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
- DVD
- Music
- User
- Member
- Staff
- Transaction
- Purchase
- Borrow
- Membership
- Author
- Genre
- Publisher

3. Define Attributes

Entity: Item

- Item ID - Char(10)
- Price - Int()
- Year - Numeric(4, 0)
- Availability Status - Enum('Available', 'Checked Out', 'Reserved', ‘Purchased’)
- 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)
- Name - Varchar(50)
- Publisher - Varchar(20)
- Edition - Int()
- Publish date - Char(10)
- Genre - Varchar(30)

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’)
- Genre - Varchar(30)

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)

Entity: Staff

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

Entity: Transaction

- Transaction ID: Char(10)
- Transaction Date: Date()
- Status: Enum('Completed', 'Pending', 'Canceled')

Entity: Purchase

- Transaction ID - Char(10)
- Purchase Amount - Decimal(10, 2)
- Payment Method - Enum(‘Credit Card’, ‘Debit Card’, ‘Cash’, ‘Other’)

Entity: Borrow

- Transaction ID - Char(10)
- Borrow Date - Date()
- Due Date - Date()
- Return Date - Date()
- Late Fee - Decimal(5, 2)

Entity: Membership

- Membership Type - Char(10)
- Borrow Limit - Int()
- Membership Fee - Decimal(3,2)
- Discount Rate - Decimal(2,2)

Entity: Author

- Author ID - Char(10)

- Name - Varchar(20)
- Biography - CLOB
- Date of Birth - Date
- Date of Death - Date

Entity: Genre

- Genre Name Varchar(20)
- Description - Char()
- Location in Library - Int() (each section of library could be divided by numbers)

Entity: Publisher

- Publisher ID - Char(10)
- Name - Varchar(20)
- Address - Varchar(30)
- Phone - Char(12)
- Email - Varchar (20)

4. Establish Relationships

Relationships:

- User “Performs” a Transaction:
    - Propose: Track who made all transactions
    - Cardinality:
        - That User may perform 0 or more transactions
        - That Transaction may be performed by 1 user
- User “Has” Membership:
    - Purpose: Track what users have what membership types
    - Cardinality:
        - That user may have 0 or 1 membership (staff has 0)
        - That membership type can be associated with 1 to many users
- Membership “Determines” a Transaction:
    - Purpose: Allows for transaction fees and rates to be determined based on users membership type
    - Cardinality:
        - That Membership type determines 0 to many transactions
        - That Transaction is determined by 1 membership type
- Transaction “Involves” an Item(s):
    - Purpose: Track which items are involved in a transaction
    - Cardinality:
        - That transaction can involve 1 or more items
        - That item can be involved in 0 or more transactions
- Author “Writes” an Item:
    - Purpose: Keep track of which items are written by what author(s)
    - Cardinality:
        - That Author can write 1 or more items
        - That Item can be written by 0 to 5 authors
- Genre “Belongs” to Item:
    - Purpose: Keep track of genres of specific items
    - Cardinality:
        - That Genre can belong to 1 or more items
        - That Item can belong to 0 or 1 Genre
- Publisher “Publishes” an Item:
    - Purpose: Keep track of who publishes what item(s)
    - Cardinality:
        - That Publisher can publish 1 to many items
        - That Item can be published by 0 or 1 publisher

