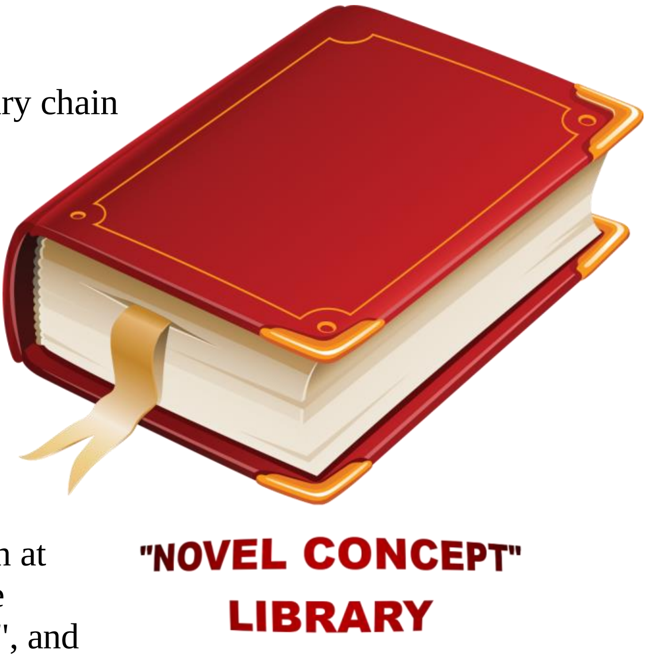
Hayden Mankin

Andrew Meade

CS3700

Schema S17

The database for our final project maintains a library chain with multiple locations. In total, there are eleven tables in the the database. The core functionality allows the user to not only keep track of every book in the library, but whether or not they have been requested, borrowed, or transferred, and which library member borrowed which book. Information about the books themselves (i.e. author, genre, and publisher) is kept in the database as well.



The "Book" table keeps records of every book with at least one copy in the library, using the ISBN as the primary key. Three more tables, "Author", "Genre", and "Publisher", all relate to "Book" in a one-to-many relationship (authors, genres, and publishers can have many books, but any given book can have only one author, genre, and publisher). We make the assumption that, for books that have more than one author, only one "primary" author will be tracked for simplicity's sake.

