

Library Management system

By: Yasmin Abdi, Divine Mobote
ITCS 3160-091

Project overview

The Library Management System database was created to be able to manage book inventories, member records, and loan transactions in a library. It accurately tracks borrowed books and due dates in a simple way for library staff to utilize.

Objectives

- To maintain a centralized system for managing library books, members, and loan transactions.
- To track book availability and due dates to prevent conflicts or overdue penalties.
- To streamline administrative tasks, including member registrations and overdue tracking.

Requirement Gathering

Data:

- Members:

Attributes: Member_ID, Name, Email, Phone, Address, Membership_Status

- Books:

Attributes: Book_ID, Title, Author, Genre, ISBN, Book_Status

- Staff:

Attributes: Staff_ID, Name, Role, Phone, Email

- Borrow Records:

Attributes: Borrow_ID, Member_ID, Book_ID, Staff_ID, Borrow_Date, Return_Date, Fine.

Requirement Gathering

Function:

- Members can register, view and update their profile.
- Books can be searched by title, author or genre.
- Staff can issue books, check returns, and manage overdue penalties.
- The system automatically calculates fines based on return dates.
- Real-time tracking of book availability

Security:

- Member data should be accessible only to authorized users.
- Sensitive modification must be restricted to staff, allowing only authorized users.
- Backup and Recovery for database integrity.

Staff: Represents the library staff who manage records and transactions.

Staff	
Staff_ID	Pk Int
Name	Varchar
Role	Varchar
Phone	Varchar
Email	Varchar

Book: Represents the books available in the library.

Books	
Book_ID	Pk Int
Title	Varchar
Author	Varchar
Genre	Varchar
ISBN	Varchar
Book_Status	Varchar

Member: Represents the library's users.

Members	
Member_ID	Pk Int
Name	Varchar
Email	Varchar
Phone	Varchar
Address	Varchar
Membership_Status	Varchar

Conceptual Design

Staff → BorrowRecords:

A staff member manages multiple borrow records.

Members → BorrowRecords:

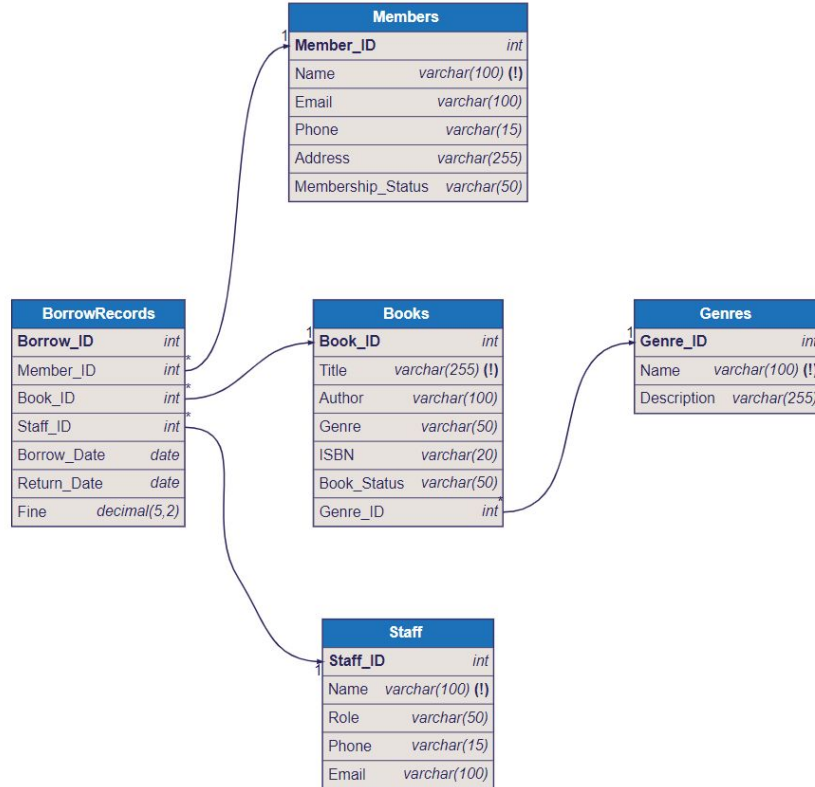
A member can borrow multiple books.

BorrowRecords	
Borrow_ID	Pk Int
Member_ID	Fk Int
Book_ID	FK Int
Staff_ID	Fk Int
Borrow_Date	Date
Return_Date	Date
Fine	Int

Books → BorrowRecords:

A book can be borrowed multiple times by different members. Book information appears in BorrowRecords

Relational Model



Physical Model - CRUD Operations

Query Doc

Create: Adding a new member

```
INSERT INTO Member (Member_ID, Name, Email, Phone,  
Address, Membership_Status)  
VALUES (12, 'Zara Zane', 'ZaraZane@example.com',  
'555-555-1234', '999 ZZ Ave', 'Active');
```

1 row(s) inserted.