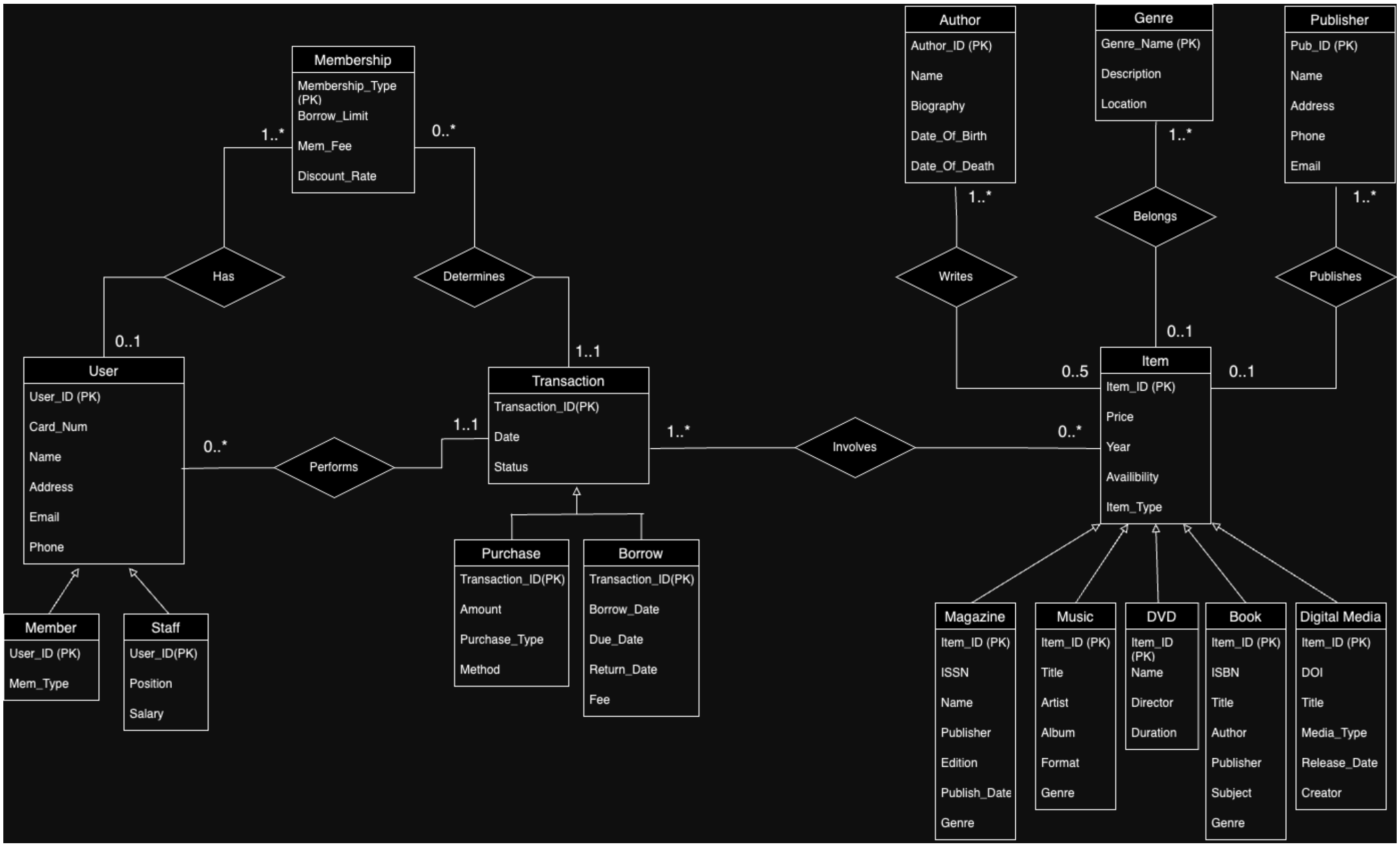
Specialization:

- 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 membership type has a borrowing limit defined and that borrowing limit must be checked for each user.
- 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.
- Due_Date attribute within Borrow not be NOT NULL
- Return_Date attribute within Borrow will be NULL until returned

5. Entity-Relationship Diagram (Primary Keys denoted by PK)


Membership
Membership_Type (PK)
Borrow_Limit
Mem_Fee
Discount_Rate
1..*
0..*
Has
Determines
0..1
1..1
User
User_ID (PK)
Card_Num
Name
Address
Email
Phone
0..*
Performs
1..1
Transaction
Transaction_ID(PK)
Date
Status
1..*
Involves
0..*
Member
User_ID (PK)
Mem_Type
Staff
User_ID(PK)
Position
Salary
Purchase
Transaction_ID(PK)
Amount
Purchase_Type
Method
Borrow
Transaction_ID(PK)
Borrow_Date
Due_Date
Return_Date
Fee
Author
Author_ID (PK)
Name
Biography
Date_Of_Birth
Date_Of_Death
1..*
Writes
0..5
Genre
Genre_Name (PK)
Description
Location
1..*
Belongs
0..1
Publisher
Pub_ID (PK)
Name
Address
Phone
Email
1..*
Publishes
0..1
Item
Item_ID (PK)
Price
Year
Availibility
Item_Type
Magazine
Item_ID (PK)
ISSN
Name
Publisher
Edition
Publish_Date
Genre
Music
Item_ID (PK)
Title
Artist
Album
Format
Genre
DVD
Item_ID (PK)
Name
Director
Duration
Book
Item_ID (PK)
ISBN
Title
Author
Publisher
Subject
Genre
Digital Media
Item_ID (PK)
DOI
Title
Media_Type
Release_Date
Creator