

DATABASE DESIGN AND IMPLEMENTATION FOR LIBRARY MANAGEMENT SYSTEM {LMS}

Professor: IAN BECKER

Student Name: Meghana Meka,

Siri Gundlapally

Course: MSIS_537L_256_23F DATA MGT I

Semester: Fall-2023

Marist College

Table of Contents:

- 1. Executive summary**
- 2. Overview**
- 3. Objectives**
- 4. Database Design**
- 5. Implementation notes**
- 6. Known problems**
- 7. Future enhancements**
- 8. Conclusion**
- 9. References**

1. Executive Summary

The Library Management System (LMS) was developed as a collaborative effort in response to difficulties faced by traditional libraries and a shared vision for a more streamlined and sophisticated solution. Long talks among stakeholders, including librarians, administrators, and technology professionals, all of whom were seeking to meet the changing face of libraries, sparked the idea. The inspiration for the project came from the observations that time-consuming processes hindered the productivity of libraries, which were progressively becoming custodians of both books and different digital and multimedia resources. Thus, I have envisioned an integrated system that would automate regular operations and improve the whole library experience, allowing librarians to focus on curating and increasing collections rather than managing administrative complications.

Collaborative brainstorming sessions were critical in creating the concept for the LMS. Librarians gave insights into their day-to-day issues, providing unique perspectives on the practical aspects of library administration. Meanwhile, technology specialists contributed their understanding of database administration, user interface design, and new automation technologies to create a system that would easily integrate into the digital era. Key components of the LMS were inspired by successful implementations in other industries, such as the use of bar code scanning and RFID technology in retail and logistics applications. The talks converged on a concept of a user-friendly interface, strong catalog management, and advanced search and discovery capabilities, all targeted at improving the user experience for both librarians and users.

Thus, this collaborative approach enabled the development of a system capable of centralizing activities across multiple branches seamlessly interacting with internet resources to meet the demands of modern libraries and fuelling the development of a solution positioned to guide libraries into a more efficient and technologically capable future. In essence, the Library Management System arose to pave way for the collaboration and innovations facing the dynamic nature of libraries infrastructure in the twenty-first century.

2. Library Database Management Overview

By smoothly integrating technology, the Library Database Management System represents a paradigm change in library operations. It enriches the typical library experience with personalized services and faster procedures as an enhanced Catalog Management system. The system's core, a comprehensive e-based book collection, unifies essential modules, allowing for the easy addition and updating of materials for accessible user experiences. It goes beyond traditional cataloging by including borrowing information and exact access rights, allowing it to adapt to a broad user base. The Transaction Management module, which manages check-ins and check-outs efficiently, accelerates library transactions and resource rotation. This enables libraries to make informed decisions, maximizing resource allocation and service growth. The Learning Management System (LMS) facilitates a dynamic evolution in library operations by emphasizing user satisfaction, operational efficiency, and improved accessibility. This watershed moment in library services corresponds with the digital era while keeping the primary goal of providing seamless access to information in mind.

3. Objectives

Objectively, LMS is entrusted to:

- a. Automate Book Inventory Management
- b. Ensure Accurate and Timely Transactions
- c. Facilitate User Record Accessibility
- d. Generate Comprehensive Library Reports
- e. Enable Secure Online Transactions

4. Database Design and Implementation

4.1. Database Design

i. Data Dictionaries

ATTRIBUTES	DESCRIPTION	DATA TYPES	SIZE
Book_ID (PK)	Primary Book ID	INT	11
Title	Title of the book	VARCHAR	255
Author_ID (FK)	Identifier Linking to Author able	INT	11
Genre_ID (FK)	Identifier linking to Genre Table	INT	11
ISBN	International Std Book Number	BIGINT	20
Quantity	Number of copies of book available	INT	11
Availability	Indicates if the Book is available for Borrowing	VARCHAR	255

Book table - table that stores Book details.

ATTRIBUTES	DESCRIPTION	DATA TYPE	SIZE
User_ID (PK)	Distinct number of users	INT	11
Full_Name	Full Name	VARCHAR	255
Email	Email address	VARCHAR	255
Phone	Users' Contact phone	VARCHAR	255
Address	Users' Residential address	VARCHAR	255
Membership_Type	Type of Library Membership	VARCHAR	255
Join_Date	Date when the User joined the Library	DATE	

User Table - This table stores Library User's Information.

ATTRIBUTES	DESCRIPTION	DATA-TYPES	LENGTH
Transaction_ID (PK)	Unique identifier of each transaction.	INT	11
User_ID (FK)	Identifier linking to User Table	INT	11
Book_ID (FK)	Identifier linking to Book Table	INT	255
Borrow_Date	Borrowed Date	DATE	
Return_Date	Returned Date	DATE	
Status	Current status of the transaction.	VARCHAR	255

Transaction table – for storing information of books in and out of the Library system

ATTRIBUTES	DESCRIPTION	DATA-TYPES	LENGTH
Genre_ID (PK)	Book Genre ID	INT	11
Genre_Name	Genre Title	VARCHAR	255

Genre Table – This Stores all book genres in the Library System.

