The Logical Relational Model

Version 1.4

EECS 447 Project

The Logical Relational Model

EECS 447

Version 1.4

Revision History

Date	Version	Description	Authors
02/07/25	1.0	Document creation, role assignment, and initial division of document sections.	Fatima Avila, Siddh Bharucha, Bhavik Goplani, Vy Luu, Suhaan Syed, Alexis Vielma
02/07/25	1.2	Brainstorming, ER Diagram creation	Fatima Avila, Siddh Bharucha, Bhavik Goplani, Vy Luu, Suhaan Syed, Alexis Vielma
03/16/25	1.3	Document finalization and revision of modeling and written sections.	Fatima Avila, Siddh Bharucha, Bhavik Goplani, Vy Luu, Suhaan Syed, Alexis Vielma
4/4/2025	1.4	Initial draft and schema mapping completed	Fatima Avila, Siddh Bharucha, Bhavik Goplani, Vy Luu, Suhaan Syed, Alexis Vielma

Database Requirements Specifications

1. Introduction

Project Overview

Tech Titans is building a robust **Library Management System (LMS)** that efficiently organizes and manages physical and digital library content. The system simplifies core library operations such as borrowing, returning, fines processing, and reservation notifications. By leveraging a relational database, LMS ensures reliable data management for members and staff alike.

Scope

The system includes functionalities for managing members, items (books, magazines, and digital media), borrowing transactions, reservations, notifications, staff roles, and fine payments. It excludes room booking, event management, and self-service account edits for members.

Glossary:

- LMS (Library Management System): A software system designed to manage library resources, including books, digital media, memberships, and borrowing transactions.
- ISBN (International Standard Book Number): A unique numeric identifier assigned to books for classification and tracking.
- ISSN (International Standard Serial Number): A unique identifier for periodicals, such as magazines and journals.
- MariaDB: The specific DBMS selected for this project, compatible with MySQL, used to store and manage library records.
- Primary Key (PK): A unique identifier assigned to each record in a database table to ensure data integrity.
- Foreign Key (FK): A database field that links one table to another, maintaining relationships between entities.
- Authentication & Authorization: Security measures that control system access, ensuring that only authorized users can modify records or access sensitive data.
- Backup & Recovery: A strategy for storing copies of data to protect against data loss due to system failures or security breaches.
- Relational Schema: A logical structure that defines tables, attributes, keys, and constraints in a relational database.

- Functional Dependency (FD): A relationship where one attribute uniquely determines another.
- Normalization: The process of organizing data to reduce redundancy

2. Relational Schema Mapping

2.1 Identify Relations

1. Member
2. Membership_Type
3. Library_Item
4. Book (inherits from Library_Item)
5. Digital_Media (inherits from Library_Item)
6. Magazine (inherits from Library_Item)
7. Borrowing_Transaction
8. Reservation
9. Payment
10. Staff
11. Notification

2.2 Define Attributes and Domains

Relation	Attribute	Domain	Notes
Member	member_id	INT	PK
	name	VARCHAR(100)	
	contact_info	VARCHAR(255)	Email/phone
	membership_type_id	INT	FK → Membership_Type

	account_status	ENUM('Active', 'Suspended', 'Overdue')	
Membership_Type	membership_type_id	INT	PK
	type_name	ENUM('Regular', 'Student', 'Senior Citizen')	
	max_borrow_limit	INT	
	fine_rate	DECIMAL(5,2)	Dollars/day
Library_Item	item_id	INT	PK
	title	VARCHAR(255)	
	item_type	ENUM('Book', 'Digital Media', 'Magazine')	
	availability_status	ENUM('Available', 'On Loan', 'Reserved')	
Book	book_id	INT	PK & FK → Library_Item
	isbn	VARCHAR(20)	Unique
	author	VARCHAR(100)	
	genre	VARCHAR(50)	
	publication_year	INT	4-digit year
Digital_Media	media_id	INT	PK & FK → Library_Item
	creator	VARCHAR(100)	

	format	ENUM('eBook', 'Audiobook', 'Video', 'Other')	
Magazine	magazine_id	INT	PK & FK → Library_Item
	issue_number	INT	
	publication_date	DATE	
Borrowing_Transaction	borrow_id	INT	PK
	member_id	INT	FK → Member
	item_id	INT	FK → Library_Item
	staff_id	INT	FK → Staff
	borrow_date	DATE	
	due_date	DATE	
	return_date	DATE	Nullable
	fine_incurred	DECIMAL(5,2)	
Reservation	reservation_id	INT	PK
	member_id	INT	FK → Member
	item_id	INT	FK → Library_Item
	reservation_date	DATE	
	expiry_date	DATE	
Payment	payment_id	INT	PK
	member_id	INT	FK → Member
	amount_paid	DECIMAL(7,2)	
	payment_date	DATE	
Staff	staff_id	INT	PK

	name	VARCHAR(100)	
	role	ENUM('Librarian', 'Administrator')	
	contact_info	VARCHAR(255)	
Notification	notification_id	INT	PK
	member_id	INT	FK → Member
	notification_date	DATETIME	
	notification_type	ENUM('Reservation', 'Due Date Alert', 'Overdue Alert')	

