Library Management System - Logical Relational Model

Introduction

Project Overview

The Library Management System (LMS) is designed to manage a small library's diverse collection of loanable items—including books, digital media, and magazines. The system supports key functions such as tracking loans and returns, managing multiple membership types (each with distinct borrowing limits and fee structures), processing reservations, and generating detailed reports to assist with decision-making. The system is intended for both library staff and patrons, aiming to streamline daily operations and improve the overall user experience.

Scope

This project covers the complete data modeling for the library system. It includes:

- **Data Entities**: Items (Books, Digital Media, Magazines), Clients, Membership Types, Loan Transactions, and Reservations.
- **Relationships**: How clients interact with items (borrow, return, reserve) and how membership types govern borrowing privileges.
- **Constraints**: Business rules such as borrowing limits, late fee calculations, and item availability status.
- **Feedback**: Additionally, the scope includes data entry, updates, deletions, and report generation functions.
- **Excluded**: Integration with external systems and advanced analytics is excluded in this phase.

Glossary

- **ERD**: Entity-Relationship Diagram, a visual representation of entities and their relationships.
- **Entity**: A real-world object or concept (e.g., Book, Client).
- **Attribute**: A property or characteristic of an entity (e.g., ISBN, Title).
- **PK**: Primary Key, a unique identifier for an entity.
- **FK**: Foreign Key, an attribute that links one entity to another.
- **Cardinality**: The numerical relationships between entities (e.g., one-to-many, many-to-many).

- **Relation**: A table in a relational database.
- **Tuple**: A row in a relation.
- **Domain**: The set of allowable values for an attribute.
- **Functional Dependency (FD)**: A constraint between two sets of attributes in a relation.
- **Normalization**: The process of organizing data to reduce redundancy and improve data integrity.
- **BCNF**: Boyce-Codd Normal Form, a normal form used in database normalization.
- **3NF**: Third Normal Form, a normal form used in database normalization.

Relational Schema Mapping

Identify Relations

Based on the ER model, we identify the following relations:

MEMBERSHIP: Represents different membership types with associated fee structures and borrowing limits.

CLIENT: Represents library patrons with their personal information and membership details.

ITEM: Represents any item in the library's collection (book, magazine, digital media) with shared attributes.

BOOK: Stores book-specific attributes (one-to-one with `ITEM` when `ItemType='Book'`).

DIGITAL_MEDIA: Stores digital media–specific attributes (one-to-one with `ITEM` when `ItemType='DigitalMedia'`).

MAGAZINE: Stores magazine-specific attributes (one-to-one with `ITEM` when `ItemType='Magazine'`).

LOAN_TRANSACTION: Records borrowing events, including dates, due dates, and fines.

RESERVATION: Tracks client reservations for items.

Define Attributes and Domains

For each relation, we define the following attributes and their domains:

MEMBERSHIP

MembershipType: VARCHAR(50), the type of membership

Description: VARCHAR(255), description of the membership

BorrowingLimit: INT, maximum number of items that can be borrowed

FeeStructure: DECIMAL(5,2), fee structure for the membership

CLIENT

ClientID: INT, unique identifier for a client

Name: VARCHAR(100), client's name

PhoneNumber: VARCHAR(20), client's phone number

Email: VARCHAR(100), client's email address

MembershipType: VARCHAR(50), client's membership type

AccountStatus: ENUM('active', 'suspended', 'closed'), current status of the client's account

ITEM

ItemID: INT, unique identifier for an item

ItemType: ENUM('Book', 'DigitalMedia', 'Magazine'), type of the item

Title: VARCHAR(255), title of the media

PublicationDate: DATE, publication date of the item

AvailabilityStatus: ENUM('available', 'borrowed', 'reserved'), current status of the item

BOOK

ItemID: INT (matches `ITEM.ItemID`)

ISBN: VARCHAR(13), International Standard Book Number

Author: VARCHAR(255), author(s) of the book

Genre: VARCHAR(50), genre of the book

DIGITAL_MEDIA

ItemID: INT (matches `ITEM.ItemID`)

Creator: VARCHAR(255), creator(s) of the digital media

Format: ENUM('DVD', 'Blu-ray', 'Digital'), format of the digital media

MAGAZINE

ItemID: INT (matches `ITEM.ItemID`)