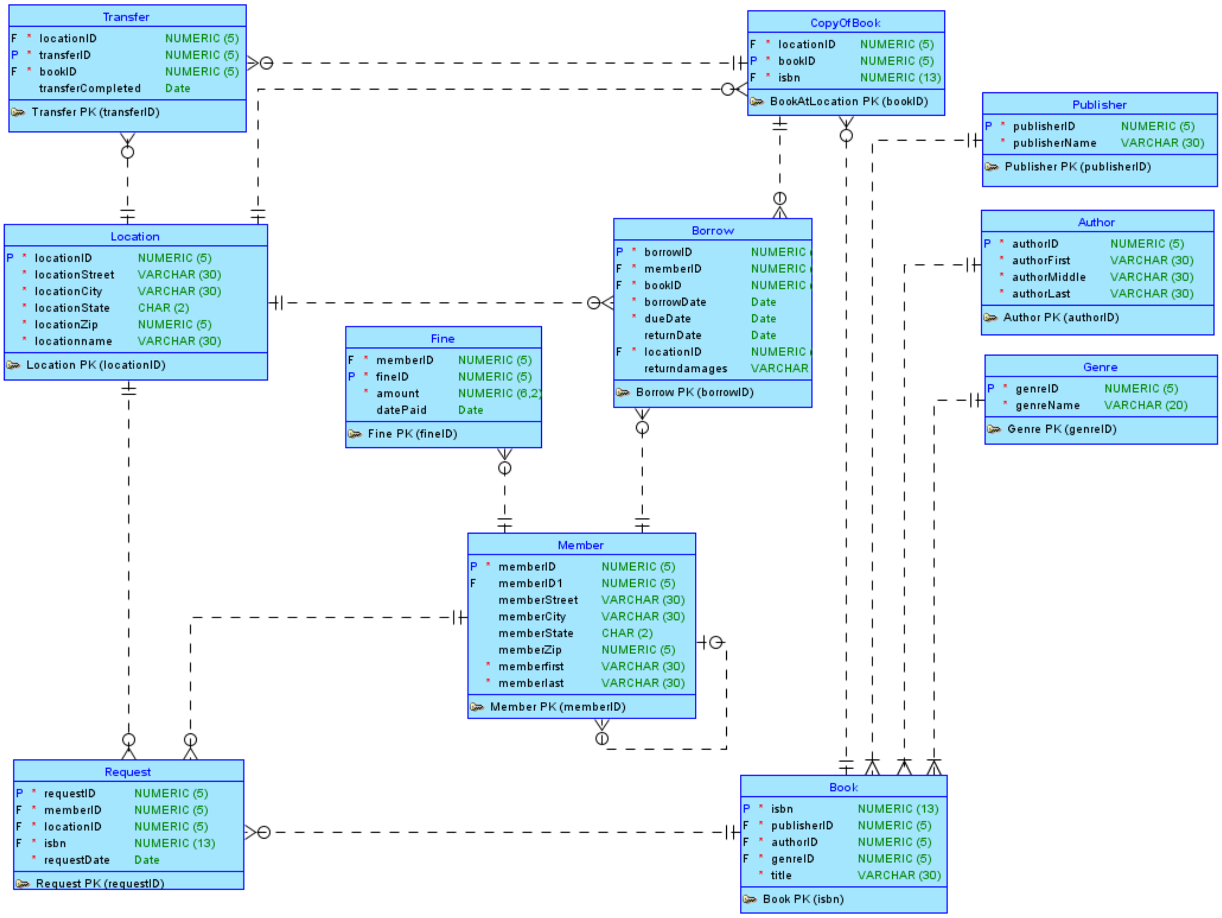
The copies of the books are recorded in the "CopyOfBook table". A book may have any number of copies, but each copy can only be of one book, making this a one-to-many relationship. A copy may be borrowed multiple times by multiple members of the library. This is reflected in the "Borrow" table, which maintains a many-to-many relationship between "CopyOfBook" and "Member", and also keeps track of due dates. When a book is not returned on time, a record should be created in the "Fine" table. A fine corresponds to a single member, and tracks the amount due and date paid, if applicable.

A book copy can also be transferred to multiple different locations multiple times. Transfers are tracked in the "Transfer" table, which maintains a many-to-many relationship

between "CopyOfBook" and "Location".

The final table, "Request", is intended for use by library members to ask to borrow a book. It has three foreign keys, one to the member who made the request, another to the book being requested, and another to the location where a book copy may have to be transferred if the request is fulfilled.

One last thing to note is that a member record can relate to another member record in a one-to-many relationship via the sponsorID. What this means is that, if a member with a valid sponsorID generates a record in the "Fine" table, it will instead be due from the sponsor rather than the member who caused the fine



Schema

Only attributes in red may be NULL, all others have the constraint NOT NULL

Author(authorID,authorFirst,authorMiddle,authorLast)

Book(isbn,*publisherID*,*authorID*,*genreID*,title)

Borrow(borrowID,*memberID*,*bookID*,borrowDate,dueDate,
returnDate,*locationID*,returndamages)

CopyOfBook(bookID,*locationID*,isbn)

Fine(fineID,memberID,amount,datePaid)

Genre(genreID,genreName)

Location(locationID,locationStreet,locationCity, locationState,locationZip)

Member(memberID,*sponsorID*,memberStreet,memberCity,
memberState,memberZip,memberFirst,memberLast) --the FK "sponsorID" points to
another member record

Publisher(publisherID,publisherName)

Request(requestID,*memberID*,*locationID*,isbn, requestDate)

Transfer(transferID,*locationID*,*bookID*, transferCompleted)