Database Requirements

Introduction

Project Overview

The Library Management System (LMS) will manage a small library's diverse collection of loanable items (books, digital media, magazines). The system enforces borrowing rules based on membership types (each with distinct borrowing limits and fee structures), tracks loan transactions and reservations, and generates a variety of reports to inform library operations and decision-making.

The primary goal is to offer hands-on experience in designing, implementing, and managing a relational database: from conceptual and logical design to physical implementation, data population, and demonstration through meaningful queries and reports.

Scope

- **Data Management:** items such as books, digital media, and magazines; clients and memberships; and transactions such as reservations
- **Constraints:** borrowing limits, fee calculations, and special item restrictions
- **User Interfaces:** separate interfaces for both librarian staff and clients, allowing them to perform various necessary transactions
    - Clients: browse, checkout, returns, reservations, manage fees
    - Staff: process checkouts and returns, manage clients, manage catalog
- **Feedback:** reports for librarian staff for item trends and client activity; notifications for clients on due dates and reservations
- **Software Stack:** MySQL or MariaDB based, version control with git and GitHub
- **Exclusions:** external integration beyond user interfaces, advanced or custom analytics besides described reports

Glossary

- **Attribute:** A specific characteristic of an entity (e.g., ISBN, Title).
- **Entity:** A database representation of a real-world object (e.g., Book, Client).
- **DBMS:** Database Management System.
- **Foreign Key (FK):** An attribute linking one entity to another.
- **IT:** Information Technology.
- **ISBN:** International Standard Book Number.
- **Primary Key (PK):** A unique identifier for an entity record.

- **Supertype/Subtype:** A technique wherein a general “Item” supertype (with shared attributes) can have specialized subtypes (e.g., Book, DigitalMedia, Magazine).

Stakeholders

1. **Library Clients:**
   - Browse, borrow, reserve, and return items
   - View borrowing history
   - Manage fees, account balance, and payments
2. **Library Staff:**
   - Manage the catalog (add/update/remove items such as books, digital items, and magazines)
   - Manage client checkouts, returns, reservations, and client memberships.
   - Generate queries and reports for decision-making
3. **Library Management:**
   - Review reports on usage, fees, and operational metrics
   - Adjust library policies (borrowing limits, fees) based on data insights
4. **IT / Database Administrators:**
   - Maintain the DBMS environment, including data integrity and validity
   - Ensure reliable and performant operation of the database

Requirements

Functional Requirements

1. User Administration

Authentication and Authorization

- Secure login for both library staff and clients.
- Role-based access control to restrict sensitive operations (e.g., modifying membership types or removing items).

User Profile Management

- Ability to add, update, or deactivate client and staff accounts.

2. Data Entry and Management

Item Management

- Create, update, and delete records for books, digital media, and magazines.

- Maintain item attributes: ISBN, title, author/creator, publication year, genre, availability status, etc.
- Automatically update an item's availability (available, borrowed, or reserved) when a loan or reservation occurs.

Membership Management

- Define and manage different membership types (e.g., regular, student, senior) with borrowing limits and fee structures.

3. Loan and Reservation Processing

Borrowing Transactions

- Record new loans with details: client ID, item ID, borrow date, due date, and automatic late fee calculation if overdue.
- Enforce borrowing limits based on membership type.

Returns Processing

- Update item records upon return, including calculating and recording applicable late fees.

Reservation Management

- Allow clients to reserve items currently on loan.
- A system to mark items as "available" for pick-up when returned, notifying the reserving client.
- Support reservation cancellation by either staff or the client.

4. Data Retrieval and Reporting

Standard Queries

- Retrieve books by specific author or publication year.
- List items (books, digital media, magazines) available in a given genre.
- Check a client's membership status and account details.

Custom Reports

- Overdue report: All clients with overdue items and calculated fines.
- Borrowing trends: Popular authors, genres, items.
- Monthly fee collection summary and borrowing statistics.

Additional Query Ideas

- Clients who never returned items late.

- Average borrowing time per membership type.
- Items not borrowed in the past 6 months.
- Identifying peak borrowing hours/days for staffing optimization.