ATTRIBUTES	DESCRIPTION	DATA-TYPES	LENGTH
Author_ID (PK)	Unique identifier of each Book Genre	INT	11
Author_Name	Name of Author	VARCHAR	255

Author table – this tuple will store Author's information

ATTRIBUTES	DESCRIPTIONS	DATA-TYPE	LENGTH
Branch_id (PK)	Distinct Branch ID	INT	11
Branch_name	Branch Name	VARCHAR	255
Location	Branch Location	VARCHAR	255
Total_Books	Total Number of Books available at the Branch	INT	11

Library Branch Table

ATTRIBUTES	DESCRIPTIONS	DATA-TYPE	LENGTH
Publisher_id (PK)	Number each publisher Genre	INT	11
Publisher_Name	Name of the Book's Publisher	VARCHAR	255
Country	Origin of the Publisher	VARCHAR	255
Year_Founded	Year founded by the Publisher	INT	11

Publisher Table

ATTRIBUTES	DESCRIPTIONS	DATA-TYPE	LENGTH
Copy_ID (PK)	Unique number of copies of the Book	INT	11
Book_ID (FK)	Identifier linking to Books' Table	INT	11
Branch_ID (FK)	Identifier linking to Branch's Table	INT	11
Status	Total Number of Books available at the Branch	VARCHAR	255

Book Copy Table

ATTRIBUTES	DESCRIPTIONS	DATA TYPE	LENGTH
History_id (PK)	History ID	INT	
User_ID (FK)	Identifier linking to user's Table	INT	11
Book_ID (FK)	Identifier linking to Book's Table	INT	11
Borrow_Date	Borrowed date	DATE	
Return_Date	Returned date	DATE	
Late_Fee	Fee charged for late return of the book	INT	11

Borrower History Table

a. Logical Database Design

This diagram below is a logical database design for the LMS. It showcases the relationships between the Database entities.

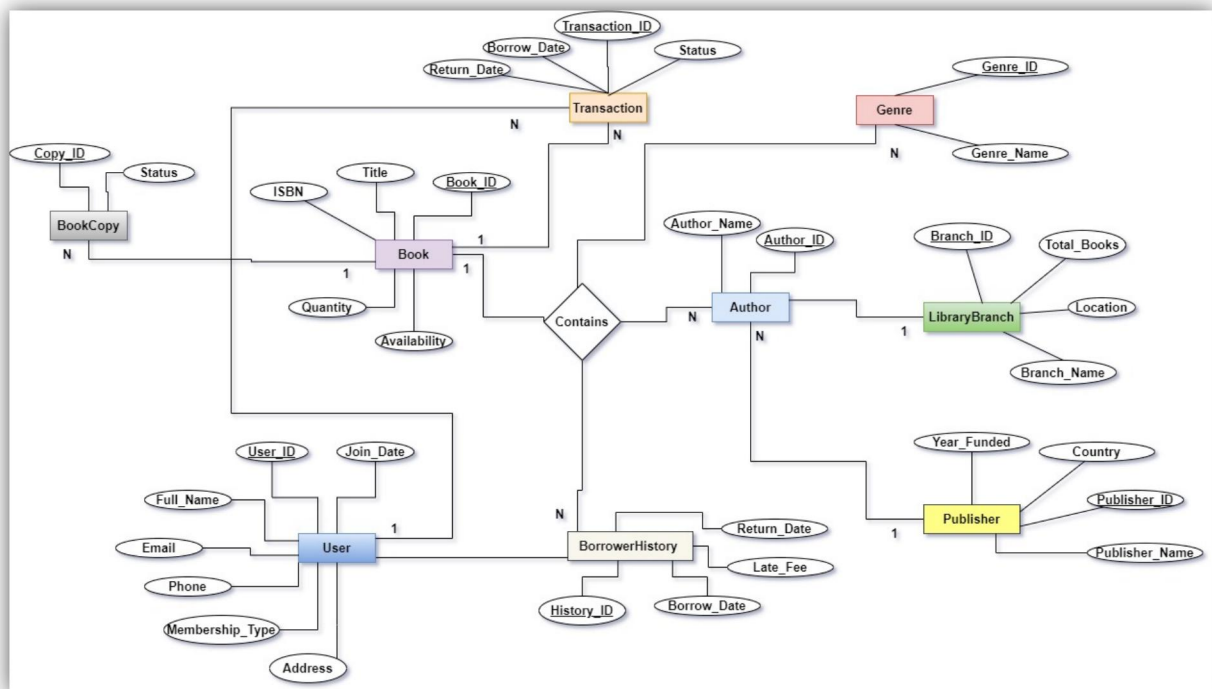


Figure 1. Logical Database Design

b. Physical Database Design

Physical database design is Vital for the performance, scalability and maintainability of the Learning Management System. It must be carefully implemented to ensure the system aligns with its objectives. The figure below shows a designed physical database Schema of LMS.

