Semester Project - Phase I

Neema George, Keshava Pranath Kakekochi, Shashank Shankaregowda

CSE560 - Data Model Query Language, Semester Project
Team BrightStars
Project Name: Library Management System
UBID: neemageo
UBID: keshavap
UBID: ss675

***Problem Statement*— The library has tons of data that has to be managed on a day-to-day basis. For example, there are thousands of books with data like book title, author, publisher, number of copies, etc. Also, every transaction of books with the user has been recorded, which is a huge amount of data. Maintaining this amount and kind of data with just an Excel sheet will be a terrible idea because the data has a lot of constraints to manage the function of the library smoothly. Having the whole system on the database lets the user know about the books available for them and lets the library staff manage the library operations better.**
In the solution we are proposing as the library management system, we will provide the user with a database that can handle all the details regarding the books, users, and the transaction of the books when any user borrows a book from the library. This will be supported with a React front-end application connected with NodeJS backend to make the library management an easier task for the library staff as well as the users.
The books table will contain all the details about the book, which can be updated when new books are purchased by the library. The transaction table will note down every transaction of books that happens between the user and the library.

***Keywords*— Include at least 5 keywords or phrases**

I. Introduction

This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

II. Target User

The library management system is designed primarily for the general public, enabling them to reference and borrow books. Five selected administrators are responsible for executing tasks such as modifications and updates to the book database. Other users do not possess the privileges to alter the Books database. In a real-life scenario, we can consider a public library in a city. The library is frequented by a diverse group of people, including students, researchers, and leisure readers, who utilize the system to search for and check out books. The five administrators are library staff members who have been trained to handle the database. In the event of a system update or the need to change the library's policy on borrowing limits, these admins would be responsible for implementing these changes in the database. Every user has a unique record with an email address and password that is protected by SHA-256. With the correct details, they can borrow a book which will be issued by the admin, and are expected to return it by the due date.

III. ER Diagram

A. Relational Schema

Admin (Admin_ID: int, Name: varchar, Email: varchar, Password: varchar, Created_At: date)

Books (Book_ID: int, Title: varchar, Average_Rating: float, ISBN13: varchar, Language_Code: varchar, Number_of_pages: int, Rating_Count: int, Publication_Date: date, Publisher: varchar, Genre: varchar, Book_Count: int)

Author (ID: int, Author_name: varchar, Book_ID: int)

Location (Book_id: int, Aisle: char, Rack: int, Count_Left: int)

Transaction (Transaction_ID: int, User_ID: int, Book_ID: int, Borrowed_Date: date, Return_Due_Date: date, Returned_Date: date)