Read: Retrieve all books that are available

```
SELECT *  
FROM Book  
WHERE Book_Status = 'Available';
```

BOOK_ID	TITLE	AUTHOR	GENRE	ISBN	BOOK_STATUS
1	To Kill a Mockingbird	Harper Lee	Fiction	1234567890	Available
3	The Great Gatsby	F. Scott Fitzgerald	Classics	1122334455	Available
4	Pride and Prejudice	Jane Austen	Romance	2233445566	Available
6	Wuthering Heights	Emily Brontë	Gothic	4455667788	Available
7	The Catcher in the Rye	J.D. Salinger	Fiction	5566778899	Available
9	Fahrenheit 451	Ray Bradbury	Dystopian	7788990011	Available

Update: Modify membership status

```
UPDATE Member  
SET Membership_Status = 'Inactive'  
WHERE Member_ID = 8;
```

1 row(s) updated.


Delete: Remove a member

```
DELETE FROM Member  
WHERE Member_ID = 11;
```

1 row(s) deleted.

Physical Model - Complex Query

```
SELECT
  m.Name AS Member_Name,
  COUNT(br.Borrow_ID) AS Total_Books_Borrowed,
  SUM(br.Fine) AS Total_Fine
FROM
  Member m
JOIN
  BorrowRecord br ON m.Member_ID = br.Member_ID
JOIN
  Book b ON br.Book_ID = b.Book_ID
GROUP BY
  m.Member_ID, m.Name
ORDER BY
  Total_Books_Borrowed DESC;
```



MEMBER_NAME	TOTAL_BOOKS_BORROWED	TOTAL_FINE
Hannah Horse	1	1
Ian Irving	1	0
Apple Alice	1	0
Eva Evans	1	5
FiFi Foster	1	0
David Darnell	1	0
George Gray	1	0
Bailey Brown	1	2
Jack Jones	1	3
Charlie Clark	1	0

Testing and Validation

Insert Queries Doc

```
SELECT m.Name AS Member_Name,  
       SUM(br.Fine) AS Total_Fine  
FROM Member m  
JOIN BorrowRecord br ON m.Member_ID =  
br.Member_ID  
GROUP BY  
m.Name;
```

AGGREGATE QUERY

```
SELECT br.Borrow_ID,  
       m.Name AS Member_Name,  
       b.Title AS Book_Title,  
       s.Name AS Staff_Name,  
       br.Borrow_Date,  
       br.Return_Date  
FROM BorrowRecord br  
JOIN Member m ON br.Member_ID =  
m.Member_ID  
JOIN Book b ON br.Book_ID = b.Book_ID  
JOIN Staff s ON br.Staff_ID = s.Staff_ID;
```

JOIN QUERY

BORROW_ID	MEMBER_NAME	BOOK_TITLE	STAFF_NAME	BORROW_DATE	RETURN_DATE
1	Apple Alice	To Kill a Mockingbird	Clake Cali	01-NOV-24	14-NOV-24
2	Bailey Brown	1984	Dalen Daves	05-NOV-24	19-NOV-24
3	Charlie Clark	The Great Gatsby	Emma Evans	06-NOV-24	20-NOV-24
4	David Darnell	Pride and Prejudice	Frank Foster	07-NOV-24	21-NOV-24
5	Eva Evans	Moby Dick	Gina Green	08-NOV-24	22-NOV-24
6	Fifi Foster	Wuthering Heights	Hank Hill	09-NOV-24	23-NOV-24
7	George Gray	The Catcher in the Rye	Ivy Ives	10-NOV-24	24-NOV-24
8	Hannah Horse	Brave New World	Jack Jones	11-NOV-24	25-NOV-24
9	Ian Irving	Fahrenheit 451	Karen Kline	12-NOV-24	26-NOV-24
10	Jack Jones	The Hobbit	Liam Lane	13-NOV-24	27-NOV-24

MEMBER_NAME	TOTAL_FINE
Eva Evans	5
George Gray	0
Hannah Horse	1
Ian Irving	0
Apple Alice	0
Charlie Clark	0
David Darnell	0
Jack Jones	3
Fifi Foster	0
Bailey Brown	2

Conclusion

- Project summary:
 - Designed a relational database for managing library operations, from member management to book inventory.
- Key Features:
 - Managed member information.
 - Tracked book inventory.
 - Staff management.
 - Automated borrowing/returns.
 - Enable complex queries for efficient report.