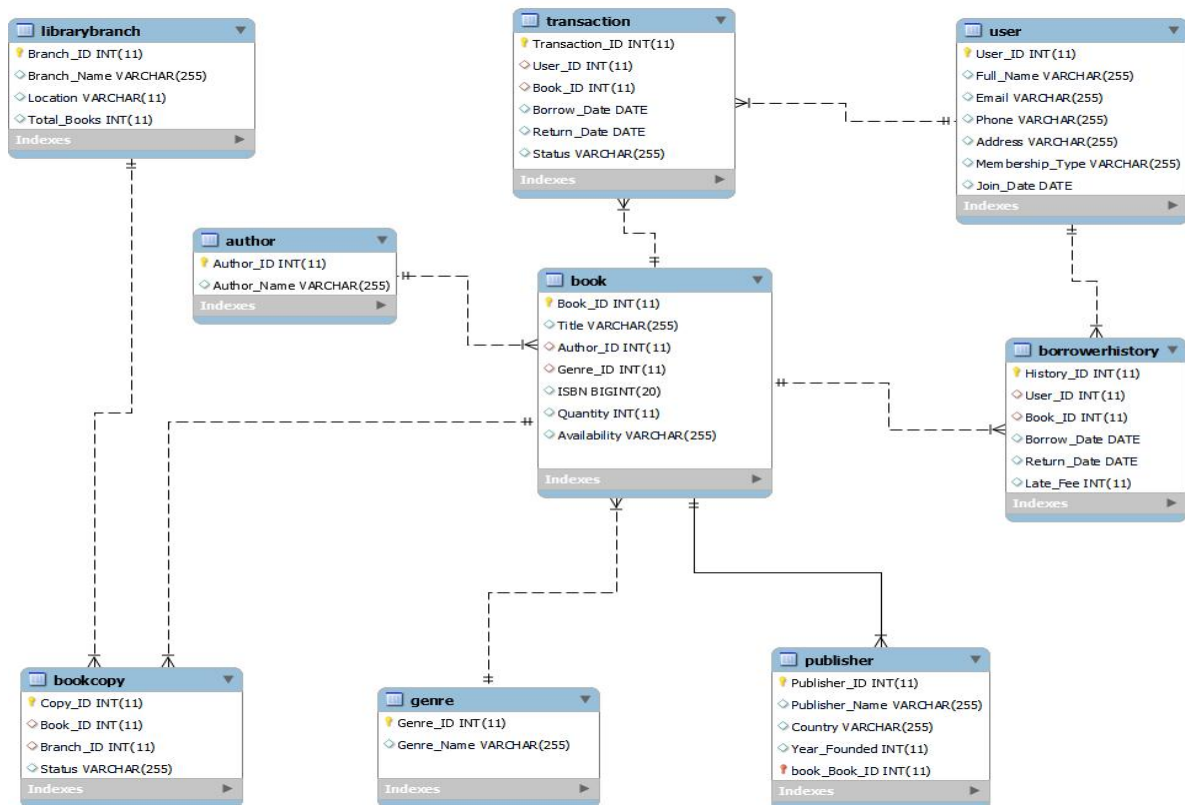


Figure 2. LMS physical database Design.

c. Data and Their Respective Sources for LMS

In the present implementation of the Library Management System, a variety of data formats are essential to guaranteeing the smooth operation of library services. To manage book, user, transaction, genre, book copy, publisher, library branch, author, and borrower history information with ease, the system draws on several data sources. The many data sources represent the facets of library management that are complex. This is a detailed summary.:

- **Book Information**

The structured information about books is gathered by publishers, external databases, and Manual entry into this entity. ISBN, Book_ID, Title, Author_ID, Quantity, Availability, and Genre_ID are the important data stored in the book's entity.

- **User Information**

User registration processes and external databases serve as sources for user information. Critical details encompass User_ID, Full_Name, Email, Phone, Address, Membership_Type, and Join_Date.

- **Transaction Details**

User interactions with the library system generate transaction data. The Transaction Table captures information such as Unique Transaction, User and Book Identifier.

- **Genre Classification**

Genre information is often predefined and managed within the library system. The Genre Table consists of Genre_ID and Genre_Name.

- **Author Information**

Authors' details are sourced from publishers, author submissions, or external databases. The Author Table includes Author_ID and Author_Name.

- **Library Branch Information**

The Library Branch Table obtains details from internal management processes. Information encompasses Branch_ID, Branch_Name, Location, and Total_Books.

- **Publisher Information**

Publisher details are managed internally within the library system and may be obtained from external databases. The Publisher Table consists of Publisher_ID, Publisher_Name, Country, and Year_Founded.

- **Book Copy Status**

Book copy status is updated based on user transactions and book conditions. The Book Copy Table includes Copy_ID, Book_ID, Branch_ID, and Status (Available, Checked Out, Damaged).

- **Borrowing History**

User transactions contribute to the Borrower History Table. Information comprises of Unique number of History, User and Book with Late_Fee information.

iv. Functional Dependencies in the Database System

Functional dependencies of the Library Management System include:

- **User Table**

user_id -> full_name, email, phone, address, membership_type, join_date

- **Book Table**

book_id -> title, author_id, ISBN, genre_id, quantity, availability

- **Transaction Table**

transaction_id -> user_id, book_id, borrow_date, return_date, status

- **Genre Table**

genre_id -> genre_name

- **Author Table**

author_id -> author_name

- **Library Branch Table**

branch_id -> branch_name, location, total_books

- **Publisher Table**

publisher_id -> publisher_name, country, year_founded

- **Book Copy Table**

copy_id -> book_id, branch_id, status

- **Borrower History Table**

history_id -> user_id, book_id, borrow_date, return_date, late_fee

iv. SQL Statements for Implementation of the Schema

The following are screen captures of SQL syntax for creating LMS schema and individual tables

```
-- SQL syntax for creating LMS schema
CREATE DATABASE LMS;
USE LMS;
```

LMS schema

```
-- Create Book Table
CREATE TABLE Book (
    Book_ID INT(11) PRIMARY KEY,
    Title VARCHAR(255),
    Author_ID INT(11),
    Genre_ID INT(11),
    ISBN BIGINT(20),
    Quantity INT(11),
    Availability VARCHAR(255),
    FOREIGN KEY (Author_ID) REFERENCES Author(Author_ID),
    FOREIGN KEY (Genre_ID) REFERENCES Genre(Genre_ID)
);
```

Book

```
-- Create User Table
CREATE TABLE User (
    User_ID INT(11) PRIMARY KEY,
    Full_Name VARCHAR(255),
    Email VARCHAR(255),
    Phone VARCHAR(255),
    Address VARCHAR(255),
    Membership_Type VARCHAR(255),
    Join_Date DATE
);
```

