# Database Requirements

The DataBasers™

# 1. Introduction

## 1.1 Project Overview

The objective of this project is to design and implement a relational database to be used by the staff and clientele of a small public library. It is meant to aid the daily needs of both library staff and clients, providing clients with a convenient source of information on the library's collection and staff with the means to record and track media loans, borrowing trends, and more.

## 1.2 Scope (Project Boundaries)

This project will include a Project Plan, which summarizes the project objectives and team organization; a Database Requirements document, which indicates the project requirements and preliminary design components; a Conceptual Modeling document, which details the complete conceptual design of the database, including an ER diagram; a Logical Relational Model document, which details the relational schema of the database; and a SQL-based Physical Database Design, which will be represented in a document displaying the process, testing, and results of the database implementation. The project will culminate in a full demonstration of the capabilities of the finished database.

## 1.3 Glossary

***Database:*** A highly structured and organized digital collection of data.

***DBMS:*** Short for Database Management System, a software that is used to store and manage data through the use of databases.

***DDL:*** Short for Data Definition Language, a computer language used to create and modify a database system

***DML:*** Short for Data Manipulation Language, a computer language used to add, delete, and manipulate data in a database.

***ENUM:*** A data type that works as a list of constants, often strings of characters.

***ER Diagram:*** Short for Entity-Relationship diagram, which is used to create and model the conceptual design of a database system, highlighting key data entities, their attributes, and relationships with other entities.

***INT:*** Short for integer, a whole number.

***ISBN:*** Short for International Standard Book Number, a unique identifier for virtually all books.

***ISSN:*** Short for International Standard Serial Number, a unique identifier for publication issues.

***Query:*** A request prepared by a user of a database to a database system. The DBMS interprets the query and prepares and returns the results of the query.

***Relational Schema:*** Describes the logical layout of the database, defining the tables of the database, their columns, and the data types of each column.

***SQL:*** Short for Structured Query Language, a computer language that includes DDL and DML elements. It is used for creating, modifying, and using a database.

***varChar:*** A string of text characters of varying length. Also known simply as a string.

# 2. Stakeholders

## 2.1 End Users

Individuals who interact with the database as part of their daily operations or activities include:

***Library Staff*:** Responsible for managing book inventory, member accounts, and transactions.

***Members*:** Use the system to search for books, check availability, reserve, and manage borrowed books.

## 2.2 Administrators

Individuals who oversee database management, security, and maintenance include:

***Database Administrator*:** Maintain database performance, security, and troubleshoot issues.

## 2.3 Other Stakeholders

Additional parties with an interest in the database include:

***Donors*:** Provide funding and support for systems development and improvement.

***Regulatory Personnel:*** Ensure library procedures, including information systems, adhere to any and all relevant regulations.

# 3. Requirements

## 3.1 Functional Requirements

### 3.1.1 User Administration

The database must be able to be accessed and used by a variety of different user types, including library members, library staff, and database administrators. Each user type will be able to perform different actions on the database as further defined in this document. Database administrators should be able to add, monitor, and remove users from the database as required. Database administrators should also be able to grant library staff users the role of database administrator as well.

### 3.1.2 Data Entry

Users should be able to insert and manipulate data in the database in accordance with the privileges associated with their user type. For example, library members should be able to add contact information. Library staff should be able to add new books and media to the database as needed. A database administrator should be able to insert new records into the database as needed.

### 3.1.3 Data Retrieval

The database should allow users of all types to retrieve information from the database in accordance with their user type. Data should be retrieved through queries to the database.

A preliminary list of queries and the user types who can make them is included below. The database administrator should be allowed to make any query.

***List all books by a specific author:*** This can be used by all user types when needed.

***Find books by publication year:*** This is used primarily by members and staff.

***Check membership status:*** This could be used by all user types, but members should only be able to check their own status.

***Fine calculation:*** Individual members should be able to retrieve their current fines, and staff should be able to retrieve fine information for all users.

***Book availability by genre:*** This query can be made by all user types, but primarily by staff and members.

***Frequent borrowers of a specific genre***: This can be made by staff.

***Books due within the next week:*** This can be made by staff.

***Members with overdue books:*** This can be made by staff.

***Average borrowing time in days:*** This can be made by staff

***Most borrowed author in the last month:*** This can be made by staff

***Monthly fees reported by member type:*** This can be made by staff.

***Exceeded borrowing limits:*** This can be made by staff.

***Frequently borrowed items by member type:*** This can be made by staff.

***Users who have never returned an item late:*** This can be made by staff.

***Average loan duration:*** This can be made by staff.

### 3.1.4 Data Updates

The database must also allow for information in a database, such as a book's availability status, to be altered when necessary and through the actions of appropriate parties.

***Library Member:*** A library member should be able to alter their own contact information and name if needed, but not their user ID.

***Library Staff:*** A library staff member should have more freedom to update material in the database, such as the availability status of a book during the checkout process or the account status of a library member. However, they too have restrictions. For instance, they should not be able to alter their user ID or the information of other library staff.

