# Library Database Management System
## Conceptual Model

**Version 1.0**

# Table of Contents

# Entity-Relationship Modeling

## 1. Introduction

### 1.1 Project Overview

The Library Database Management System project will deliver an efficient, user-friendly, and secure database system to support small library operations. The DBMS will enable seamless management of loanable items (books, digital media, magazines), enforce borrowing rules based on diverse membership categories, and generate insightful reports. By integrating modern database design principles, the system will streamline item tracking, membership management, and financial oversight.

### 1.2 Scope

The Library Database Management System (LMS) project will design and implement a relational database for a small library to manage loanable items, memberships, borrowing rules, and generate reports. It will model entities like books, digital media, magazines, and clients, and enforce constraints like borrowing limits and fees based on membership type. The system will include features for managing loans, returns, and client accounts, with user interfaces for both staff and clients. Advanced queries will generate financial and activity reports, while concurrency and transaction management will ensure seamless multi-user operations. The LMS will be developed through domain modeling, database design, and implementation phases, ensuring functionality and data integrity.

### 1.3 Glossary

LMS - Library Management System
DBMS - Database Management System

## 2. Identify Entities

Entities:

- Item
    - Book
    - Magazine
    - Digital Media Item
    - DVD
    - Music
- User
    - Member
    - Staff
- Transaction
    - Purchase
    - Borrow

## 3. Define Attributes

Entity: Item

- Item ID - Char(10)
- Price - Int()
- Year - Numeric(4, 0)
- Availability Status - Enum('Available', 'Checked Out', 'Reserved')
- Item Type - Enum('Book', 'Digital Media', 'Magazine', ‘DVD’, ‘Music’)

Entity: Book

- Item ID - Char(10)
- ISBN - Varchar(13)
- Title - Varchar(50)
- Author - Varchar(50)
- Publisher - Varchar(20)
- Subject - Varchar(20)
- Genre - Varchar(30)

Entity: Magazine

- Item ID - Char(10)
- ISSN - Varchar(13)
- Title - Varchar(50)
- Publisher - Varchar(20)
- Edition - Int()
- Publish date - Char(10)

Entity: Digital Media

- Item ID - Char(10)
- DOI - Int()
- Media Type - Varchar(15)
- Release Date - Char(10)
- Title - Char(20)
- Creator - Char(20)

Entity: DVD
- Item ID - Char(10)
- Name - Varchar(100)
- Director - Varchar(20)
- Duration - Time()

Entity: Music
- Item ID - Char(10)
- Title - Char(20)
- Artist - Varchar(50)
- Album - Varchar(50)
- Format - Enum(‘CD’, ‘Vinyl’, ‘Digital’)

Entity: User
- User ID - Char(10)
- Card Number - Int()
- Name - Varchar(20)
- Address - Varchar(30)
- Email - Varchar (20)
- Phone Number - Char(12)

Entity: Member
- User ID - Char(10)
- Membership Type - Varchar(15)
- Borrowing Limit: Int()
- Membership Fee: Decimal(7, 2)

Entity: Staff
- User ID - Char(10)
- Position - Varchar(50)
- Salary - Decimal(10,2)

Entity: Transaction
- Transaction ID: Varchar(20)
- User ID: Varchar(20)
- Item ID: Varchar(20)
- Transaction Date: Date()
- Status: Enum('Completed', 'Pending', 'Canceled')

Entity: Purchase
- Transaction ID - Int()
- Purchase Amount - Decimal(10, 2)
- Payment Method - Enum('Credit Card', 'Debit Card', 'Cash', 'Other')

Entity: Borrow
- Transaction ID - Int()
- Borrow Date - Date()
- Due Date - Date()
- Return Date - Date()
- Late Fee - Decimal(5, 2)

## 4. Establish Relationships

Relationships:
- Member "Makes" a Transaction
    - That member can be part of 0 to many transactions
    - That transaction is executed for 1 member
- Staff "Manages" a Transaction
    - That staff can be apart of 0 to many transactions
    - That transaction can me managed by 1 or more staff
- Purchase "Involves" Item
    - That purchase may involve 1 to many items
    - That item may be involved in 0 or 1 purchase
- Borrow "Involves" Item
    - That borrow may involve 1 to many items
    - That item may be involved in 0 to many borrows

Specialization:
- The User Entity has subgroupings Member and Staff
- The Transaction Entity has subgroupings Purchase and Borrow
- The Item Entity has subgroupings Music, DVD, Book, Magazine, Digital Media

Constraints:
- Borrowing Limits: Each member has a borrowing limit defined by their membership type.
- Availability Status: An item can only be borrowed if it is Available.
- Late Fees: Late fees are calculated based on the due date and return date in the Borrow relationship.
- Unique User IDs: Each user (whether member or staff) must have a unique ID.
- Purchase: Items can only be purchased once, after which they may be marked as unavailable for future purchases.

## 5. Entity-Relationship Diagram