User

1. Transaction

```
-- Create Transaction Table
CREATE TABLE Transaction (
    Transaction_ID INT(11) PRIMARY KEY,
    User_ID INT(11),
    Book_ID INT(11),
    Borrow_Date DATE,
    Return_Date DATE,
    Status VARCHAR(255),
    FOREIGN KEY (User_ID) REFERENCES User(User_ID),
    FOREIGN KEY (Book_ID) REFERENCES Book(Book_ID)
);
```

2. Genre

```
-- Create Genre Table
CREATE TABLE Genre (
    Genre_ID INT(11) PRIMARY KEY,
    Genre_Name VARCHAR(255)
);
```

3. Author

```
-- Create Author Table
CREATE TABLE Author (
    Author_ID INT(11) PRIMARY KEY,
    Author_Name VARCHAR(255)
);
```

4. Library Branch

```
-- Create LibraryBranch Table
CREATE TABLE LibraryBranch (
    Branch_ID INT(11) PRIMARY KEY,
    Branch_Name VARCHAR(255),
    Location VARCHAR(11),
    Total_Books INT(11)
);
```

5. Publisher

```
-- Create Publisher Table
CREATE TABLE Publisher (
    Publisher_ID INT(11) PRIMARY KEY,
    Publisher_Name VARCHAR(255),
    Country VARCHAR(255),
    Year_Founded INT(11)
);
```

6. Book Copy

```
-- Create BookCopy Table
CREATE TABLE BookCopy (
    Copy_ID INT(11) PRIMARY KEY,
    Book_ID INT(11),
    Branch_ID INT(11),
    Status VARCHAR(255),
    FOREIGN KEY (Book_ID) REFERENCES Book(Book_ID),
    FOREIGN KEY (Branch_ID) REFERENCES LibraryBranch(Branch_ID)
);
```

7. Borrower History

```
-- Create BorrowerHistory Table
CREATE TABLE BorrowerHistory (
    History_ID INT(11) PRIMARY KEY,
    User_ID INT(11),
    Book_ID INT(11),
    Borrow_Date DATE,
    Return_Date DATE,
    Late_Fee INT(11),
    FOREIGN KEY (User_ID) REFERENCES User(User_ID),
    FOREIGN KEY (Book_ID) REFERENCES Book(Book_ID)
);
```

iv. Sample Data Insertion into the Schema and its Outputs

Here is the SQL syntax for inserting sample data into the Respective Tables with their corresponding outputs.

- **Data Insertion into Books Table and display**

```

-- Insert data into Book table
INSERT INTO Book (Book_ID, Title, Author_ID, Genre_ID, ISBN, Quantity, Availability)
VALUES
(1, 'The Great Gatsby', 1, 1, 1234567890, 5, 'Available'),
(2, 'To Kill a Mockingbird', 2, 2, 2345678901, 8, 'Available'),
(3, '1984', 3, 3, 3456789012, 10, 'Checked Out'),
(4, 'Pride and Prejudice', 4, 1, 4567890123, 3, 'Available'),
(5, 'The Catcher in the Rye', 5, 2, 5678901234, 6, 'Checked Out'),
(6, 'Brave New World', 3, 3, 6789012345, 7, 'Available'),
(7, 'Wuthering Heights', 4, 1, 7890123456, 4, 'Available'),
(8, 'To the Lighthouse', 5, 2, 8901234567, 2, 'Available'),
(9, 'The Hobbit', 1, 4, 9012345678, 9, 'Checked Out'),
(10, 'The Odyssey', 2, 5, 1234567809, 12, 'Available');

-- Displaying the LMS Books Information
select * From Book;

```

Functional Dependencies

book_id -> title, author_id, ISBN, genre_id, quantity, availability

Sample Data

```

201 -- Displaying the LMS Books Information
202 select * From Book;

```

Book_ID	Title	Author_ID	Genre_ID	ISBN	Quantity	Availability
2	To Kill a Mockingbird	2	2	2345678901	8	Available
3	1984	3	3	3456789012	10	Checked Out
4	Pride and Prejudice	4	1	4567890123	3	Available
5	The Catcher in the Rye	5	2	5678901234	6	Checked Out
6	Brave New World	3	3	6789012345	7	Available
7	Wuthering Heights	4	1	7890123456	4	Available
8	To the Lighthouse	5	2	8901234567	2	Available
9	The Hobbit	1	4	9012345678	9	Checked Out
10	The Odyssey	2	5	1234567809	12	Available

- Data Insertion into Users Table

```

-- Insert data into User table
INSERT INTO User (User_ID, Full_Name, Email, Phone, Address, Membership_Type, Join_Date)
VALUES
(1, 'John Doe', 'john.doe@example.com', '123-456-7890', '123 Main St', 'Regular', '2023-01-01'),
(2, 'Jane Smith', 'jane.smith@example.com', '987-654-3210', '456 Oak St', 'Premium', '2023-02-15'),
(3, 'Bob Johnson', 'bob.johnson@example.com', '555-123-4567', '789 Pine St', 'Regular', '2023-03-20'),
(4, 'Alice Williams', 'alice.williams@example.com', '111-222-3333', '101 Elm St', 'Premium', '2023-04-10'),
(5, 'Charlie Brown', 'charlie.brown@example.com', '444-555-6666', '202 Birch St', 'Regular', '2023-05-05'),
(6, 'Eva Davis', 'eva.davis@example.com', '777-888-9999', '303 Cedar St', 'Premium', '2023-06-18'),
(7, 'Frank Miller', 'frank.miller@example.com', '333-999-7777', '404 Maple St', 'Regular', '2023-07-30'),
(8, 'Grace Wilson', 'grace.wilson@example.com', '666-444-1111', '505 Walnut St', 'Premium', '2023-08-22'),
(9, 'David Lee', 'david.lee@example.com', '222-666-8888', '606 Pine St', 'Regular', '2023-09-14'),
(10, 'Sophie Turner', 'sophie.turner@example.com', '999-111-2222', '707 Oak St', 'Premium', '2023-10-09');

-- Displaying the LMS User Information
select * From User;

