Library Management System

By
Reema Mehta
(Data Management and Database Design)

Problem Statement

This project is focused on the current library management system of Sardar Patel Institute of Technology. Currently librarians do have system but more of manual and keeping the records of data in record book. Process is quite cumbersome and time consuming for students.his system will give enhancement to current system and involves more technology for making management more easier.

Background

Since a considerable measure of books are issued and returned every day at the library, the framework is significantly information subordinate. The present arrangement that is being executed by the Sardar Patel Institute of technology is in part on the web yet predominantly manual. The present framework offers understudies an online entryway to see the accessible books, yet these passages are made physically by the bookkeeper as and when he discovers time. Thus, the framework isn't extremely state-of-the-art to the extent stock records are concerned. Likewise, this framework is exceptionally inclined to manual blunders and physical printed material is dependably in threat of being stolen or decimated by flame, and so on. The present framework is additionally exceptionally work concentrated.

The library at Sardar Patel Institute of technology, University of Mumbai, is a colossal library with books from no less than, 4 designing branches. The books in this library must be acquired by understudies and workforce at Sardar Patel Institute of technology who have been issued a library card by the school. A man can just get two books at any given moment. Each book obtained is expected in 21 days. Returns made after due date are fined Rs. 5/day

Users for LMS

Admin - The one who have overall access of the system and can add/modify/delete both borrower and librarian’s data

Borrower(Students) - The one who are going to access system for checking books online , borrowing and returning books

Librarian - The one who can add/modify/delete borrower accounts. Librarians can add a new book, modify existing information such as shelf number or publication, etc. or delete book. Librarians can see details of a borrower by input of their name or library card number.

Entities and Relationships

Publisher	**Stores all publishers records**
publisher_id publisher_name publisher_officeAddress	PRIMARY KEY: Each publisher will have a unique ID to identify publisher
Book	**Stores all the book details**
book_id book_title book_ISBN publisher_id release_date book_backCoverText book_comments flag_reserved	PRIMARY KEY: Each book will have a unique ID to identify the book FOREIGN KEY: Associated with primary key of 'Publisher' table The descriptive text given on the back of the book If the book is not supposed to be issued, flag is true.
UserRoles	**Stores the type of users.** **Reason for making a separate table is to provide future flexibility. New users roles can be added easily in future.**
user_typeId user_typeDesc	PRIMARY KEY: Each type of user is given a unique ID Contains each role of user. [admin, librarian or borrower]
User	**Contains all types of user information**
user_id user_firstName user_lastName user_email user_typeId	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: Associated with Primary key of 'UserRoles' table. Determines the role. Possible values : 1,2 or 3
Admin	**Child entity of User. Contains admin information**
user_id admin_officeExtension	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: associated with primary key of 'User' table.

Librarian	**Child entity of User. Contains librarian information**
user_id librarian_officeExtension	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: associated with primary key of 'User' table.
Borrower	**Child entity of User. Contains Borrower's information. Faculty of the college and students have the same privileges in the designed system, so distinction has not been made between the types of borrowers**
user_id user_createdById user_libraryCardNumber	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: associated with primary key of 'User' table. FOREIGN KEY: associated with primary key of 'Admin' table and primary key of 'Librarian' table
BorrowerRecord	**Maintains records of what books are borrowed by what users. Since many borrowers can borrow many books, this association table was created to avoid multivalued attributes.**
record_id book_id date_borrowed flag_returned	PRIMARY KEY: Auto incremented Integer identifying each record. FOREIGN KEY: associated with primary key of 'Book' table Boolen. Will be false by default. True when the book is returned.
UserLogin	**Contains login credentials for users.** **Reason for making a separate table: Provides better security.**
user_id login_username login_password	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: associated with primary key of 'User' table.
Shelf	**Contains information of shelves where books are kept**
shelf_id shelf_label shelf_floor	PRIMARY KEY: Each user has a unique shelf ID to identify name the shelf is known by floor where the shelf is located
Compartment	**Contains information of each compartment is a shelf**
compartment_id shelf_id	PRIMARY KEY: Integer identifying compartments of shelf FOREIGN KEY: associated with primary key of 'Shelf' table.
BookLocation	**Maintains records of where the books are located. It is created as a separate table because locations constantly get updated. A separate table would be more efficient.**
book_id compartment_id	PRIMARY KEY: Integer identifying each book individually FOREIGN KEY: associated with primary key of 'Book' table FOREIGN KEY: associated with primary key of 'compartment' table.

Author	**Maintains record of authors of the book**
author_id author_firstName author_lastName author_stageName author_email	PRIMARY KEY: Integer identifying each author uniquely Some authors like to have a different name displayed on book
BookAuthor	**Maintains records of which books are written by what authors. Sometimes books can have multiple authors so this association table was created to avoid multivalued attributes**
record_id book_id author_id	PRIMARY KEY: Integer identifying each record. Auto incremented field. FOREIGN KEY: associated with primary key of 'Book' table. FOREIGN KEY: associated with primary key of 'Author' table.

REFERENCES

https://www.tutorialspoint.com/mysql/mysql-delete-query.htm

https://www.w3schools.com

"Beginning Database Design Solutions"

"Beginning SQL"