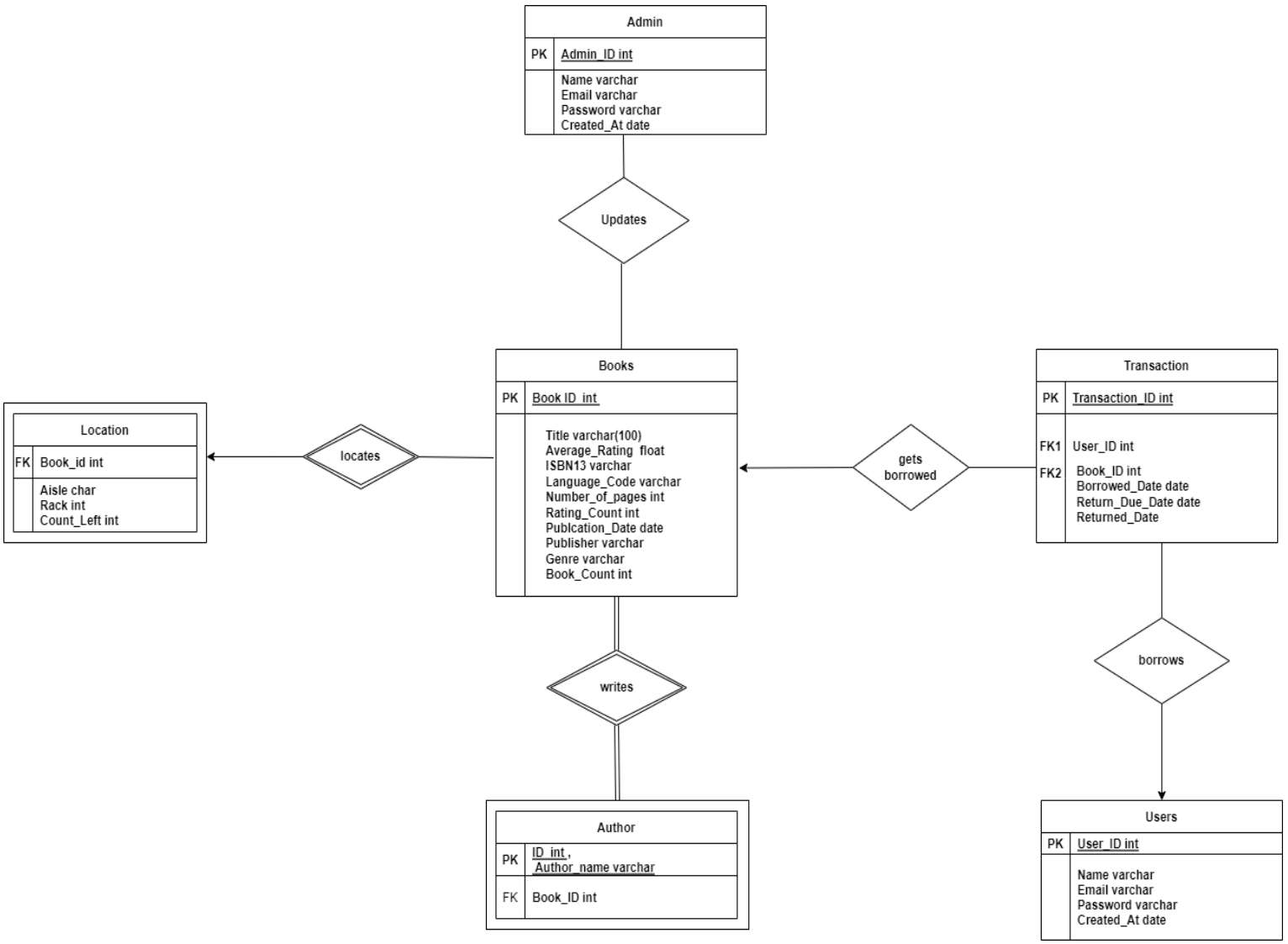

Fig. 1 ER Diagram

Users (User_ID: int, Name: varchar, Email: varchar, Password: varchar, Created_At: date)

IV. Table Description

We have 6 tables in total after normalization. Let's start with the books table. In this table, we have only the Book_ID column with NOT_NULL.

Book_ID	varchar	This is a primary key. It will note the ID of each book
Title	varchar	This attribute will be the title of the book
Average_rating	float	It will have the rating of the book.
ISBN13	varchar	It is an International standard book serial number
Language_code	varchar	It represents the language of book
Number_of_pages	int	Number of pages in the book
Rating_Count	int	Number of rating present for this book which led to

		average rating.
Publication_date	date	Date on which the book is published
Publisher	varchar	Publisher of the book
Genre	varchar	Gernre of the book
Book_count	int	Number of books the library owns

Next is the Author table. In this table, the ID and the Author_name columns together form the primary key.

ID	int	Id and Author_name column combined a primary key.
Author_name	varchar	Name of the author for the book id
Book_ID	int	This is the foreign-key from the books table

We also have a Location table which holds the location of the book in the library.

Book_ID	int	Foreign key from Books table
Aisle	char	Character from A to Z
Rack	int	Rack number to identify the location of the book.
Count_left	int	Count of the books left in the rack.

Next, we have a User table that holds user details.

User_ID	int	Primary key, Id of the user
Name	varchar	Name of the user
Email	varchar	distinct email ids of the user
Password	varchar	Password of the user will be stored as a hash value using the SHA256 hashing algorithm
Created_at	date	This attribute tells when the user was added to the system and this will be added automatically to the system during the insert query execution

We are also maintaining Admin table to manage the system who has the authority to update the data in the database. All the attributes are similar to User table, just that the admin table has Admin_ID instead of User_ID.

Lastly, we have a transaction table that holds the record of every book transaction that happens in the library. Following are the details.

Transaction_ID	Varchar	Primary key, Storing an id of every transaction.
User_ID	Varchar	Foreign key referring the User_ID from User table
Book_ID	Varchar	Foreign key referring the Book_ID from Books table
Borrowed_Date	date	Automatically populated the current date while inserting to note the date the book is borrowed.
Return_Due_date	date	Automatically populates a date 15 days from the current date as the due date for the book to be returned.
Return_date	date	Once the user returns the book, this data gets populated as the returned date for the book.

The Return_date in the transaction table will be initially Nul value as the data can be populated only when the book is returned. And for the primary keys, the data will be automatically populated by the database on the set rules which makes it impossible to be left NULL with value.

V. Task 3 and Task 4

We have downloaded the book dataset from Kaggle. The dataset contained the following columns: 'BookID', 'Book Title', 'Authors', 'avg_rating', 'ISBN', 'ISBN13', 'Language_code', 'Rating_count', 'publication_date', 'publisher', 'text_review_count'. From this dataset, we removed the 'text_review_count', and 'ISBN' columns and added the 'Genre' and 'book_count' columns by using a Python script which is included in the submission.

We generated user data using a Python script which is also included in the submission.

We have 'Name', 'Email', 'Password', and 'created_at' columns in this table. A password will be hashed using the sha256 hashing algorithm. Similarly, we have an admin table with the same columns as the user table for the list of administrators to manage the database and the application.

We also have a Location table which helps locate books in the library based on the aisle and rack. We have the library set up in 26 aisles with 30 racks in each aisles which holds 500 books in each rack.

In the Books relational table, we observe that the 'Authors' attribute has multiple values in one cell which is a violation of the first normal form. Hence we create a separate table for authors with the columns 'ID', 'Author_Name', and 'Book_ID' with Book_ID as the foreign key referencing the Book table and ID as the primary key. It is now in BCNF. In every other relational table, we observe that each non-key attribute is dependent only on the candidate key or super key. Hence they are in BCNF.

References

[1] BooksDataset.[online].Available: https://www.kaggle.com/datasets/saurabhbagchi/books-dataset

[2] Avi Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, Seventh, Edition, McGraw-Hill, 2019

[3] H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems: The Complete Book, Second Edition, Prentice Hall, 2009.