by libraries. It effectively models a library's core operations—such as managing member data, tracking book inventory, and monitoring loan transactions—using a structured and relational approach.

**Key Achievements:**

* Designed an efficient relational database schema with well-defined tables and relationships.
* Created an intuitive Entity-Relationship (ER) diagram to visually map out entities and their associations.
* Implemented a variety of SQL queries to retrieve valuable insights, such as identifying overdue books, most borrowed authors, and active members.
* Ensured data integrity and consistency through the use of primary and foreign keys.
* Allowed future scalability by designing the system to support enhancements like fine calculation and reservations.

**Impact on Real Life:**

* Helps libraries **digitize** their operations and reduce manual work.
* Improves **accuracy and accessibility** of member and loan records.
* Enables **quick decision-making** for librarians through meaningful data insights.
* Can be adapted by **educational institutions, public libraries, or small community libraries** with minimal customization

**Final Thoughts:**

This project not only strengthened our understanding of **SQL, data modeling, and normalization**, but also demonstrated the importance of database systems in managing day-to-day operations. Through this, we’ve gained practical experience in transforming real-world problems into structured, logical solutions using DBMS principles.