Library Book Borrowing System

Basic CRUD Operations with MongoDB

Project Overview

This presentation introduces a straightforward library system, enabling students to efficiently borrow and return books.

Student-Centric

Designed for ease of use by students.

Book Management

Comprehensive tracking of book availability.

Core Data Collections

Manages Students, Books, BorrowRecords, and Librarians.

CRUD Functionality

Supports fundamental Create, Read, Update, and Delete operations.

Key Objectives

Our system focuses on maintaining accurate records and streamlining the borrowing process.

Maintain Accurate Book Records

Ensure all book details are current and easily accessible.

Track Student Borrowings

Monitor which student has borrowed which book at any given time.

Record Dates

Store precise borrow and return dates for all transactions.

Prevent Duplicate Borrowings

Implement checks to ensure a student cannot borrow the same book twice simultaneously.

Clean Database Structure

Maintain an organised and efficient database for optimal performance.

System Features

The system provides a suite of features to manage library operations effectively.

Add New Students

Register new student profiles with essential contact information.

Add & Update Books

Easily add new titles or modify existing book details and availability.

Borrowing Books

Streamlined process for students to check out books.

Returning Books

Effortless procedure for students to return borrowed items.

View All Borrow Records

Access a comprehensive log of all borrowing transactions.

MongoDB Collections Overview

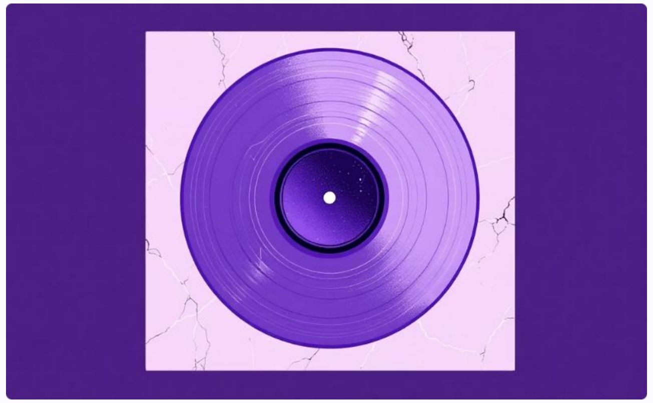
Our system utilises four distinct MongoDB collections to store and manage library data.

Students

Stores student details: `studentId`, `name`, `class`, `phone`.

Books

Contains book information: `bookId`, `title`, `author`, `genre`, `availableCopies`.



BorrowRecords

Tracks borrowing activities: `recordId`, `studentId`, `bookId`, `borrowDate`, `returnDate`, `status`.

Librarians

Librarian profiles: `librarianId`, `name`, `phone`.

Core CRUD Operations

Understanding the fundamental MongoDB operations that power our library system.

1

Create

Insert new documents into collections (e.g., adding a new student or book).

2

Read

Use the `find()` method to retrieve specific data (e.g., searching for a book).

3

Update

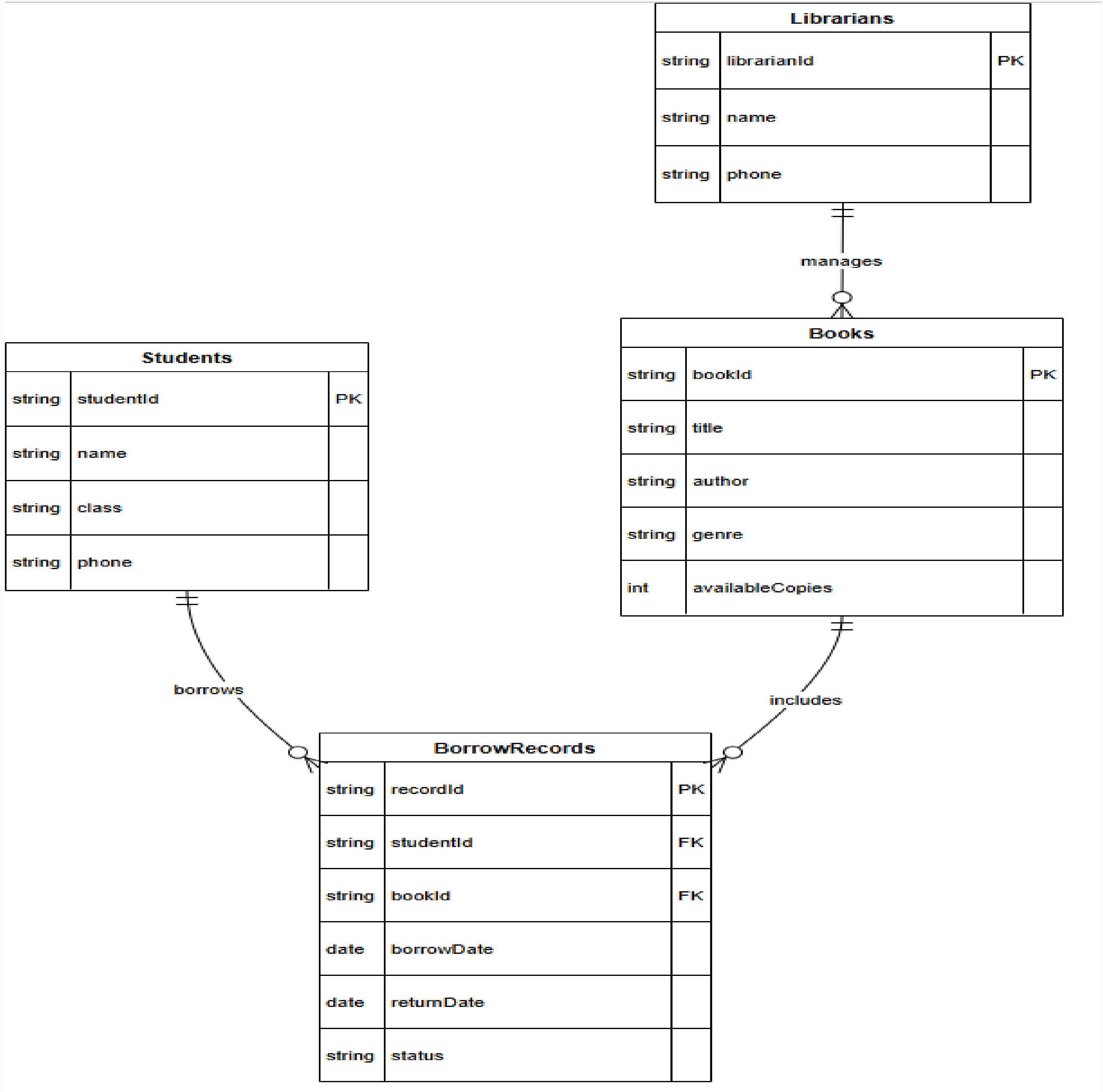
Employ `$set` to modify fields or `$inc` to increment values like available copies.

