- Programming language used: Java
- Database used: MySQL
- My folder includes a file named “SQL Schema creation file” which has all the ‘create table’ and ‘insert into’ commands I used for my database.
- **BOOK SEARCH AND AVAILABILITY**

  You can search for a book, given any combination of book_id, title, company name and/or Author, which may be either author_name or any combination of parts of an author's name (i.e. First name and/or Last name). When you click on the search button, a window opens up displaying book_id, title, author_name, branch_id, Number of copies at each branch, Number of available copies at each branch.

- **BOOK LOANS**

Checking Out Books

You can check out a book, given the combination of BOOK_COPIES(book_id, branch_id) and BORROWER(Card_no), i.e. a new tuple is created in BOOK_LOANS. A new unique primary key is generated for loan_id. The date_out is today’s date. The due_date is 14 days after the date_out. Each BORROWER is permitted a maximum of 3 BOOK_LOANS. If a BORROWER already has 3 BOOK_LOANS, then the checkout *fails* and returns an error message “Card no: (the corresponding card number) already has 3 books checked out. No more books can be checked out! “ . If the number of BOOK_LOANS for a given book at a branch already equals the No_of_copies (i.e. there are no more book copies available at the library_branch), then the checkout *fails* and returns an error message “No more copies of Book_id (the corresponding book_id) are available at branch (branch_id)”.

Checking In Books

On the Main GUI, if you click on the Check-in button, a new frame opens up. You can check in a book. You can locate BOOK_LOANS tuples by searching on any of book_id, Card_no, and/or any part of BORROWER name. So you can enter any of these value, and click the “OK” button and the BOOK_LOANS tuples will show up in the box below. And then you can select the appropriate tuple among potentially multiple results and then when you click on the “Update” button, the book will get checked in and the date_in value will be updated for the corresponding BOOK_LOANS tuple.

- **BORROWER MANAGEMENT**

  You can create new borrowers in the system. All name and address attributes are required to create a new account, so enter all these values and click on the add button. A new card_no primary key is automatically generated for each new tuple because I have put AUTO_INCREMENT for loan_id. Borrowers are allowed to possess exactly one library card. If a new borrower is attempted with the same fname, lname, and address, then the system rejects and returns an error message.

- **FINES**

I have created a new table FINES(loan_id, fine_amt, paid). The primary key loan_id is also a foreign key that references BOOK_LOANS(loan_id). fine_amt attribute is a dollar amount that has two decimal places. paid attribute is an integer 0/1 that indicates whether a fine has been paid. When you click on the Fines button, a new window opens up. In the new window, if you click on the Refresh button, the fines table gets updated and displays a message “Refreshed Successfully ! “ , and then another window opens up, you can enter a card no. here and when you click on the Pay fine button, the fine is paid and the paid attribute in the fines table is updated to 1. Payment of a fine is not allowed for books that are not yet returned. So if a book hasn’t been checked in yet and you click on the pay fine button, nothing happens. Also fines is displayed for each book that has been borrowed by that particular user.