

**The following are the tables used in the Library management system:**

**Tables:**

**1.Users:**

This is the user table that holds the information about all the valid library users. Each user has a unique userID.

**Primary Key - ( userID )**

- userID (Primary Key)
- Name
- password
- address
- profession

**2.Librarians:**

This holds the information about all the librarians who manage the library. They have a unique LibrarianID.

**Primary Key - ( librarianID )**

- librarianID
- Name
- Password
- address

**3.Books:**

This table contains the information about all the books that are available in the library along with the information of the book.

Each different book has a unique ISBN number and all similar books have different copy number

**Primary key - ( ISBNnumber , copyNo )**

**Foreign Key - (shelfID from table shelf)**

- ISBNnumber
- copyNo
- Title
- Publication\_year
- shelfID
- current\_status

#### **4.Authors:**

This has the information related to authors like authorID, their names and a little of description related to their works.

##### **Primary Key - (authorID)**

- authorID
- name
- details

#### **5.Book\_authors:**

This table is a relational table for the books and authors. It contains the bookID and AuthorID, gives us information about the authors of each book.

##### **Primary Key - (bookID , authorID)**

##### **Foreign Key - (bookID from books table, authorID from authors table)**

- bookID
- authorID

#### **6.Shelf:**

This table accounts for the information related to shelves containing books, for instance the capacity of a shelf,shelfID etc..

##### **Primary Key -(shelfID)**

- shelfID
- capacity
- shelfcol

#### **7.Ratings&Reviews:**

This holds the ratings and reviews of the books including the information of the book being rated/reviewed and the user performing the action.

##### **Primary Key - (user\_ID,book\_ID)**

- user\_ID
- book\_ID
- Rating
- review

### **8. PersonalBookShelf:**

This is the list of users and the books they added to their personal bookshelf.