```

Functional Dependencies

user_id -> full_name, email, phone, address, membership_type, join_date

Sample Data


```

204 -- Displaying the LMS User Information
205 • select * From User;

```

User_ID	Full_Name	Email	Phone	Address	Membership_Type	Join_Date
1	John Doe	john.doe@example.com	123-456-7890	123 Main St	Regular	2023-01-01
2	Jane Smith	jane.smith@example.com	987-654-3210	456 Oak St	Premium	2023-02-15
3	Bob Johnson	bob.johnson@example.com	555-123-4567	789 Pine St	Regular	2023-03-20
4	Alice Williams	alice.williams@example.com	111-222-3333	101 Elm St	Premium	2023-04-10
5	Charlie Brown	charlie.brown@example.com	444-555-6666	202 Birch St	Regular	2023-05-05
6	Eva Davis	eva.davis@example.com	777-888-9999	303 Cedar St	Premium	2023-06-18
7	Frank Miller	frank.miller@example.com	333-999-7777	404 Maple St	Regular	2023-07-30
8	Grace Wilson	grace.wilson@example.com	666-444-1111	505 Walnut St	Premium	2023-08-22
9	David Lee	david.lee@example.com	222-666-8888	606 Pine St	Regular	2023-09-14
10	Sophie Turner	sophie.turner@example.com	999-111-2222	707 Oak St	Premium	2023-10-09
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Data Insertion into Transactions Table**

```

-- Insert data into Transaction table
INSERT INTO Transaction (Transaction_ID, User_ID, Book_ID, Borrow_Date, Return_Date, Status)
VALUES
(1, 1, 3, '2023-01-05', '2023-01-15', 'Closed'),
(2, 2, 5, '2023-02-20', '2023-03-01', 'Closed'),
(3, 3, 9, '2023-03-25', '2023-04-05', 'Closed'),
(4, 4, 2, '2023-04-15', '2023-04-25', 'Closed'),
(5, 5, 8, '2023-05-10', '2023-05-20', 'Closed'),
(6, 6, 1, '2023-06-25', '2023-07-05', 'Closed'),
(7, 7, 7, '2023-07-15', '2023-07-25', 'Closed'),
(8, 8, 6, '2023-08-01', '2023-08-11', 'Closed'),
(9, 9, 4, '2023-09-20', '2023-09-30', 'Closed'),
(10, 10, 10, '2023-10-15', '2023-10-25', 'Closed');

-- Displaying the LMS Transaction Information
select * from Transaction;

```

Functional dependency

transaction_id -> user_id, book_id, borrow_date, return_date, status

Sample

Data

```

207 -- Displaying the LMS Transaction Information
208 • select * from Transaction;

```

Transaction_ID	User_ID	Book_ID	Borrow_Date	Return_Date	Status
1	1	3	2023-01-05	2023-01-15	Closed
2	2	5	2023-02-20	2023-03-01	Closed
3	3	9	2023-03-25	2023-04-05	Closed
4	4	2	2023-04-15	2023-04-25	Closed
5	5	8	2023-05-10	2023-05-20	Closed
6	6	1	2023-06-25	2023-07-05	Closed
7	7	7	2023-07-15	2023-07-25	Closed
8	8	6	2023-08-01	2023-08-11	Closed
9	9	4	2023-09-20	2023-09-30	Closed
10	10	10	2023-10-15	2023-10-25	Closed
NULL	NULL	NULL	NULL	NULL	NULL

- **Data Insertion into Genre Book Table**

```

-- Insert data into Genre table
INSERT INTO Genre (Genre_ID, Genre_Name)
VALUES
(1, 'Fiction'),
(2, 'Classic'),
(3, 'Science Fiction'),
(4, 'Fantasy'),
(5, 'Epic');

-- Displaying Genre Information
select * from Genre;

```

Output


```

210  -- Displaying Genre Information
211  •  select * from Genre;

```

Genre_ID	Genre_Name
1	Fiction
2	Classic
3	Science Fiction
4	Fantasy
5	Epic
NULL	NULL

- **Data Insertion into Author's Table**

```

-- Insert data into Author table
INSERT INTO Author (Author_ID, Author_Name)
VALUES
(1, 'F. Scott Fitzgerald'),
(2, 'Harper Lee'),
(3, 'George Orwell'),
(4, 'Jane Austen'),
(5, 'J.D. Salinger');

-- Selecting Author Information
select * from Author;

```

Output

```

213  -- Selecting Author Information
214  •  select * from Author;

```

Author_ID	Author_Name
1	F. Scott Fitzgerald
2	Harper Lee
3	George Orwell
4	Jane Austen
5	J.D. Salinger
NULL	NULL

- **Data Insertion into LMS Branch Table**

```

-- Insert data into LibraryBranch table
INSERT INTO LibraryBranch (Branch_ID, Branch_Name, Location, Total_Books)
VALUES
(1, 'Main Branch', 'City Center', 5000),
(2, 'East Branch', 'East Side', 3000),
(3, 'West Branch', 'West Side', 2500),
(4, 'North Branch', 'North Side', 2000),
(5, 'South Branch', 'South Side', 1800);

-- Selecting the Library Branches of LMS
select * from LibraryBranch;