Data Entities

1. **Client**
   - ClientID (INT, PK, AUTO_INCREMENT)
   - Name (VARCHAR, NOT NULL)
   - ContactInfo (VARCHAR)
     i. PhoneNumber (VARCHAR)
     ii. Email (VARCHAR)
   - MembershipType (VARCHAR, FK → Membership(MembershipType))
   - AccountStatus (ENUM: 'active', 'suspended', 'closed')
2. **Membership**
   - MembershipType (VARCHAR, PK)
   - Description (VARCHAR)
   - BorrowingLimit (INT, NOT NULL)
   - FeeStructure (DECIMAL(5,2), NOT NULL)
3. **Book**
   - ISBN (VARCHAR, PK)
   - Title (VARCHAR, NOT NULL)
   - Author (VARCHAR, NOT NULL)
   - PublicationYear (YEAR)
   - Genre (VARCHAR)
   - AvailabilityStatus (ENUM: 'available', 'borrowed', 'reserved')
4. **DigitalMedia**
   - MediaID (INT, PK, AUTO_INCREMENT)
   - Title (VARCHAR, NOT NULL)
   - Creator (VARCHAR)
   - PublicationYear (YEAR)
   - Format (ENUM: 'DVD', 'Blu-ray', 'Digital')
   - AvailabilityStatus (ENUM: 'available', 'borrowed', 'reserved')
5. **Magazine**
   - MagazineID (INT, PK, AUTO_INCREMENT)
   - Title (VARCHAR, NOT NULL)
   - IssueNumber (INT, NOT NULL)
   - PublicationDate (DATE, NOT NULL)

   - AvailabilityStatus (ENUM: 'available', 'borrowed', 'reserved')

6. **LoanTransaction**
   - TransactionID (INT, PK, AUTO_INCREMENT)
   - ClientID (INT, FK → Client(ClientID))
   - ItemID (VARCHAR or INT - references Book/MediaID/MagazineID)
   - ItemType (ENUM: 'Book', 'DigitalMedia', 'Magazine')
   - BorrowDate (DATE, NOT NULL)
   - DueDate (DATE, NOT NULL)
   - ReturnDate (DATE, NULL)
   - CalculatedFine (DECIMAL(5,2), DEFAULT 0, CHECK (CalculatedFine >= 0))
7. **Reservation**
   - ReservationID (INT, PK, AUTO_INCREMENT)
   - ClientID (INT, FK → Client(ClientID))
   - ItemID (VARCHAR or INT - references Book/MediaID/MagazineID)
   - ItemType (ENUM: 'Book', 'DigitalMedia', 'Magazine')
   - ReservationDate (DATE, NOT NULL)
   - Status (ENUM: 'active', 'cancelled', 'fulfilled')

Non-Functional Requirements

1. **Performance:**
   - Support many concurrent library client and staff transactions
   - Ensure smooth operation as library catalog and management databases grow
2. **Integrity:**
   - Maintain database integrity between transactions
   - Allow creation and restoration from backups
3. **Security:**
   - Restrict unauthorized access to out of scope database sections for users
   - Ensure authorization before performing any administrative transactions
   - Restrict unauthorized access from third parties to the library
4. **Regulatory Compliance:**
   - Comply to any local privacy laws and guidelines
   - Retain rights of library clients to request and remove personal data

Hardware and Software Requirements

Hardware Requirements

- **Server:**
  - Minimum 2 CPU cores, 8 GB RAM, 250 GB storage.
  - Stable network connectivity for remote client access.
- **Client Machines:**
  - Standard desktop/laptop capable of running modern web browsers or SQL client tools.

Software Requirements

- **Database Server:**
  - MariaDB or MySQL on a Linux server (or a cloud-hosted environment).
- **Operating System:**
  - Linux (server-side) preferred; Windows or macOS acceptable for client-side development.
- **Development Tools:**
  - SQL tools.
  - Git version control (hosted on GitHub).
- **Web Technologies (If Applicable):**
  - HTML, CSS, JavaScript for a basic web front-end interface (for both staff and client portals).

Appendices

Appendix: Design Choices and Assumptions

- **Item Modeling:**
  We have modeled Books, DigitalMedia, and Magazines as separate entities to capture their unique attributes. Alternatively, an “Item” supertype with subtypes could be used for a more unified approach.
- **LoanTransaction and Reservation Linking:**
  Both LoanTransaction and Reservation use a composite reference (ItemID and ItemType) to indicate which item is involved. This design supports multiple item types without requiring separate transaction tables.
- **Business Rules:**
  - Clients cannot exceed their borrowing limits as defined by their Membership.
  - Late fees are automatically calculated in LoanTransaction based on the difference between DueDate and ReturnDate.
  - The system enforces that only available items (with AvailabilityStatus = 'available') can be loaned or reserved.