Data was given in .csv format in two files- one for the borrowers and one for the books. The information for the borrowers, was pulled directly from the file and put into a table named BORROWER in the library database as is. However, the book information had to be normalized prior to storing the information in the database. The book.csv data has an authors column that has repeated author names for different books that could be grouped as distinct author names to avoid this wasted disk space. Therefore, a relationship can be crafted between a list of books and their authors to normalize the data and create a more flexible database.

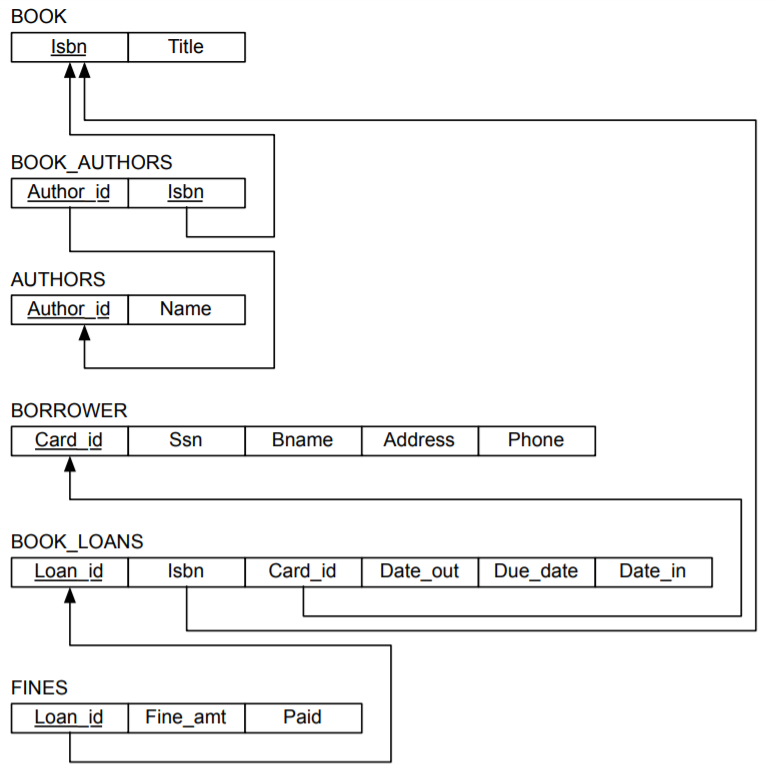
A book is composed of author name(s) and a book id along with other book details such as number of pages. To normalize the data, the authors column was separated from the book information and three tables were created in the database named BOOK, AUTHORS, and BOOK\_AUTHOR. AUTHORS is a table with all the unique authors found and their respective ids, BOOK is a table with all the unique books found and their respective ISBNs, and BOOK\_AUTHOR is the key for joining these two tables- it states which author ids match with which ISBNs.

At first, all the book.csv data is pulled into the BOOK table, including the author names, and then each distinct author name is selected and placed into the AUTHOR table where each author receives a unique id- this is the primary key for the author table. Then the BOOK table is altered to remove the author column so that it holds all the book details that are not authors and also adds the column of isAvailable which is set to 1 by default and changes to 0 if a book is ever checked out. The BOOK table has a primary key of ISBN10.

The BORROWER table holds all the unique borrowers that have registered to use the library system. Every time a new borrower registers, they will receive an automatically generated and unique borrower id which is the primary key to this table. Even though the database will allow for a manual insertion of two borrowers with the same SSN but different borrower ids, through software verification, a borrower with the same SSN cannot exist in the database. Another added implementation is the flag column which is 0 by default, but can change to 1 if the borrower has more than three books checked out or if a fine of greater than $5.00 is owed. More criteria could be added as required.

A table named BOOK\_LOAN was created to store the loan information. This includes an automatically generated and unique loan id, the ISBN of the book and the borrower id it pertains to, the date it was checked out the due date and the date it was checked back in. The date out and date in are set by the MySQL system date, the due date is by default 14 days after the date out. When a book is checked out, a new loan is created with a new loan id and when a book is returned, the date in is updated and the loan is no longer active. Only one book exists per loan.

A FINE table was created for any book loans that were/are out after the due date. Every time the database is refreshed, if there is a loan that is still out or was checked in past the due date a new row is created in the fine table with the associated loan id as the primary key, the fine amount, and a paid column. Through software implementation, the fine amount accrues $0.25 every day that the book is checked out past the due date and the paid column is set to 0 by default but can change to 1 if the fine amount is paid in full by the borrower. No fine can be paid if the associated book is not checked back in.



ISBN10

ISBN13

title

publisher

cover

pages

isAvailable

ISBN10

author\_id

author\_id

name

borrower\_id

ssn

first\_name

last\_name

email

address

city

state

phone

flag

loan\_id

ISBN10

borrower\_id

date\_out

due\_date

date\_in

loan\_id

fine\_amt

paid

Figure 1: Schema of LIBRARY database