***Database Administrator:*** The database administrator, a trusted role, should have the ability to update any and all data in the database.

### 3.1.5 Data Deletion

The database should also permit the deletion of data and records as needed, but these actions must only be allowed to be committed by certain user types. Each user type's deletion privileges are listed below:

***Library Member:*** A library member should have the ability to delete contact information as long as one method of contact is kept.

***Library Staff:*** A library staff member should be able to delete records of books and media from the database as needed.

***Database Administrator:*** The administrator should ultimately be granted full privileges over the database, including deletions.

### 3.1.6 Report Generation

The database should be able to generate a number of simple reports, typically requested through queries. A number of these reports are provided below.

***Monthly summary report:*** This report will summarize the number of items loaned, fees collected, and popular items.

***Statistics breakdown:*** This report will provide borrowing statistics by membership type and media category.

***Client borrowing report:*** This report lists the borrowing history, outstanding fees, and media reservations of a given member.

***Item availability and history:*** This report lists the availability and date last borrowed for all items in the collection. This report provides emphasis on items that have not been borrowed within the last six months.

***Overdue items report:*** This report lists all overdue items, also reporting the responsible member and current late fee for each item.

***Revenue summary:*** This report lists the total revenue gained through late fees, showing the fees gained from each membership type as well as media type.

The database should also be able to produce a number of more elaborate reports. These reports include the following:

***Collection analysis:*** This report will identify acquisition trends in the library's collection, divided by genre, over a period of five years. It will provide information on the age of the material in the collection, emphasizing outdated material, and the borrowing patterns of material, with the intent to identify under-represented authors and media genres.

***Problem member analysis:*** This report will comprehensively identify all "problem members" in the database. This report will analyze information on active members' borrowing trends and late fee payments, searching for repeated and sustained instances of overdue returns and late-fee repayment tardiness. This report will provide staff with a comprehensive list of all members who may require intervention or even deactivation of their accounts. It will also highlight potential conflicts that may arise from any actions taken against these members. For instance, a problem member who is currently borrowing a book may not be recommended for deactivation until that book has been returned.

### 3.1.7 Notifications

The database system must provide notifications to members to remind them of upcoming loan return deadlines, current overdue items, and new availability of reserved media items.

## 3.2 Non-functional Requirements

### 3.2.1 Performance

Interactions with the database should be perceived as instantaneous, with minimal delay. The results of each query should be displayed fully, without error, and in a timely manner. The database should also use the minimum computational resources required to function.

### 3.2.2 Security

Data integrity is paramount to the utility of the database system. As such, user privileges, as defined in Section 3.1 must be strictly enforced. Unauthorized attempts to retrieve, insert, modify, and delete data must be detected and prevented. To facilitate user separation, library staff and library members should be allowed to access the database through distinct interfaces.

### 3.2.3 Portability

Access to the database should not be restricted to one device or location. Users should be able to interact with the database from multiple devices and locations.

### 3.2.4 Regulatory Compliance

All design, implementation, and use of the database and its information must comply with any and all applicable regulations.

## 3.3 Data Entities

A preliminarily set of data entities to be used in the Conceptual Design are included below:

***User:***

**Attributes:**

userID (primary key, INT)
name (varChar)
phoneNumber (varChar)
emailAddress (varChar)
physicalAddress (varChar)
userType (ENUM)
accountStatus (ENUM)

**Constraints:**

userID must be a unique integer.
email should be in a valid email address format.
phoneNumber should be in a valid phone number format.
physicalAddress should be in a valid mailing address format.
userType is an enum (admin, staff, member).
accountStatus is an enum (active, inactive)

**Notes:**

The User entity specializes to other entities based on the userType attribute. The Administrator and Staff roles do not require additional attributes and, therefore, may not need new entities defined.

***Member***:

**Attributes:**

userID (primary key, INT) - Inherited from User entity

membershipType (composite attribute)

typeName (ENUM)

borrowingLimit (INT)

lateFeeRate (DECIMAL)

**Constraints:**

typeName is an enum (regular, students, or senior citizens).

borrowingLimit must be an integer greater than zero.

lateFeeRate must be a floating point number greater than zero.

**Notes:**

The Member is a specialization of the User entity and inherits all of its attributes.

membershipType will determine borrowing limits.

***Media:***

**Attributes:**

itemID (primary key, INT)

title (varChar)

itemType (ENUM)

publicationYear (INT)

availabilityStatus (ENUM)

specialPremium (DECIMAL)

specialRestriction (ENUM)

**Constraints:**

itemID must be a unique integer greater than or equal to zero.

availabilityStatus is an enum (available, unavailable).

itemType is an enum (book, digital, magazine).

publicationYear must be an integer greater than zero.

specialPremium must be a floating point number greater than or equal to zero.

specialRestriction is an enum (common, rare)

**Notes:**

itemID will be used to differentiate between items as well as copies.

