Library DB Conceptual Modeling

Team SQLibrary

March 16, 2025

1 Introduction

1.1 Project Overview

This database project is focused on designing an SQL-based Library Management System (LMS) to efficiently manage books, magazines, borrowers, transactions, and overdue monitoring. By replacing traditional paper-based methods, the LMS will enhance organization, improve data accuracy, and streamline daily library operations. It will feature a structured, user-friendly interface that simplifies record-keeping, enforces borrowing policies, and provides automated tracking of overdue materials. Additionally, the system will generate comprehensive reports to assist librarians in decision-making and overall management.

1.2 Scope

The Library Management System (LMS) will oversee various types of materials, such as physical and digital books, as well as magazines. It will facilitate user management, enforce borrowing policies, and track overdue items. The system will regulate borrowing limits according to membership categories, enable reservations, and produce reports. However, it will not incorporate physical checkout mechanisms like barcode scanners or RFID tags, nor will it integrate with external digital content providers.

1.3 Glossary

Below is a list of key terms and acronyms used in this project:

- **Library Management System (LMS)**: An architecture created to manage library assets, member enrollments, and transaction operations.
- **Overdue Monitoring**: A device that tracks and records overdue books and calculates corresponding fees.
- **Membership Type**: A classification of customers that identifies the borrower ability and charge terms for these customers (for example, normal, student, old person (senior)).

- **Reservation**: A feature allowing clients to place holds on borrowed items
- **Notifications**: Automated alerts for due dates, overdue items, and reserved book availability
- **Item**: A generalized category for the objects the library contains, categorized by the upper-case "I" to represent the entity that DM, Magazine, and Book inherit from.
- **DM**: Shorthand for Digital Media.
- **ISBN**: International Standard Book Number is a unique identifier assigned to books and other published materials.

2 Identify ER Modeling Components

2.1 Identify Entities

List all the major entities that will be part of your database.

- **Book**
- **Digital Media**
- **Magazine**
- **Member**
- **Membership Type**
- **Loan**
- **Fine**
- **Reservation**
- **Librarian**
- **Item**

2.2 Define Attributes

Attributes and key type/data type specification for each entity. The format is Entity (Attribute (data_type)). The format for Key and Foreign attributes is **(key, data_type)**.

2.3 Define Attributes

Attributes and key type/data type specification for each entity. The format is Entity (Attribute (data_type)). The format for Key and Foreign attributes is **(key, data_type)**.

Entity	Attributes
Book	ISBN (**PK, INT**), Title (STR), Author (STR), Genre (STR), PubYear (INT), Availability (BOOL)
Digital Media	MediaID (**PK, INT**), Title (STR), Creator (STR), MediaType (STR), Availability (BOOL)
Magazine	IssueID (**PK, INT**), Title (STR), IssueNumber (INT), PublicationDate (INT), Status (BOOL)
Member	MemberID (**PK, INT**), Name (STR), Contact (STR), TypeID (**FK, CHAR**), AccountStatus (STR)
Membership Type	TypeID (**PK, CHAR**), TypeName (STR), BorrowLimit (INT), LateFeeRate (INT)
Loan	LoanID (**PK, INT**), MemberID (**FK, INT**), ItemID (**FK, INT**), ItemType (ITEM), LoanDate (STR), DueDate (STR), ReturnDate (STR), FineID (**FK, INT**)
Fine	FineID (**PK, INT**), LoanID (**FK, INT**), Amount (INT), PaymentStatus (BOOL)
Reservation	ReservationID (**PK, INT**), MemberID (**FK, INT**), ItemID (**FK, INT**), ItemType (ITEM), RequestDate (STR), Status (BOOL)
Librarian	LibrarianID (**PK, INT**), Name (STR), Contact (STR), Role (STR)
Item	ItemID (**PK, INT**), Title (STR), Availability (BOOL)

Table 1: Entity Attributes and Data Types

2.4 Define Relationships

Determine how the entities are related to each other, including multiplicity (one-to-one, one-to-many, many-to-many) and constraints. The relationship format is "**relationship** Entity" or vice versa.

- **Book**
  - **Relationships: is_an** Item
  - **Multiplicity:** One-to-One (one item can be one book)
  - **Constraints:** PRIMARY KEY (ISBN), unique ISBN, availability could be an ENUM
- **Digital Media**
  - **Relationships: is_an** Item

  - **Multiplicity:** One-to-One (one item can be one piece of DM)
  - **Constraints:** PRIMARY KEY (MediaID), unique MediaID, availability could be an ENUM

- **Magazine**
  - **Relationships: is_an** Item
  - **Multiplicity:** One-to-One (one item can be one magazine)
  - **Constraints:** PRIMARY KEY (IssueID), unique IssueID, Status could be an ENUM

- **Member**
  - **Relationships: is_given** Loan, **has_type** Member Type, makes Reservation with **made_by**
  - **Multiplicity:** One-to-Many (one member can have many loans), One-to-One (one member can have one type), One-to-One (one member makes the reservation)
  - **Constraints:** PRIMARY KEY (MemberID), references MembershipType.TypeID

- **Membership Type**
  - **Relationships:** Gives type to Member with **has_type**
  - **Multiplicity:** One-to-One (one member has one type)
  - **Constraints:** PRIMARY KEY (TypeID), BorrowLimit could be TINYINT

- **Loan**
  - **Relationships: processed_by** Librarian, **incurs** Fine, **includes** Item, and gives loan to Member with **is_given**
  - **Multiplicity:** Many-to-One (many loans are processed by one Librarian), One-to-One (one loan can incur one associated Fine), One-to-Many (one Loan includes many Items), Many-to-One (many Loans can be given to one Member)
  - **Constraints:** PRIMARY KEY (LoanID), references MemberID & FineID

- **Fine**
  - **Relationships:** Received from a Loan with **incurs**
  - **Multiplicity:** One-to-One (one loan incurs one fine at a time)
  - **Constraints:** PRIMARY KEY (FineID), references LoanID, PaymentStatus could be an ENUM

- **Reservation**
  - **Relationships: made_by** Member, **reserves** Item
  - **Multiplicity:** One-to-One (one Member has one Reservation at a time), One to Many (one Reservation reserves many Items)
  - **Constraints:** PRIMARY KEY (ReservationID), references MemberID
- **Librarian**
  - **Relationships:** Handles Loans with **processed_by**
  - **Multiplicity:** One-to-Many (one Librarian processes many Loans)
  - **Constraints:** PRIMARY KEY (LibrarianID), Role could be ENUM('Admin', 'Staff')
- **Item**
  - **Relationships:** Is obtained by a Loan with **includes**, or by Reservation with **reserves**
  - **Multiplicity:** Many-to-One (many Items are in one Loan), Many-to-One (many Items are in one Reservation)
  - **Constraints:** PRIMARY KEY (Item ID), unique ITEM ID

2.5 ER Diagram

Below is the ER diagram for the library database:

Our ER Diagram will go here when finished.