2.3 Determine Primary Keys

RELATION	PRIMARY KEY
Member	member_id
Membership_Type	membership_type_id
Library_Item	item_id
Book	book_id (same as item_id)
Digital_Media	media_id (same as item_id)
Magazine	magazine_id (same as item_id)
Borrowing_Transaction	borrow_id
Reservation	reservation_id
Payment	payment_id
Staff	staff_id
Notification	notification_id

2.4 Establish Foreign Keys

Relation	Foreign Key	References
Member	membership_type_id	Membership_Type(membership_type_id)
Book	book_id	Library_Item(item_id)
Digital_Media	media_id	Library_Item(item_id)
Magazine	magazine_id	Library_Item(item_id)
Borrowing_Transaction	member_id	Member(member_id)
Borrowing_Transaction	item_id	Library_Item(item_id)
Borrowing_Transaction	staff_id	Staff(staff_id)
Reservation	member_id	Member(member_id)
Reservation	item_id	Library_Item(item_id)
Payment	member_id	Member(member_id)
Notification	member_id	Member(member_id)

These FKs support multiplicity as described in the ER modeling:

- A member can have many borrowings/reservations/payments.
- Each reservation is for exactly one item.
- Every borrow transaction is tied to one staff.

2.5 Establish Functional Dependencies (FDs)

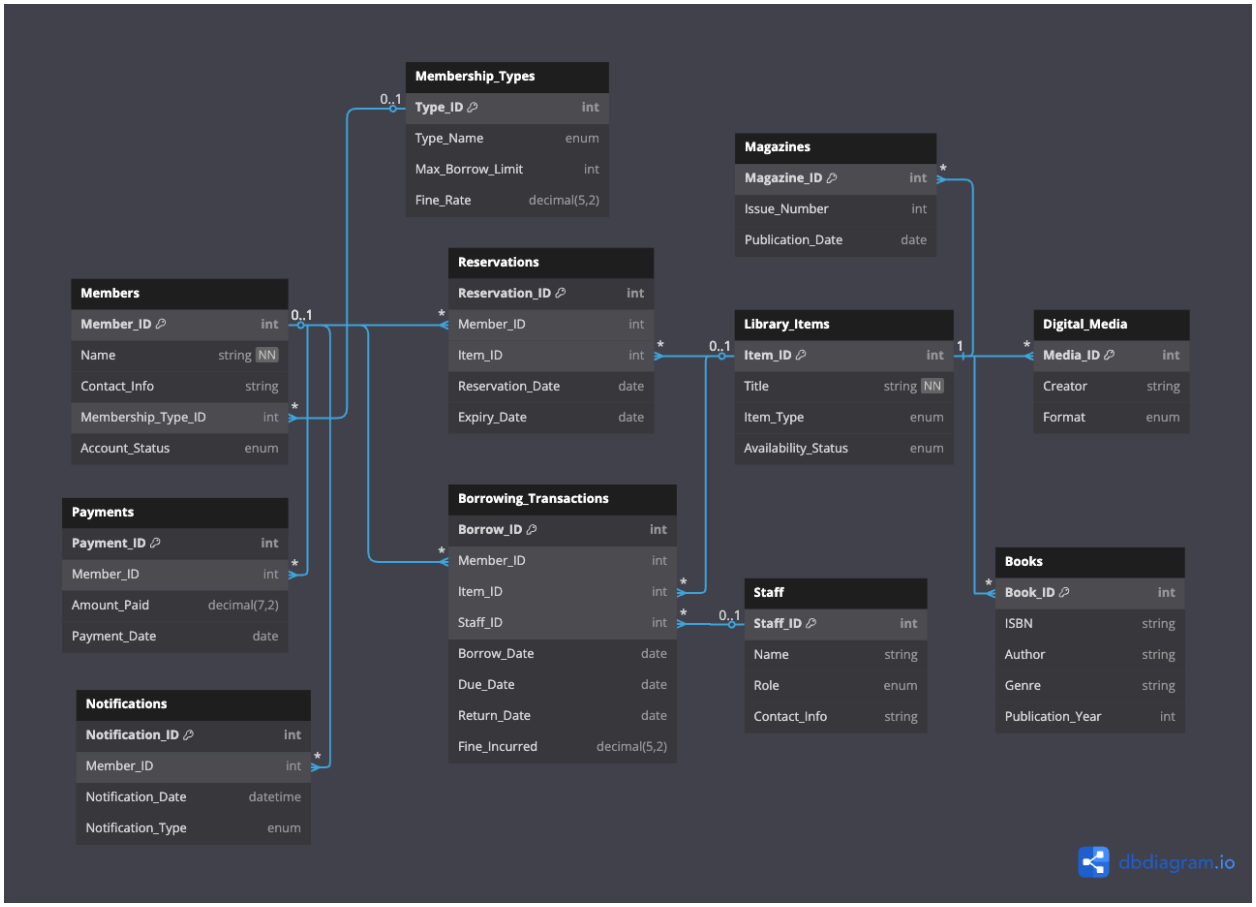
Entity-Level FDs

- member_id → name, contact_info, membership_type_id, account_status
- membership_type_id → type_name, max_borrow_limit, fine_rate
- item_id → title, item_type, availability_status
- book_id → isbn, author, genre, publication_year
- media_id → creator, format
- magazine_id → issue_number, publication_date
- borrow_id → member_id, item_id, staff_id, borrow_date, due_date, return_date, fine_incurred
- reservation_id → member_id, item_id, reservation_date, expiry_date
- payment_id → member_id, amount_paid, payment_date
- staff_id → name, role, contact_info
- notification_id → member_id, notification_date, notification_type

Business Rule Constraints

- One member_id is linked to one membership_type_id
- A member cannot exceed max_borrow_limit set by their membership type
- Fine is computed as (return_date - due_date) * fine_rate
- A reservation is only allowed if the item's availability_status ≠ 'Available'
- availability_status is updated automatically upon borrowing/return
- notification records are system-triggered based on overdue/reservation events

3. Relational schema diagram



4. Schema Documentation with Data Dictionary

Attribute Name	Data Type	Description
member_id	INT	Unique identifier for each member (Primary Key)
membership_type_id	INT	Foreign key referencing Membership_Type
name	VARCHAR(100)	Full name of the member
contact_info	VARCHAR(255)	Email or phone number
account_status	VARCHAR	Status of the member account ('Active', 'Suspended', 'Overdue')
membership_type_id	INT	Unique identifier for each membership type (Primary Key)
type_name	VARCHAR	Membership category ('Regular', 'Student', 'Senior Citizen')
max_borrow_limit	INT	Maximum number of items a member can borrow