The Media attribute specializes to other entities based on the itemType attribute.

specialPremium can be used to add the late fee rate for rare or special items.

specialRestriction can be used to apply borrowing restrictions to rarer items.

***Book:***

**Attributes:**

itemID (primary key, INT) - Inherited from Media entity

ISBN (INT)
author (varChar)
genre (varChar)

**Constraints:**

itemID must be a unique integer greater than or equal to zero.
ISBN must be an integer greater than zero.

**Notes:**

This is a specialization of the Media entity and inherits all of its attributes.

***DigitalMedia:***

**Attributes:**

itemID (primary key, INT) - Inherited from Media entity
digitalMediaID (INT)
creator (varChar)
genre (varChar)

**Constraints:**

itemID must be a unique integer greater than or equal to zero.
digitalMediaID must be an integer greater than zero.

**Notes:**

This is a specialization of the Media entity and inherits all of its attributes.

***Magazine***:

**Attributes:**

itemID (primary key, INT) - Inherited from Media entity
ISSN (INT)
publicationDate (DATE)

**Constraints:**

itemID must be a unique integer greater than or equal to zero.
ISSN must be an integer greater than or equal to zero.
publicationDate must be in a valid date format.

**Notes:**

The is a specialization of the Media entity and inherits all of its attributes.

***Loan:***

**Attributes:**

loanID (primary key, INT)
memberID (key to Member.userID, INT)
itemID (key to Media.itemID, INT)
checkoutDate (DATE)
dueDate (DATE)

returnDate (DATE)
lateFeeCharge (DECIMAL)

**Constraints:**

loanID must be a unique integer greater than zero.
returnDate must be on or later than the checkoutDate.
All date attributes must be in a valid date format.
lateFeeCharge must be a floating point number greater than or equal to zero.

**Notes:**

overdueFees is calculated based on Member.membershipType.
Many-to-one relationship with Member entity.
Many-to-one relationship with Media entity.

***Reservation***

**Attributes:**

reservationID (primary key, INT)
memberID (key to Member.userID, INT)
itemID (key to Media.itemID, INT)
reservationDate (DATE)
expirationDate (DATE)
status (ENUM)

**Constraints:**

status is an enum (active, inactive).
expirationDate must be on or later than the reservationDate.

**Notes:**

expirationDate should be defined by library policy.
Many-to-one relationship with *Member* entity.
Many-to-many relationship with *Media* entity, since a reservation can be fulfilled using any copy of a media item.

***Fine***

**Attributes:**

fineID (primary key, INT)
memberID (key to Member.userID, INT)
loanID (key to Loan.loanID, INT)
amount (DECIMAL)
issueDate (DATE)
status (ENUM)

**Constraints:**

fineID must be an integer greater than or equal to zero.

amount must be a floating point number greater than or equal to zero.
Many-to-one relationship with *Loan* entity.
Many-to-one relationship with *Member* entity.

***Payment:***

**Attributes:**

paymentID (primary key, INT)
memberID (key to Member.userID, INT)
fineID (key to Fine.fineID, INT)
paymentDate (DATE)
amountPaid (DECIMAL)

**Constraints:**

paymentID must be an integer greater than or equal to zero.
amountPaid must be a floating point number greater than or equal to zero.
paymentDate must be in a valid date format.
Many-to-one relationship with *Member* entity.
Many-to-one relationship with *Fine* entity.

# 4. Hardware and Software Requirements

## 4.1 Hardware Requirements

The use of this database system requires two hardware components. For accessibility among several users, the database must be stored on a server or a series of servers. This server or series of servers must have the storage and computational resources required to store a small database system as well as facilitate several connections and data requests simultaneously. Secondly, individual users will require an internet-enabled device to access the database system through this server or servers. Any modern laptop or desktop computer should have the necessary capabilities required to connect to and request from the database system.

## 4.2 Software Requirements

To create and manage the database, a DBMS will be needed. The DBMS used must support the use of a DDL to implement the logical relational design of the database. It must also support the use of a DML to facilitate data entry, update, retrieval, and deletion. In particular, the DBMS must support SQL, which can be used for all of these purposes. The DBMS should also implement concurrency control and enforce user privileges.

# 5. Appendix

## 5.1 Library Policies

Borrowing limits based on the membershipType attribute of the Member entity are not typically defined by the database designers but by preexisting library policies. For context, the relevant policies are listed below:

***Regular Members:*** Regular members can borrow up to five items at any given time and have a set late fee rate of $4/day.

***Senior Members:*** Senior members can borrow up to seven items at any given time and have a set late fee rate of $2/day.

***Student Members:*** Student members can borrow up to ten items at any given time and have a set late fee rate of $1/day.

Additionally, according to library policy, reservations on a media item expire three days after the item has been returned to the library. Rarer items will have customized borrowing restrictions and late fee premiums that will reflect their rarity.