```

Sample data. Functional dependency; Branch_ID -> Branch_Name, Location, Total_Books

```
216 -- Selecting the Library Branches of LMS
217 • select * from LibraryBranch;
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
Branch_ID	Branch_Name	Location	Total_Books	
1	Main Branch	City Center	5000	
2	East Branch	East Side	3000	
3	West Branch	West Side	2500	
4	North Branch	North Side	2000	
5	South Branch	South Side	1800	
NULL	NULL	NULL	NULL	

- **Data Insertion into Publisher's Table**

```
-- Insert data into Publisher table
INSERT INTO Publisher (Publisher_ID, Publisher_Name, Country, Year_Founded)
VALUES
(1, 'Penguin Random House', 'USA', 1927),
(2, 'HarperCollins', 'USA', 1989),
(3, 'Simon & Schuster', 'USA', 1924),
(4, 'Macmillan Publishers', 'UK', 1843),
(5, 'Hachette Livre', 'France', 1826);

-- Display Publisher Table
select * From Publisher;
```

Sample - Functional Dependencies; Publisher_ID -> Publisher_Name, Country, year_Founded

```
219 -- Display Publisher Table
220 • select * From Publisher;
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
Publisher_ID	Publisher_Name	Country	Year_Founded	
1	Penguin Random House	USA	1927	
2	HarperCollins	USA	1989	
3	Simon & Schuster	USA	1924	
4	Macmillan Publishers	UK	1843	
5	Hachette Livre	France	1826	
NULL	NULL	NULL	NULL	

- **Data Insertion into the information related to Copies of Book.**

```
-- Insert data into BookCopy table
INSERT INTO BookCopy (Copy_ID, Book_ID, Branch_ID, Status)
VALUES
(1, 1, 1, 'Available'),
(2, 2, 2, 'Checked Out'),
(3, 3, 3, 'Available'),
(4, 4, 4, 'Available'),
(5, 5, 5, 'Checked Out'),
(6, 6, 1, 'Available'),
(7, 7, 2, 'Available'),
(8, 8, 3, 'Available'),
(9, 9, 4, 'Checked Out'),
(10, 10, 5, 'Available');

-- Display BookCopies
select * From BookCopy;
```

Functional Dependencies: Copy_ID ->

Book_ID,Branch_ID,Status]

```
222 -- Display BookCopies
223 • select * From BookCopy;
```

	Copy_ID	Book_ID	Branch_ID	Status
▶	1	1	1	Available
	2	2	2	Checked Out
	3	3	3	Available
	4	4	4	Available
	5	5	5	Checked Out
	6	6	1	Available
	7	7	2	Available
	8	8	3	Available
	9	9	4	Checked Out
	10	10	5	Available
•	NULL	NULL	NULL	NULL

- information related Borrowing History of LMS.

```
-- Insert data into BorrowerHistory table
INSERT INTO BorrowerHistory (History_ID, User_ID, Book_ID, Borrow_Date, Return_Date, Late_Fee)
VALUES
(1, 1, 3, '2023-01-05', '2023-01-15', 0),
(2, 2, 5, '2023-02-20', '2023-03-01', 0),
(3, 3, 9, '2023-03-25', '2023-04-05', 0),
(4, 4, 2, '2023-04-15', '2023-04-25', 0),
(5, 5, 8, '2023-05-10', '2023-05-20', 0),
(6, 6, 1, '2023-06-25', '2023-07-05', 0),
(7, 7, 7, '2023-07-15', '2023-07-25', 0),
(8, 8, 6, '2023-08-01', '2023-08-11', 0),
(9, 9, 4, '2023-09-20', '2023-09-30', 0),
(10, 10, 10, '2023-10-15', '2023-10-25', 0);

-- Display Borrowing History
select * From BorrowerHistory;
```

Functional Dependencies;history_id ->user_id,book_id,borrow_date, return_date, late_fee

```
225 -- Display Borrowing History
226 • select * From BorrowerHistory;
```

	History_ID	User_ID	Book_ID	Borrow_Date	Return_Date	Late_Fee
▶	1	1	3	2023-01-05	2023-01-15	0
	2	2	5	2023-02-20	2023-03-01	0
	3	3	9	2023-03-25	2023-04-05	0
	4	4	2	2023-04-15	2023-04-25	0
	5	5	8	2023-05-10	2023-05-20	0
	6	6	1	2023-06-25	2023-07-05	0
	7	7	7	2023-07-15	2023-07-25	0
	8	8	6	2023-08-01	2023-08-11	0
	9	9	4	2023-09-20	2023-09-30	0
	10	10	10	2023-10-15	2023-10-25	0
•	NULL	NULL	NULL	NULL	NULL	NULL

v. SQL Views and their Sample Output

SQL syntax for creating and Displaying regular Views and materialized views.

