Brainwave Matrix Solutions

Internship Task – Abhilash Antony

Introduction

The Library Management System (LMS) is designed to facilitate the efficient management of library resources, enhancing operations for both librarians and patrons. This system addresses key functionalities such as book cataloging, member management, loan processing, and staff administration.

By utilizing a relational database, the LMS allows for organized storage and easy retrieval of data, enabling librarians to track book availability, manage member information, and oversee loan transactions effectively. The system comprises several interconnected entities, including Books, Members, Staff, Loans, and Categories, each with specific attributes and relationships.

This report outlines the database schema, implementation details, and sample data entries for the LMS, highlighting its role in improving library management processes and service delivery.

Database Design

The database design for the Library Management System (LMS) focuses on creating an organized structure that effectively manages the relationships between various entities involved in library operations. This design comprises an Entity-Relationship Diagram (ERD) and detailed table structures for each entity.

Entity-Relationship Diagram (ERD)

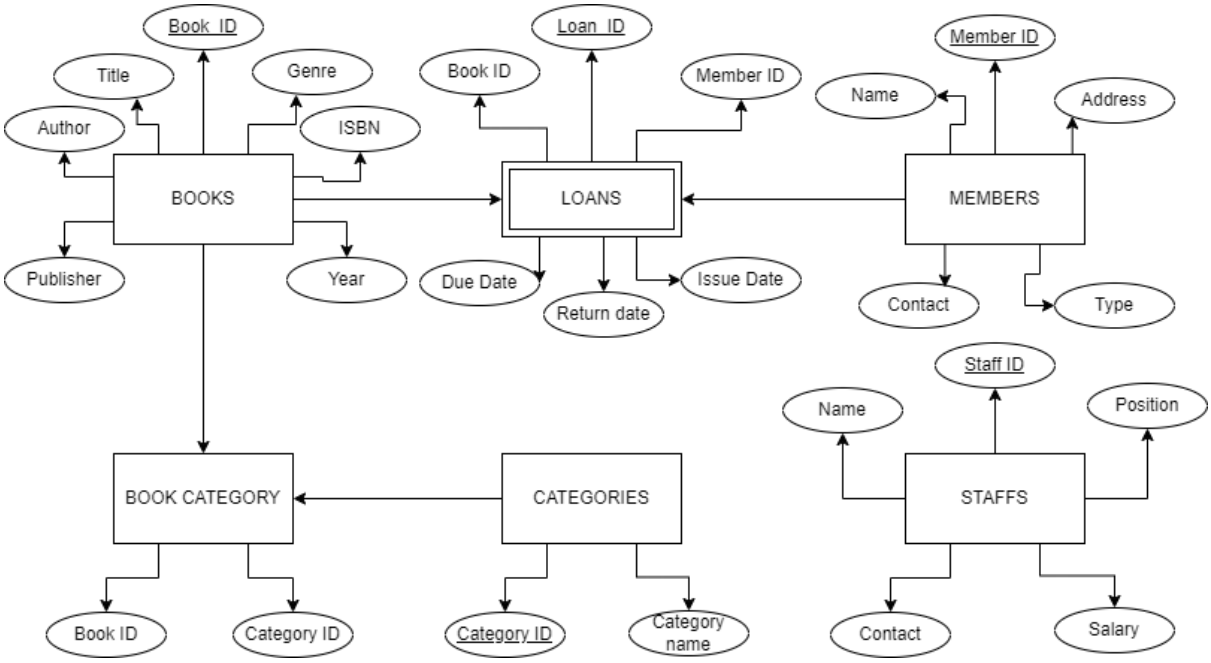

Table Structures:

- **Books Table**: `Book_ID`, `Title`, `Author`, `Genre`, `ISBN`, `Publisher`, `Year_Published`, `Available`
- **Members Table**: `Member_ID`, `Name`, `Address`, `Contact_Info`, `Membership_Type`
- **Staff Table**: `Staff_ID`, `Name`, `Position`, `Contact_Info`, `Salary`
- **Loans Table**: `Loan_ID`, `Book_ID`, `Member_ID`, `Issue_Date`, `Due_Date`, `Return_Date`
- **Categories Table**: `Category_ID`, `Category_Name`
- **Book_Category Table**: `Book_ID`, `Category_ID`