VolumeNumber: INT, volume number of the magazine

IssueNumber: INT, issue number of the magazine

LOAN_TRANSACTION

TransactionID: INT, unique identifier for a loan transaction

ClientID: INT, identifier of the client who borrowed the item

ItemID: INT, identifier of the borrowed item

BorrowDate: DATE, date the item was borrowed

DueDate: DATE, date the item is due to be returned

ReturnDate: DATE, date the item was returned (NULL if not yet returned)

CalculatedFine: DECIMAL(5,2), fine calculated for late return (0 if returned on time)

RESERVATION

ReservationID: INT, unique identifier for a reservation

ClientID: INT, identifier of the client who made the reservation

ItemID: INT, identifier of the reserved item

ReservationDate: DATE, date the reservation was made

Status: ENUM('active', 'cancelled', 'fulfilled'), current status of the reservation

Determine Primary Keys

MEMBERSHIP: `MembershipType` (PK)

CLIENT: `ClientID` (PK)

ITEM: `ItemID` (PK)

BOOK: `ItemID` (PK)

DIGITAL_MEDIA: `ItemID` (PK)

MAGAZINE: `ItemID` (PK)

LOAN_TRANSACTION: `TransactionID` (PK)

RESERVATION: `ReservationID` (PK)

Establish Foreign Keys

CLIENT:

- `MembershipType` → **MEMBERSHIP**.`MembershipType`

LOAN_TRANSACTION:

- `ClientID` → **CLIENT**.`ClientID`
- `ItemID` → **ITEM**.`ItemID`

RESERVATION:

- `ClientID` → **CLIENT**.`ClientID`
- `ItemID` → **ITEM**.`ItemID`

BOOK:

- `ItemID` → **ITEM**.`ItemID`

DIGITAL_MEDIA:

`ItemID` → **ITEM**.`ItemID`

MAGAZINE:

`ItemID` → **ITEM**.`ItemID`

Establish Functional Dependencies (FDs)

MEMBERSHIP

`MembershipType → Description, BorrowingLimit, FeeStructure`

CLIENT

`ClientID → Name, PhoneNumber, Email, MembershipType, AccountStatus`

(`MembershipType` is an FK to `MEMBERSHIP.MembershipType`)

ITEM

`ItemID → ItemType, Title, PublicationDate, AvailabilityStatus`

BOOK

`ItemID → ISBN, Author, Genre`

(1:1 FK to `ITEM(ItemID)`; if `ItemType='Book'`, exactly one `BOOK` row matches)

DIGITAL_MEDIA

`ItemID → Creator, Format`

(1:1 FK to `ITEM(ItemID)`; if `ItemType='DigitalMedia'`, exactly one row matches)

MAGAZINE

`ItemID → VolumeNumber, IssueNumber`

(1:1 FK to `ITEM(ItemID)`; if `ItemType='Magazine'`, exactly one row matches)

LOAN_TRANSACTION

`TransactionID → ClientID, ItemID, BorrowDate, DueDate, ReturnDate, CalculatedFine`

(`ClientID` is an FK to `CLIENT.ClientID`)

