

Database Requirements

Project Overview:

The intended use of this database is to manage library memberships as well as serve as a working inventory for said library. This includes but is not limited to, adding a member to the library membership, finding details(e.g., SBN, year, author, genre, ...) about books and other media types, checking out various media types to members, late fees incurred, and determining the availability of a particular book or other media type.

Scope:

The database will store a series of books and other media being held at a non-specified library, along with any relevant information, and allow for the addition, and deletion, along with any needed updates for the database. The database will also house information regarding borrowers and members of the library, including age, education, and any other relevant information. The database will keep track of members and their items checked out as well as any fees they incur. It will also keep track of the library inventory, which includes the availability of items stored in the library.

Glossary:

SBN: standard book number

Stakeholders:

- Library staff
 - Regular employees
 - Managers
 - Administrators
- Clients
 - Membership holding users
 - Guest users
- Library Board (who oversees funding for the system)
- IT support/developers (who might work and troubleshoot the database)

Functional Requirements:

1. User Administration

- Add, update, or delete library staff and members
- Assign user roles (administrator, client)
- Manage different membership types (regular, student, senior citizen)

2. Data Entry and Management

- Add new books, digital media, and magazines
- Update book info
- Remove items from database

3. Data Retrieval

- Search for books and digital media by title, author, ISBN, genre, other fields
- View magazines by title, issue number, or publication date
- Get client info based on unique ID, name, or membership type

4. Borrowing and Returning Transactions

- Check out items based on user max borrowing limits
 - Depends on membership type
- Track borrowing timestamps and due dates
 - Track item returns and update availability status
- Calculate late fees based on membership type and time elapsed past due date

5. Reservations and Notifications

- Allow users to place reservations for checked-out books
- Notify users when reserved items become available
- Send reminders for upcoming due dates and overdue notices

6. Report Generation

- Generate reports on:
 - Book availability and circulation trends
 - Overdue items and fine calculations
 - Client activity
 - Popular books and media items

Data Entities

1. Books

- book_id (INT, Primary Key, Auto-increment)
- title (VARCHAR(255), Not Null)
- author (VARCHAR(255), Not Null)
- isbn (VARCHAR(20), Unique, Not Null)
- publication_year (INT, Check > 1000)
- genre (VARCHAR(100))
- availability_status (ENUM('Available', 'Checked Out', 'Reserved'))

2. Digital Media

- media_id (INT, Primary Key, Auto-increment)
- title (VARCHAR(255), Not Null)
- creator (VARCHAR(255))
- media_type (ENUM('E-book', 'Audiobook', 'Video'))
- publication_year (INT, Check > 1000)
- availability_status (ENUM('Available', 'Checked Out', 'Reserved'))

3. Magazines

- magazine_id (INT, Primary Key, Auto-increment)
- title (VARCHAR(255), Not Null)
- issue_number (VARCHAR(50), Not Null)
- publication_date (DATE, Not Null)
- availability_status (ENUM('Available', 'Checked Out'))

4. Clients

- client_id (INT, Primary Key, Auto-increment)
- name (VARCHAR(255), Not Null)
- contact_info (VARCHAR(255))
- membership_type (ENUM('Regular', 'Student', 'Senior Citizen'))
- account_status (ENUM('Active', 'Suspended'))
- max_books (INT, Nullable)

5. Borrowing Transactions

- transaction_id (INT, Primary Key, Auto-increment)
- client_id (INT, Foreign Key → Clients)
- item_id (INT, Foreign Key → Books/Digital Media/Magazines)
- borrow_date (DATETIME, Not Null)
- due_date (DATETIME, Not Null)
- return_date (DATETIME, Nullable)
- overdue_fine (DECIMAL(5,2), Default 0)

6. Reservations

- reservation_id (INT, Primary Key, Auto-increment)
- client_id (INT, Foreign Key → Clients)
- item_id (INT, Foreign Key → Books/Digital Media)
- reservation_date (DATETIME, Not Null)
- status (ENUM('Pending', 'Available', 'Cancelled'))

Hardware Requirements:

The library database system will be hosted on a standalone server running MariaDB on EECS servers, requiring robust hardware components to ensure reliability, efficiency, and scalability. The server should be equipped with multi-core processors (such as Intel Xeon) to ensure smooth concurrent access and processing of queries from multiple users (clients). The servers will have a substantial amount of RAM (32-64GB) to efficiently manage large datasets and complex queries related to media attributes, membership details, and borrowing activities. The system will require a high-capacity hard drive or solid-state drive (SSD) with at least 1TB of storage for fast data access and to accommodate the growing library data over time, including records of books, digital media, magazines, clients, and borrowing transactions. A dedicated network interface card (NIC) will handle network traffic efficiently, ensuring that data is transferred without interruption between the database and clients. The system will also include an uninterrupted power supply to safeguard against power failures and prevent data corruption. Regular backups will be scheduled on separate storage cloud-based (Google Cloud storage) for data integrity and recovery.

Software Requirements:

We plan to use MariaDB for our database, so MariaDB will of course need to be installed on the server that hosts the database. We will also need to use a command line interface to interact with the database (i.e. create tables, add/remove/alter data). Later on, we will also need to design user interfaces for both clients as well as library staff. We have not officially decided on a programming language to create the user interface, but the code will need to be stored in the same place as the database so that the program can easily query the database.

Project Meeting Log:

Date: February 27, 2025

Time: 4:00 PM

Location: Virtual via Discord

Objective: Document the purpose, scope, and requirements for our project

Team Members Present: Evans, Brisa, Jared, Spencer, John

Task Completion Confirmation:

- Spencer: Brainstormed and documented all possible stakeholders (complete)
- John: Documented software requirements and brainstormed functional requirements with the team (complete).

- Evans: Documented Hardware requirements for our library database system (complete)
- Jared: Purpose statement in introduction (complete)
- Brisa: Documented scope of the database and brainstormed functional requirements (complete)

Brainstorming Session:

- Discussed the purpose and scope of our project.
- We discussed which database we wanted to use.
- Discussed whether to host the database on EECS cycle servers or a cloud server.
- Determined the requirements for our project (functional, non-functional, hardware, and software).

Tasks Allocated:

- Spencer: Brainstorm key database entities to include in the ER model and collaborate with the team to design the model.
- John: Brainstorm key database entities to include in the ER model and collaborate with the team to design the model.
- Evans: Brainstorm key database entities to include in the ER model and collaborate with the team to design the model.
- Jared: Brainstorm key database entities to include in the ER model and collaborate with the team to design the model.
- Brisa: Brainstorm key database entities to include in the ER model and collaborate with the team to design the model.

Follow-up Actions:

Schedule the next meeting for Friday, March 7, 2025, at 6:00 PM to create our ER diagram.