1. Regular Views




a. View of Available Books

```
-- Regular Views
-- View for available books
CREATE VIEW AvailableBooks AS
SELECT Book_ID, Title, Quantity
FROM Book
WHERE Availability = 'Available';
select * from AvailableBooks;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Book_ID	Title	Quantity	
1	The Great Gatsby	5	
2	To Kill a Mockingbird	8	
4	Pride and Prejudice	3	
6	Brave New World	7	
7	Wuthering Heights	4	
8	To the Lighthouse	2	
10	The Odyssey	12	

b. View of the Checked-Out Books

```
-- 2. View for checked out books
CREATE VIEW CheckedOutBooks AS
SELECT
    B.Book_ID,
    B.Title,
    B.Quantity,
    A.Author_Name,
    G.Genre_Name
FROM
    Book B
JOIN
    Author A ON B.Author_ID = A.Author_ID
JOIN
    Genre G ON B.Genre_ID = G.Genre_ID
WHERE
    B.Availability = 'Checked Out';
```

Result Grid				Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: <input type="checkbox"/>
	Book_ID	Title	Quantity	Author_Name	Genre_Name	
▶	3	1984	10	George Orwell	Science Fiction	
	5	The Catcher in the Rye	6	J.D. Salinger	Classic	
	9	The Hobbit	9	F. Scott Fitzgerald	Fantasy	

The *Available Books* regular view captures book details, including Book_ID, Title, and Quantity, specifically for books marked as *Available* in the Library Management System. This view facilitates easy access to the current available inventory. Conversely, the *Checked Out Books* regular view provides information about books that are currently *Checked Out*,

aiding in tracking borrowed items and managing library circulation effectively. Both views enhance visibility into book statuses for streamlined library administration.

2. Materialized View

a. User Transactions

-- 3. View for user transactions
CREATE VIEW UserTransactions AS
SELECT t.Transaction_ID, t.User_ID, u.Full_Name, t.Book_ID, b.Title, t.Borrow_Date, t.Return_Date, t.Status
FROM Transaction t
JOIN User u ON t.User_ID = u.User_ID
JOIN Book b ON t.Book_ID = b.Book_ID;
select * from UserTransactions;

255 • select * From UserTransactions;
256

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Transaction_ID	User_ID	Full_Name	Book_ID	Title	Borrow_Date	Return_Date	Status
▶	1	1	John Doe	3	1984	2023-01-05	2023-01-15	Closed
	2	2	Jane Smith	5	The Catcher in the Rye	2023-02-20	2023-03-01	Closed
	3	3	Bob Johnson	9	The Hobbit	2023-03-25	2023-04-05	Closed
	4	4	Alice Williams	2	To Kill a Mockingbird	2023-04-15	2023-04-25	Closed
	5	5	Charlie Brown	8	To the Lighthouse	2023-05-10	2023-05-20	Closed
	6	6	Eva Davis	1	The Great Gatsby	2023-06-25	2023-07-05	Closed
	7	7	Frank Miller	7	Wuthering Heights	2023-07-15	2023-07-25	Closed
	8	8	Grace Wilson	6	Brave New World	2023-08-01	2023-08-11	Closed
	9	9	David Lee	4	Pride and Prejudice	2023-09-20	2023-09-30	Closed
	10	10	Sophie Turner	10	The Odyssey	2023-10-15	2023-10-25	Closed

b. User Information

-- 4. View for user Information
CREATE VIEW UserInformation AS
SELECT User_ID, Full_Name, Email, Phone, Address, Membership_Type, Join_Date
FROM User;
select * from UserInformation;

261 • select * from UserInformation;
262

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	User_ID	Full_Name	Email	Phone	Address	Membership_Type	Join_Date
▶	1	John Doe	john.doe@example.com	123-456-7890	123 Main St	Regular	2023-01-01
	2	Jane Smith	jane.smith@example.com	987-654-3210	456 Oak St	Premium	2023-02-15
	3	Bob Johnson	bob.johnson@example.com	555-123-4567	789 Pine St	Regular	2023-03-20
	4	Alice Williams	alice.williams@example.com	111-222-3333	101 Elm St	Premium	2023-04-10
	5	Charlie Brown	charlie.brown@example.com	444-555-6666	202 Birch St	Regular	2023-05-05
	6	Eva Davis	eva.davis@example.com	777-888-9999	303 Cedar St	Premium	2023-06-18
	7	Frank Miller	frank.miller@example.com	333-999-7777	404 Maple St	Regular	2023-07-30
	8	Grace Wilson	grace.wilson@example.com	666-444-1111	505 Walnut St	Premium	2023-08-22
	9	David Lee	david.lee@example.com	222-666-8888	606 Pine St	Regular	2023-09-14
	10	Sophie Turner	sophie.turner@example.com	999-111-2222	707 Oak St	Premium	2023-10-09

The *User Transactions* materialized view will consolidates transaction data, providing insights into user interactions with the library. The *User Information* view compiles essential user details. Both views will enhance data access and analytical capabilities, with the trade-off of periodic refreshes for real-time accuracy in the Library Management System.

c. More User Information.