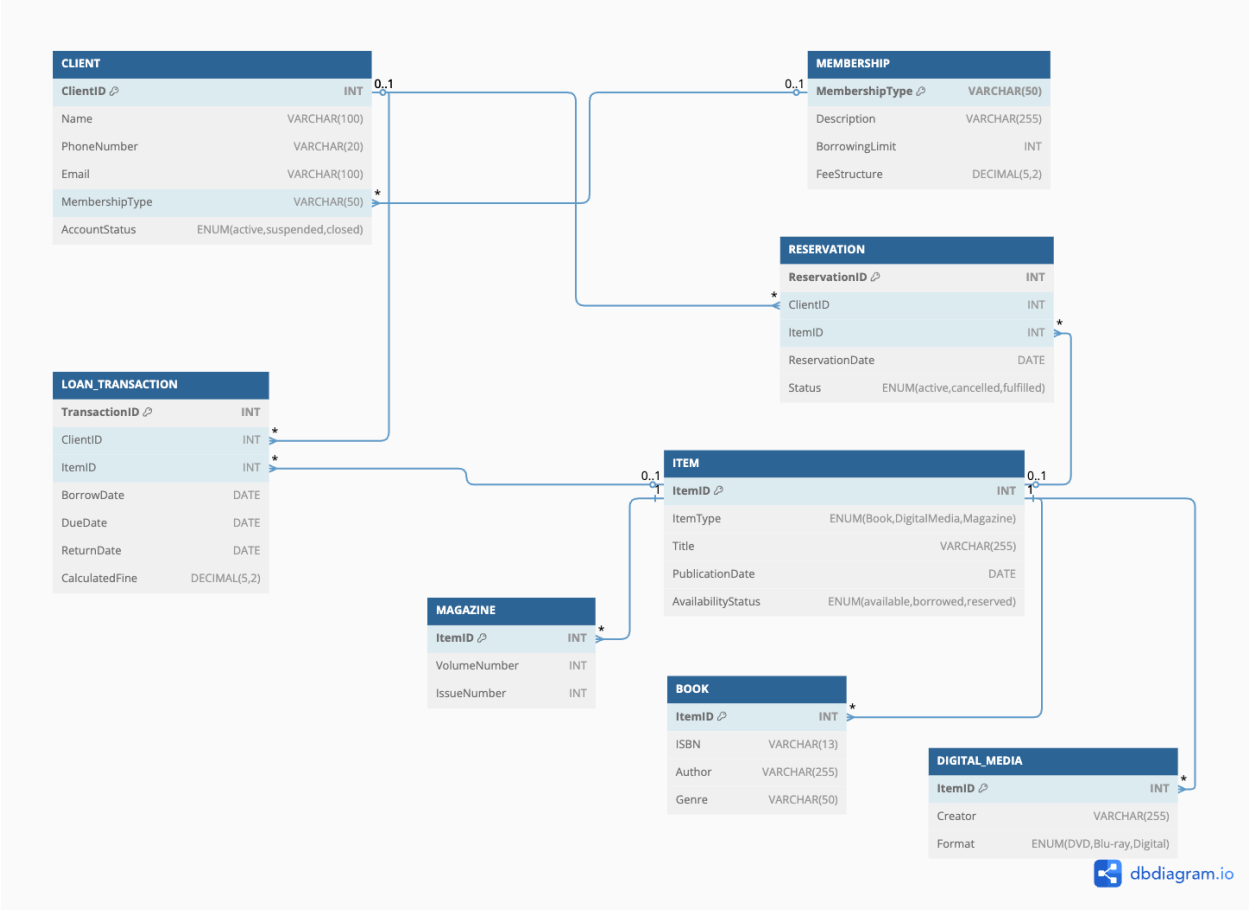
(`ItemID` is an FK to `ITEM.ItemID`)

RESERVATION

`ReservationID → ClientID, ItemID, ReservationDate, Status`

(`ClientID` is an FK to `CLIENT.ClientID`)

(`ItemID` is an FK to `ITEM.ItemID`)



Schema Documentation with a Data Dictionary

Relation	Attribute	Data Type	Domain	Constraint	Description
MEMBERSHIP	MembershipType	VARCHAR(50)	Any valid string	PK	Type of membership (e.g., regular, student, senior)
	Description	VARCHAR(255)	Any valid string		Description of the membership
	BorrowingLimit	INT	Positive integer	NOT NULL	Maximum number of items that can be borrowed
	FeeStructure	DECIMAL(5,2)	Positive decimal	NOT NULL	Fee structure for the membership
CLIENT	ClientID	INT	Positive integer	PK, AUTO_INCREMENT	Unique identifier for a client
	Name	VARCHAR(100)	Any valid string	NOT NULL	Client's name
	PhoneNumber	VARCHAR(20)	Valid phone number format		Client's phone number