This structured approach ensures that all relevant data is captured and organized, facilitating efficient operations within the library and improving overall service delivery.

```sql
-- create DB
DROP DATABASE ReadNet;
CREATE DATABASE ReadNet;
USE ReadNet;

-- create tables
CREATE TABLE Books (
    Book_ID INT PRIMARY KEY AUTO_INCREMENT,
    Title VARCHAR(255) NOT NULL,
    Author VARCHAR(255),
    Genre VARCHAR(100),
    ISBN VARCHAR(20) UNIQUE,
    Publisher VARCHAR(100),
    Year_Published YEAR,
    Available BOOLEAN DEFAULT TRUE
);

CREATE TABLE Members (
    Member_ID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    Address VARCHAR(255),
    Contact_Info VARCHAR(50),
    Membership_Type ENUM('Regular', 'Premium') DEFAULT 'Regular',
    Join_Date DATE
);

CREATE TABLE Categories (
    Category_ID INT PRIMARY KEY AUTO_INCREMENT,
    Category_Name VARCHAR(100) UNIQUE NOT NULL
);

CREATE TABLE Loans (
    Loan_ID INT PRIMARY KEY AUTO_INCREMENT,
    Book_ID INT,
    Member_ID INT,
    Issue_Date DATE,
```

```
    Due_Date DATE,
    Return_Date DATE,
    FOREIGN KEY (Book_ID) REFERENCES Books(Book_ID),
    FOREIGN KEY (Member_ID) REFERENCES Members(Member_ID)
);

CREATE TABLE Staff (
    Staff_ID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    Position VARCHAR(100),
    Contact_Info VARCHAR(50),
    Salary DECIMAL(10, 2)  -- Assuming salary can have up to 10 digits, with 2
decimal places
);

CREATE TABLE Book_Category (
    Book_ID INT,
    Category_ID INT,
    PRIMARY KEY (Book_ID, Category_ID),
    FOREIGN KEY (Book_ID) REFERENCES Books(Book_ID),
    FOREIGN KEY (Category_ID) REFERENCES Categories(Category_ID)
);
```