fine_rate	DECIMAL(5,2)	Daily fine rate for overdue items
item_id	INT	Unique identifier for each item (Primary Key)
title	VARCHAR(255)	Title of the item
item_type	VARCHAR	Type of item ('Book', 'Magazine', 'Digital Media')
availability_status	VARCHAR	Item availability ('Available', 'On Loan', 'Reserved')

book_id	INT	Foreign key to Library_Item (Primary Key)
isbn	VARCHAR(20)	Unique ISBN of the book
author	VARCHAR(100)	Author of the book
genre	VARCHAR(50)	Genre of the book
publication_year	INT	Year the book was published
media_id	INT	Foreign key to Library_Item (Primary Key)
creator	VARCHAR(100)	Creator of the digital media
format	VARCHAR	Format of the media ('eBook', 'Audiobook', 'Video', etc.)
magazine_id	INT	Foreign key to Library_Item (Primary Key)
issue_number	INT	Magazine issue number
publication_date	DATE	Publication date of the magazine
borrow_id	INT	Unique identifier for each borrow transaction (Primary Key)
member_id	INT	Foreign key referencing Member
item_id	INT	Foreign key referencing Library_Item
staff_id	INT	Foreign key referencing Staff

borrow_date	DATE	Date the item was borrowed
due_date	DATE	Due date for returning the item
return_date	DATE	Actual return date (nullable)
fine_incurred	DECIMAL(5,2)	Fine incurred if returned late
reservation_id	INT	Unique identifier for each reservation (Primary Key)
member_id	INT	Foreign key referencing Member
item_id	INT	Foreign key referencing Library_Item
reservation_date	DATE	Date the reservation was made
expiry_date	DATE	Date the reservation will expire
payment_id	INT	Unique identifier for each payment (Primary Key)
member_id	INT	Foreign key referencing Member
amount_paid	DECIMAL(7,2)	Amount paid by the member
payment_date	DATE	Date of the payment
staff_id	INT	Unique identifier for each staff member (Primary Key)
name	VARCHAR(100)	Full name of the staff

role	VARCHAR	Role of staff ('Librarian', 'Administrator')
contact_info	VARCHAR(255)	Contact information (email or phone)
notification_id	INT	Unique identifier for each notification (Primary Key)
member_id	INT	Foreign key referencing Member
notification_date	DATETIME	Timestamp of when the notification was sent
notification_type	VARCHAR	Type of notification ('Reservation', 'Due Date Alert', 'Overdue Alert')

Normalization Considerations:

The relational schema developed for the Tech Titans Library Management System was derived from a well-structured ER model and adheres to Third Normal Form (3NF). Each relation has a clearly defined primary key, and all non-key attributes are fully functionally dependent on their respective keys. No transitive or partial dependencies exist. As a result, the schema minimizes data redundancy and maintains data integrity without requiring additional normalization steps. Most relations also satisfy Boyce-Codd Normal Form (BCNF), given their functional dependency structures.

5. Github Repository

- Link: https://github.com/aelxxs/tech-titans