	Email	VARCHAR (100)	Valid email format		Client's email address
	MembershipType	VARCHAR (50)	Any valid string	FK → MEMBERSHIP.MembershipType	Client's membership type
	AccountStatus	ENUM	'active', 'suspended', 'closed'		Current status of the client's account
ITEM	ItemID	INT	Positive Integer	PK, AUTO_INCREMENT	Unique identifier for an item
	ItemType	ENUM	'Book', 'DigitalMedia', 'Magazine'	NOT NULL	Type of the item
	Title	VARCHAR (255)	Valid string	NOT NULL	Title of the item
	Publication Date	DATE	Valid Date	NOT NULL	Publication date of the item
	Availability Status	ENUM	'available', 'borrowed', 'reserved	NOT NULL	Current status of the digital media
BOOK	ISBN	VARCHAR (13)	Valid ISBN format		International Standar

					d Book Number
	Author	VARCHAR (255)	Any valid string	NOT NULL	Author(s) of the book
	Genre	VARCHAR (50)	Any valid string		Genre of the book
DIGITAL_MEDIA	Creator	VARCHAR (255)	Any valid string	NOT NULL	Creator(s) of the digital media
	Format	ENUM	'DVD', 'Blu-ray', 'Digital'		Format of the digital media
MAGAZINE	VolumeNumber	INT	Positive integer	NOT NULL	Volume number of the magazine
	IssueNumber	INT	Positive integer	NOT NULL	Issue number of the magazine
LOAN_TRANSACTION	TransactionID	INT	Positive integer	PK, AUTO_INCREMENT	Unique identifier for a loan transaction
	ClientID	INT	Positive integer	FK → CLIENT.ClientID	Identifier of the client who borrowe

					d the item
	ItemID	INT	Positive integer	FK → ITEM.ItemID	Identifier of the borrowed item
	BorrowDate	DATE	Valid date	NOT NULL	Date the item was borrowed
	DueDate	DATE	Valid date	NOT NULL	Date the item is due to be returned
	ReturnDate	DATE	Valid date or NULL		Date the item was returned (NULL if not yet returned)
	CalculatedFine	DECIMAL(5,2)	Non-negative decimal	DEFAULT 0, CHECK (CalculatedFine >= 0)	Fine calculated for late return
RESERVATION	ReservationID	INT	Positive integer	PK, AUTO_INCREMENT	Unique identifier for a reservation

	ClientID	INT	Positive integer	FK → CLIENT.ClientID	Identifier of the client who made the reservation
	ItemID	INT	Positive integer	FK → ITEM.ItemID	Identifier of the reserved item
	ReservationDate	DATE	Valid date	NOT NULL	Date the reservation was made
	Status	ENUM	'active', 'cancelled', 'fulfilled'		Current status of the reservation

Normalization Considerations

The relational schema design already conforms to the Third Normal Form (3NF) and Boyce-Codd Normal Form (BCNF) because:

1. **First Normal Form (1NF)**: All attributes contain atomic values, and there are no repeating groups.
2. **Second Normal Form (2NF)**: All non-key attributes are fully functionally dependent on the primary key.
3. **Third Normal Form (3NF)**: There are no transitive dependencies of non-key attributes on the primary key.
4. **Boyce-Codd Normal Form (BCNF)**: For every non-trivial functional dependency X → Y, X is a superkey.

The design decisions that contribute to this normalization include:

- Separating different types of items (BOOK, DIGITAL_MEDIA, MAGAZINE) into different relations to handle their unique attributes.

- Creating separate relations for MEMBERSHIP, CLIENT, LOAN_TRANSACTION, and RESERVATION to avoid redundancy.
- Using appropriate foreign keys to establish relationships between relations.

Appendices

Appendix A: Design Choices and Assumptions

1. **Item Modeling**:
    - Books, DigitalMedia, and Magazines are now modeled as specializations of an Item generalization to unify the common attributes. Each specialization captures unique attributes of the ItemType.
    - This current approach provides more unity between the item types at the cost of introducing slightly more complexity with inheritance.
2. **LoanTransaction and Reservation Linking**:
    - LoanTransaction and Reservation now only use ItemID to reference an Item.
3. **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.
    - Only items with an AvailabilityStatus of 'available' may be loaned or reserved.
    - These rules should be enforced through application logic and/or database triggers.
4. **Data Types and Constraints**:
    - ISBN is stored as VARCHAR(13) to accommodate both 10-digit and 13-digit ISBN formats.
    - Enums are used for fields with a fixed set of possible values to ensure data integrity.
    - Default values and check constraints are specified where appropriate to enforce business rules.