Data Insertion

In this section, I provide the SQL commands used for inserting sample data into the tables of the Library Management System (LMS). This data has been carefully selected to ensure a diverse representation across different categories, members, and staff.

```
INSERT INTO Books (Title, Author, Genre, ISBN, Publisher, Year_Published,
Available) VALUES
('The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction', '9780743273565',
'Scribner', 1925, TRUE),
('1984', 'George Orwell', 'Dystopian', '9780451524935', 'Houghton Mifflin',
1949, TRUE),
('To Kill a Mockingbird', 'Harper Lee', 'Fiction', '9780061120084', 'J.B.
Lippincott & Co.', 1960, TRUE),
('Brave New World', 'Aldous Huxley', 'Dystopian', '9780060850524', 'Chatto &
Windus', 1932, TRUE),
('Moby Dick', 'Herman Melville', 'Fiction', '9780142437247', 'Harper &
Brothers', 1932, TRUE),
('The Catcher in the Rye', 'J.D. Salinger', 'Fiction', '9780316769488',
'Little Brown and Company', 1951, TRUE),
('Fahrenheit 451', 'Ray Bradbury', 'Dystopian', '9781451673319', 'Ballantine
Books', 1953, TRUE),
('Pride and Prejudice', 'Jane Austen', 'Fiction', '9780679783268', 'T.
Egerton', 1912, TRUE),
('The Odyssey', 'Homer', 'Epic', '9780140268867', 'Penguin Classics', 1950,
TRUE),
```

```
('The Iliad', 'Homer', 'Epic', '9780140447941', 'Penguin Classics', 1913,
TRUE);

INSERT INTO Members (Name, Address, Contact_Info, Membership_Type) VALUES
('Rajesh Kumar', '12 MG Road, Bangalore', 'rajesh.kumar@gmail.com',
'Premium'),
('Sneha Verma', '45 Jayanagar, Bangalore', 'sneha.verma@gmail.com',
'Regular'),
('Anjali Sharma', '89 Connaught Place, Delhi', 'anjali.sharma@gmail.com',
'Regular'),
('Amit Patel', '25 Ashok Nagar, Ahmedabad', 'amit.patel@gmail.com',
'Premium'),
('Neha Singh', '67 Gandhi Marg, Lucknow', 'neha.singh@gmail.com', 'Regular'),
('Vikram Reddy', '108 Jubilee Hills, Hyderabad', 'vikram.reddy@gmail.com',
'Premium'),
('Priya Nair', '32 Marine Drive, Mumbai', 'priya.nair@gmail.com', 'Regular'),
('Rohan Desai', '78 Ellis Bridge, Ahmedabad', 'rohan.desai@gmail.com',
'Premium'),
('Aishwarya Iyer', '23 T Nagar, Chennai', 'aishwarya.iyer@gmail.com',
'Regular'),
('Suresh Pillai', '56 Panampilly Nagar, Kochi', 'suresh.pillai@gmail.com',
'Premium');

INSERT INTO Categories (Category_Name) VALUES
('Fiction'), ('Dystopian'), ('Science Fiction'),
('Biography'),('Fantasy'),('History'),('Epic'),
('Philosophy'),('Drama'),('Mystery');

INSERT INTO Loans (Book_ID, Member_ID, Issue_Date, Due_Date, Return_Date)
VALUES
(1, 1, '2024-10-15', '2024-10-25', NULL),
(2, 2, '2024-10-16', '2024-10-30', '2024-10-29'),
(3, 3, '2024-10-17', '2024-10-28', NULL),
(4, 4, '2024-10-18', '2024-11-01', NULL),
(5, 5, '2024-10-19', '2024-11-03', '2024-11-01'),
(6, 6, '2024-10-20', '2024-10-29', NULL),
(7, 7, '2024-10-21', '2024-11-05', NULL),
(8, 8, '2024-10-22', '2024-11-07', '2024-11-06'),
(9, 9, '2024-10-23', '2024-11-10', NULL),
(10, 10, '2024-10-24', '2024-11-12', '2024-11-10');

INSERT INTO Book_Category (Book_ID, Category_ID)VALUES
(1, 1),  -- 'The Great Gatsby' in Fiction
(2, 2),  -- '1984' in Dystopian
(2, 3),  -- '1984' also in Science Fiction
(3, 1),  -- 'To Kill a Mockingbird' in Fiction
(4, 2),  -- 'Brave New World' in Dystopian
(5, 1),  -- 'Moby Dick' in Fiction
```

```
(6, 1),  -- 'The Catcher in the Rye' in Fiction
(7, 2),  -- 'Fahrenheit 451' in Dystopian
(8, 1),  -- 'Pride and Prejudice' in Fiction
(9, 7),  -- 'The Odyssey' in Epic
(10, 7); -- 'The Iliad' in Epic

INSERT INTO Staff (Name, Position, Contact_Info, Salary) VALUES
('Ravi Sharma', 'Librarian', 'ravi.sharma@readnest.in', 50000),
('Anita Desai', 'Assistant Librarian', 'anita.desai@readnest.in', 40000),
('Vikram Singh', 'Library Manager', 'vikram.singh@readnest.in', 60000),
('Neha Gupta', 'Library Assistant', 'neha.gupta@readnest.in', 35000),
('Priya Patel', 'Cataloger', 'priya.patel@readnest.in', 45000),
('Suresh Nair', 'Archivist', 'suresh.nair@readnest.in', 47000),
('Aditi Verma', 'Library Technician', 'aditi.verma@readnest.in', 43000),
('Rahul Mehta', 'Library Assistant', 'rahul.mehta@readnest.in', 37000),
('Kiran Rao', 'IT Specialist', 'kiran.rao@readnest.in', 55000),
('Simran Kaur', 'Acquisitions Librarian', 'simran.kaur@readnest.in', 52000);
```

Notes on the Data

- **Diversity of Books**: The `Books` table contains a mix of genres, including fiction, dystopian, and epic literature, catering to various reader preferences.
- **Variety of Members**: The `Members` table includes individuals from different regions in India, reflecting a wide range of membership types (Premium and Regular).
- **Staff Representation**: The `Staff` table comprises various positions within the library, ensuring a well-rounded team to support library operations.
- **Loan Transactions**: The `Loans` table tracks issued books, their due dates, and return dates, facilitating effective management of library resources.

Querying the Database

This section presents SQL queries utilized to interact with the Library Management System's database, focusing on various aspects such as book availability, member borrowing status, and inventory management.