4

Delete

Utilise `deleteOne()` to remove documents, such as a completed borrow record.

Entity-Relationship Diagram



Librarians

string	librarianId	PK
string	name	
string	phone	

Books

string	bookId	PK
string	title	
string	author	
string	genre	
int	availableCopies	

Students

string	studentId	PK
string	name	
string	class	
string	phone	

BorrowRecords

string	recordId	PK
string	studentId	FK
string	bookId	FK
date	borrowDate	
date	returnDate	
string	status	

This diagram illustrates the relationships between the different entities within our library database, highlighting how students interact with books through borrowing records, and how librarians manage the book collection.

Flow of Borrowing a Book

A step-by-step process outlining how a student can successfully borrow a book from the system.

Student Selects Book

The student identifies the desired book.

Check Availability

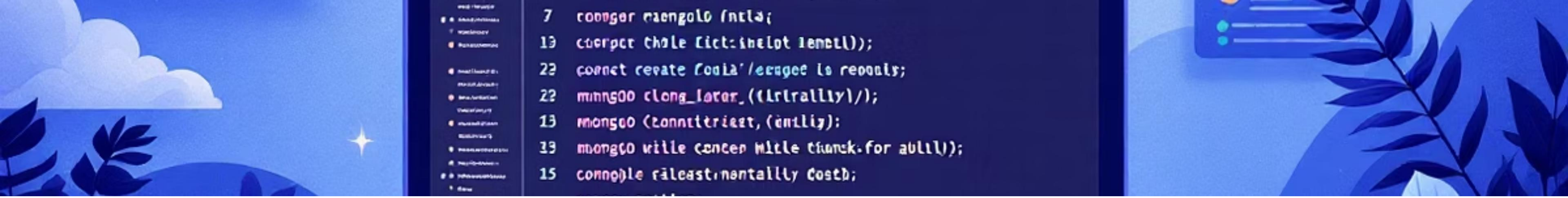
The system verifies if the book has available copies.

Create Borrow Record

A new record is generated, linking the student and the book.

Reduce Available Copies

The count of available copies for the book is decremented by one.



Simple MongoDB Example

Illustrative code snippets showcasing basic MongoDB operations for our library system.

Adding New Books

```
db.books.insertMany([  { bookId: "B001", title: "The Great Novel", author: "A.
Author", genre: "Fiction", availableCopies: 5 },  { bookId: "B002", title:
"Tech Guide", author: "B. Coder", genre: "Technology", availableCopies: 3 }]);
```

Creating a Borrow Record

```
db.borrowRecords.insertOne({  recordId: "BR001",  studentId: "S101",  bookId:
"B001",  borrowDate: ISODate("2023-10-26"),  returnDate: null,  status:
"borrowed"});
```

Updating Available Copies after borrowing:

```
db.books.updateOne( { bookId: "B001" }, { $inc: { availableCopies: -1 } });
```

Conclusion

This system lays a robust foundation for a scalable library management application.

Basic Working System

Successfully implements core library functionalities.

CRUD Operations Completed

Full Create, Read, Update, and Delete capabilities are in place.

Expandable for Future Growth

Ready for additional features and full application development.