```
-- Creating UserInformation2 View
CREATE VIEW UserInformation2 AS
SELECT
    U.User_ID,
    U.Full_Name,
    U.Email,
    U.Phone,
    U.Address,
    U.Membership_Type,
    U.Join_Date,
    B.Branch_Name,
    BH.Borrow_Date,
    BH.Return_Date
FROM
    User U
JOIN
    BorrowerHistory BH ON U.User_ID = BH.User_ID
JOIN
    BookCopy BC ON BH.Book_ID = BC.Book_ID
JOIN
    LibraryBranch B ON BC.Branch_ID = B.Branch_ID;
```

Sample Data

User_ID	Full_Name	Email	Phone	Address	Membership_Type	Join_Date	Branch_Name	Borrow_Date	Return_Date
6	Eva Davis	eva.davis@example.com	777-888-9999	303 Cedar St	Premium	2023-06-18	Main Branch	2023-06-25	2023-07-05
4	Alice Williams	alice.williams@example.com	111-222-3333	101 Elm St	Premium	2023-04-10	East Branch	2023-04-15	2023-04-25
1	John Doe	john.doe@example.com	123-456-7890	123 Main St	Regular	2023-01-01	West Branch	2023-01-05	2023-01-15
9	David Lee	david.lee@example.com	222-666-8888	606 Pine St	Regular	2023-09-14	North Branch	2023-09-20	2023-09-30
2	Jane Smith	jane.smith@example.com	987-654-3210	456 Oak St	Premium	2023-02-15	South Branch	2023-02-20	2023-03-01
8	Grace Wilson	grace.wilson@example.com	666-444-1111	505 Walnut St	Premium	2023-08-22	Main Branch	2023-08-01	2023-08-11
7	Frank Miller	frank.miller@example.com	333-999-7777	404 Maple St	Regular	2023-07-30	East Branch	2023-07-15	2023-07-25
5	Charlie Brown	charlie.brown@example.com	444-555-6666	202 Birch St	Regular	2023-05-05	West Branch	2023-05-10	2023-05-20
3	Bob Johnson	bob.johnson@example.com	555-123-4567	789 Pine St	Regular	2023-03-20	North Branch	2023-03-25	2023-04-05
10	Sophie Turner	sophie.turner@example.com	999-111-2222	707 Oak St	Premium	2023-10-09	South Branch	2023-10-15	2023-10-25

5. Implementation Notes

November 15, 2023 - Selection and Installation of DBMS

I selected and installed MySQL workbench due to its scalability, compatibility and performance. I performed more Configuration settings optimized to meet the specific requirements of the LMS.

November 20, 2023 - Schema Design and Table Relationships

I designed the schema with its respective relationship to accurately represent entities and associations in the library domain.

November 25, 2023 - Sample Data Loading

Sample data was loaded into the tables to simulate real-world scenarios and facilitate testing. Ensured that data reflected various scenarios

December 5, 2023 - View Creations

Regular and materialized views were created to enhance data retrieval and reporting capabilities.

6. Known Problems

A few flaws and anomalies with the Library Management System (LMS) installation may have an impact on how effectively the system performs. These issues must be resolved

in order to ensure that the LMS is accurate and reliable, and that it contains the appropriate entities and attributes.

The LMS has challenges due to the availability of library items and data consistency, particularly with regard to material types. Differences in availability statuses (checked out versus available) and medium (print versus digital) cause inconsistencies. It might be difficult to keep track of members' information since they may have many accounts, alter their contact information, or borrow in different ways. This is particularly true in terms of their borrowing history.

Because literature is dynamic, identifying its genres creates challenges that necessitate a flexible system that can adapt to changing literary situations. Due to the dynamic nature of the library's collections, which are characterized by new acquisitions, deletions, and changes in item formats, a system that can dynamically adapt to these changes without interfering with the integrity of earlier data is required. In addition, changes to user membership types or the development of new categories may result in inconsistent access and rights for users. The LMS must handle these fast changes in membership arrangements.

Certain assumptions led to the original classification of data categories as required during the early phases of deployment, causing difficulties in the LMS database. Further data analysis reveals variations, necessitating revisions and maybe repeats of the database structure. Because the database is iterative, with changing demands and data structures, dealing with changes between database versions involves the use of a powerful version control system.

7. Future Enhancements

The major purpose of LMS is to enhance security infrastructure with cutting-edge features such as multi-factor authentication and encryption technologies. enhancing system security to ensure the protection of critical LMS data and promoting confidence in the integrity and confidentiality of the database. Integration with digital libraries is a main goal that intends to enable seamless access to a wider range of content, such as multimedia and e-books. The LMS will work in combination with digital platforms as a sophisticated database management system (DBMS), allowing it to store and organize a variety of digital materials.

The interface will be built to meet a wide variety of user expectations and preferences in order to facilitate effective data access. To ensure equitable access for all users, including those with disabilities, accessibility components will be implemented into the DBMS. To give complete insights into user behavior, resource utilization, and general performance, the DBMS must improve its data analytics capabilities. This feature will provide administrators with valuable data to help them make strategic decisions about how to organize and use LMS data. Furthermore, given the vast majority of people who access data through mobile devices, having a seamless and responsive mobile experience will be critical.

8. Conclusion:

The objective of the Library Database Design and Implementation project was to create a strong database in order to improve the functionality and efficiency of the library system. Several significant findings and accomplishments, including requirement analysis, data modeling, database schema, implementation, and record keeping, surfaced during the course of the project life cycle.

conclusively, the project on Library Database Design and Implementation was successful in providing the library with a system that is efficient and satisfies its present demands. To guarantee that the system stays current and adaptable to changing needs, periodic upgrades, user input, and ongoing monitoring will be crucial.

9. References:

- 1.** Database Design in DBMS Tutorial: Your Ultimate Guide to Mastering Data Modeling - SQL Knowledge Center (sql-easy.com)
- 2.** Designing a Relational Database and Creating an Entity Relationship Diagram | by Craig Dickson | Towards Data Science.
- 3.** Database Design Phase 3: Implementation - MariaDB Knowledge Base
- 4.** Five Steps For Choosing and Implementing a Database | NTEN
- 5.** Database Design in DBMS Tutorial: Learn Data Modeling (guru99.com)
- 6.** Ten Common Database Design Mistakes - Simple Talk (red-gate.com)
- 7.** The Different Steps in Database Development Life Cycle | by Laura Malovich | Medium
- 8.** Learn SQL: CREATE DATABASE & CREATE TABLE Operations (sqlshack.com)