1. **List All Books and Their Availability**: The query retrieves the titles, authors, and availability status of all books, providing an overview of the library's collection.

```
SELECT Title, Author, Available FROM Books;
```

Title	Author	Available
The Great Gatsby	F. Scott Fitzgerald	1
1984	George Orwell	1
To Kill a Mockingbird	Harper Lee	1
Brave New World	Aldous Huxley	1
Moby Dick	Herman Melville	1
The Catcher in the Rye	J.D. Salinger	1
Fahrenheit 451	Ray Bradbury	1
Pride and Prejudice	Jane Austen	1
The Odyssey	Homer	1
The Iliad	Homer	1

2. **Identify Members with Unreturned Books**: This query identifies members who have borrowed books but have not returned them, displaying the member's name along with the book title, issue date, and due date.

```
SELECT Members.Name, Books.Title, Loans.Issue_Date, Loans.Due_Date FROM Members
JOIN Loans ON Members.Member_ID = Loans.Member_ID
JOIN Books ON Loans.Book_ID = Books.Book_ID
WHERE Loans.Return_Date IS NULL;
```

Name	Title	Issue_Date	Due_Date
Anjali Sharma	To Kill a Mockingbird	2024-10-17	2024-10-28
Amit Patel	Brave New World	2024-10-18	2024-11-01
Vikram Reddy	The Catcher in the Rye	2024-10-20	2024-10-29
Priya Nair	Fahrenheit 451	2024-10-21	2024-11-05
Aishwarya Iyer	The Odyssey	2024-10-23	2024-11-10

3. **Filter Books by Category**: This query lists books belonging to a specific category (e.g., Fiction), allowing users to focus on particular genres.

```
SELECT Books.Title, Books.Author FROM Books
JOIN Book_Category ON Books.Book_ID = Book_Category.Book_ID
JOIN Categories ON Book_Category.Category_ID = Categories.Category_ID
WHERE Categories.Category_Name = 'Fiction';
```

Title	Author
The Great Gatsby	F. Scott Fitzgerald
To Kill a Mockingbird	Harper Lee
Moby Dick	Herman Melville
The Catcher in the Rye	J.D. Salinger
Pride and Prejudice	Jane Austen

4. **Count Borrowed Books per Member**: This query counts the total number of books borrowed by each member, helping to track borrowing patterns and library engagement.

```
SELECT Members.Name, COUNT(Loans.Loan_ID) AS Total_Borrowed FROM Members
LEFT JOIN Loans ON Members.Member_ID = Loans.Member_ID
GROUP BY Members.Member_ID;
```

Name	Total_Borrowed
Rajesh Kumar	1
Sneha Verma	1
Anjali Sharma	1
Amit Patel	1
Neha Singh	1
Vikram Reddy	1
Priya Nair	1
Rohan Desai	1
Aishwarya Iyer	1
Suresh Pillai	1

5. **Update Book Availability Upon Return**: The first update query marks a book as available after its return, while the second updates the loan record with the return date, ensuring accurate inventory management.

```
UPDATE Books SET Available = TRUE WHERE Book_ID = 1;
UPDATE Loans SET Return_Date = CURDATE() WHERE Loan_ID = 1;
-- returned books
SELECT Book_ID, Member_ID, Return_Date FROM Loans WHERE Return_Date IS NOT NULL;
```

Book_ID	Member_ID	Return_Date
1	1	2024-10-10
2	2	2024-10-29
5	5	2024-11-01
8	8	2024-11-06
10	10	2024-11-10

Conclusion

In conclusion, the Library Management System (LMS) provides an organized and efficient way to manage library resources. By using a relational database, the LMS makes it easy for librarians to keep track of books, members, and loans. The system includes important features like book cataloguing, member management, loan processing, and staff administration, all of which help improve the library's operations.

The database design, including the Entity-Relationship Diagram (ERD) and table structures, ensures that all relevant data is stored and organized effectively. The sample data inserted into the tables reflects a diverse selection of books, members, and staff, which enhances the library's services.

Overall, the LMS plays a crucial role in streamlining library management processes, making it easier for librarians to serve their patrons and maintain an up-to-date inventory of resources. With this system in place, libraries can offer better services to their members and foster a love for reading and learning.