**Primary Key** - ( user\_id, book\_id )

**Foreign Key** - (user\_id from user table, book\_id from book table)

- user\_id
- book\_id

### **9. Category:**

This table contains all the categories which are available in the library

**Primary Key** - ( Category\_name, book\_id )

**Foreign Key** - ( book\_id from books table )

- bookISBN
- Category\_name

### **10. BorrowedBooks:**

This table contains all the books that are issued from the library to users

**Primary Key** - ( user\_id, book\_id )

**Foreign Key** - ( user\_id from user table, book\_id from book table )

- user\_id
- book\_id
- copy number
- issue date
- status
- due\_id

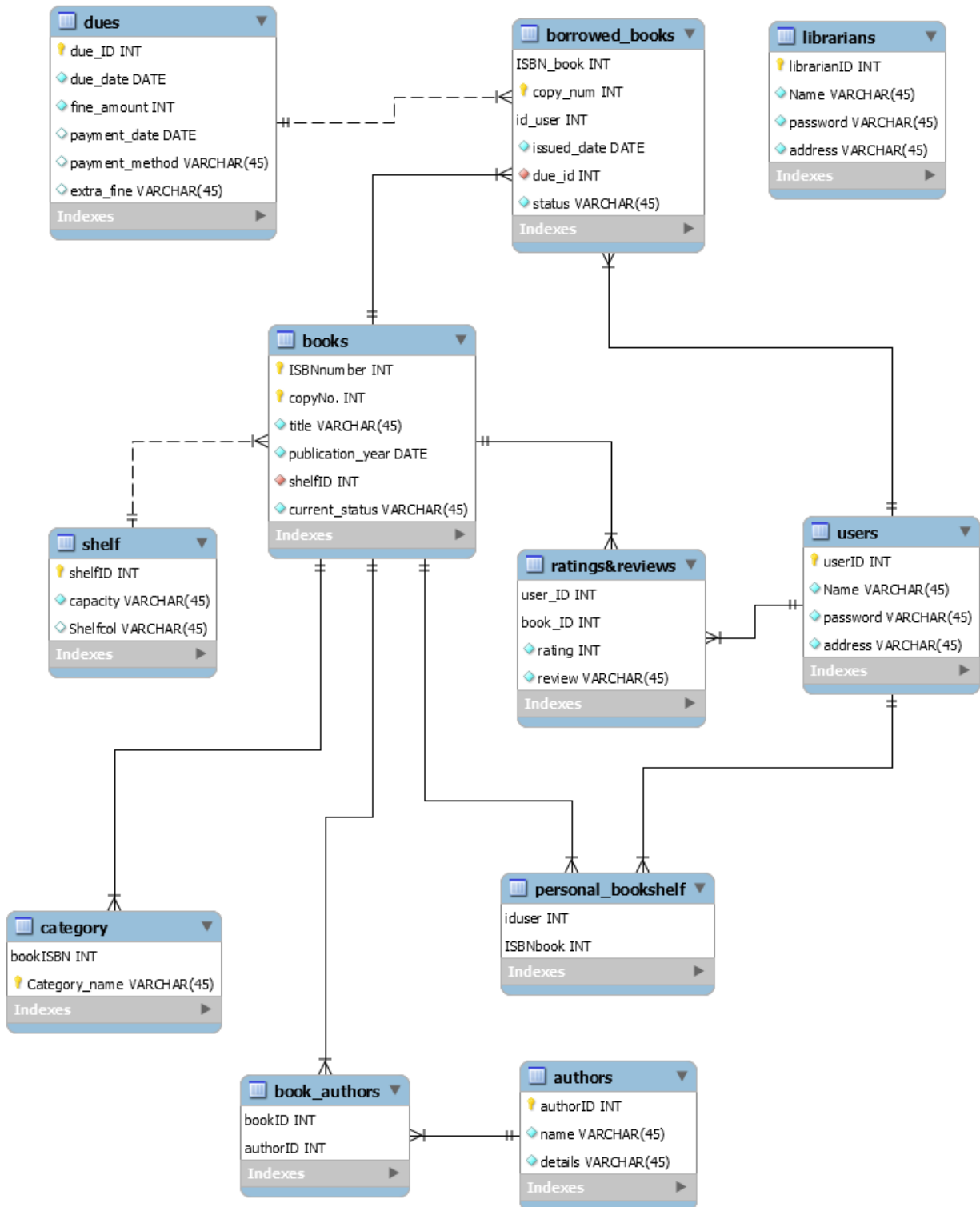
### **11. Dues table:**

This table contains all the dues and their payment modes

**Primary Key** - ( Due\_id )

- Due\_id
- Due\_date
- Fine amount
- Extra fine
- Payment date
- payment\_method

## ER from Mysql workbench:



## **Relationships among the tables:**

### **1.user table:**

In the user table, we have all the required details of the users along with their login details. We have 2 kinds of users: students and faculty, in the last attribute we can fill that.

### **2.Users to books\_borrowed:**

In this, the list of users who borrowed the books from the library. It contains the book issued and the status of the book, including all the remaining information about book issue.

### **3.Books\_borrowed to dues:**

Books\_borrowed table also has a foreign key dueID, referencing the **dues table**. Where the dues table contains the complete information about each due. Date of payment, mode of payment, extra fines if applicable etc.

### **4.Books to Authors:**

As some books could have multiple authors, we cannot use authorID in the books table. The relation between them is **many to many**. So, we need to make one more table to make this relationship. We have the **Book\_Author** table which contains the **bookID** and **authorID** as attributes. With both of them together as the primary key.

### **5.Books to category:**

Here a book may fall under different categories and in a category, there can be a lot of books. So, this is a **many to many** relationship. That's why we made a new table **category** where we keep the list of books and their categories.

The primary key is bookID and category combined.

### **6.Book to shelf:**

There are different shelves with unique ids. Each shelf has a different capacity. One book can be kept on only 1 shelf but a lot of books can be kept on 1 shelf. So, it is **many to one** relationship. Therefore, we keep shelfID as the foreign key in the **books** table referencing the **shelf** table.

**7.Books-Users-Reviews and ratings:**

In this library management system, the user has the chance to give a rating and review of the book issued by him. It contains bookID, userID of the user and the review along with rating.

**8.User to Personal bookshelf:**

This is the personal bookshelf that each user maintains on his own. It is just like a wishlist where the user can add his favourite books for later reverences and also maybe for later issuing.