LibAlpha: A Digital Library

Alex Fritz, Althea Pamintawan, Sanjay DK, Victor Qiu — Database Management, Fall 2023

Entities and Fields

A. Books

ISBN (Primary Key)
Book-Title
Book-Author
Year-Of-Publication
Publisher
Image-URL-S
Image-URL-M
Image-URL-L

B. Users

User-ID (Primary Key)
Location
Age

C. Ratings

User-ID (Foreign Key, part of composite primary key)
ISBN (Foreign Key, part of composite primary key)
Book-Rating

D. Librarians

LibrarianID (Primary Key)
Name
Role

E. Signings

SigningID (Primary Key)
ISBN (Foreign Key)
Date
Time

F. Readings

ReadingID (Primary Key)
ISBN (Foreign Key)
LibrarianID (Foreign Key)
Date
Time

G. Fines

FineID (Primary Key)
UserID (Foreign Key)
ISBN (Foreign Key)
Amount
Status (Paid/Unpaid)
DueDate

H. Requests

RequestID (Primary Key)
UserID (Foreign Key)
BookTitle
RequestDate
Status (Pending/Fulfilled/Declined)

I. Copies

CopyID (Primary Key)
ISBN (Foreign Key)
Status (Available/Reserved/Checked Out)
UserID (Foreign Key, nullable)

J. Reservations

ReservationID (Primary Key)
UserID (Foreign Key)
CopyID (Foreign Key)
ReservationDate
ExpirationDate

Relationships

A. Books to Ratings

One-to-Many: Each book (identified by ISBN) can have many ratings, but each rating is associated with one book.

B. Users to Ratings

One-to-Many: Each user can give ratings to many books, but each rating is given by one user.

C. Books to Signings

One-to-Many: A book (identified by ISBN) can have multiple signing events.

D. Books to Readings

One-to-Many: A book (identified by ISBN) can be read in multiple reading events.

E. Librarians to Readings

One-to-Many: A librarian can host multiple reading events, but each reading is hosted by one librarian.

F. Users to Fines

One-to-Many: A user can have multiple fines, but each fine is associated with one user.

G. Books (via ISBN) to Fines

One-to-Many: A book can be associated with multiple fines (through its ISBN), but each fine is associated with one book.

H. Users to Requests

One-to-Many: A user can make multiple book requests, but each request is made by one user.

I. Books (via ISBN) to Copies

One-to-Many: A book can have multiple copies, but each copy is associated with one book.

J. Users to Copies

One-to-Many: A user can have multiple copies checked out or reserved, but each copy is associated with one user (nullable for available copies).

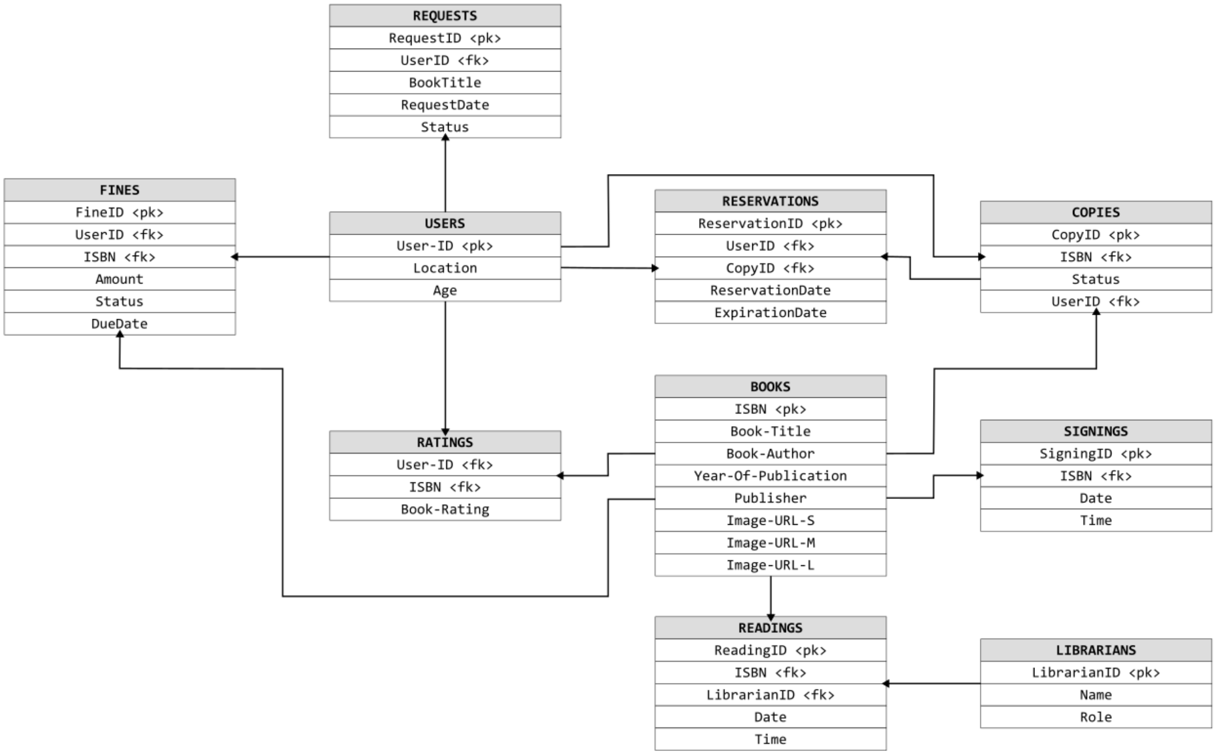
K. Users to Reservations

One-to-Many: A user can make multiple reservations, but each reservation is made by one user.

L. Copies to Reservations

One-to-One: Each copy can be reserved once at a time; this is a one-to-one relationship if only one active reservation is allowed per copy.

Entity Relationship Diagram



ER Model as Relations

Books: ISBN (PK), Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, Image-URL-L
Users: User-ID (PK), Location, Age
Ratings: User-ID (FK, PK), ISBN (FK, PK), Book-Rating
Librarians: LibrarianID (PK), Name, Role
Signings: SigningID (PK), ISBN (FK), Date, Time
Readings: ReadingID (PK), ISBN (FK), LibrarianID (FK), Date, Time
Fines: FineID (PK), UserID (FK), ISBN (FK), Amount, Status, DueDate
Requests: RequestID (PK), UserID (FK), BookTitle, RequestDate, Status
Copies: CopyID (PK), ISBN (FK), Status, UserID (FK, nullable)
Reservations: ReservationID (PK), UserID (FK), CopyID (FK), ReservationDate, ExpirationDate

Functional Dependencies

Books: ISBN → {Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, Image-URL-L}
Users: User-ID → {Location, Age}
Ratings: {User-ID, ISBN} → {Book-Rating}
Librarians: LibrarianID → {Name, Role}

Normalization

Our data, thanks to some strategic planning on our part, is already in 3NF. Each table has a primary key, all attributes are dependent on the primary key, and there are no *transitive* dependencies in any table. Huzzah!