UNIVERSITY LIBRARY DATABASE SYSTEM PROPOSAL

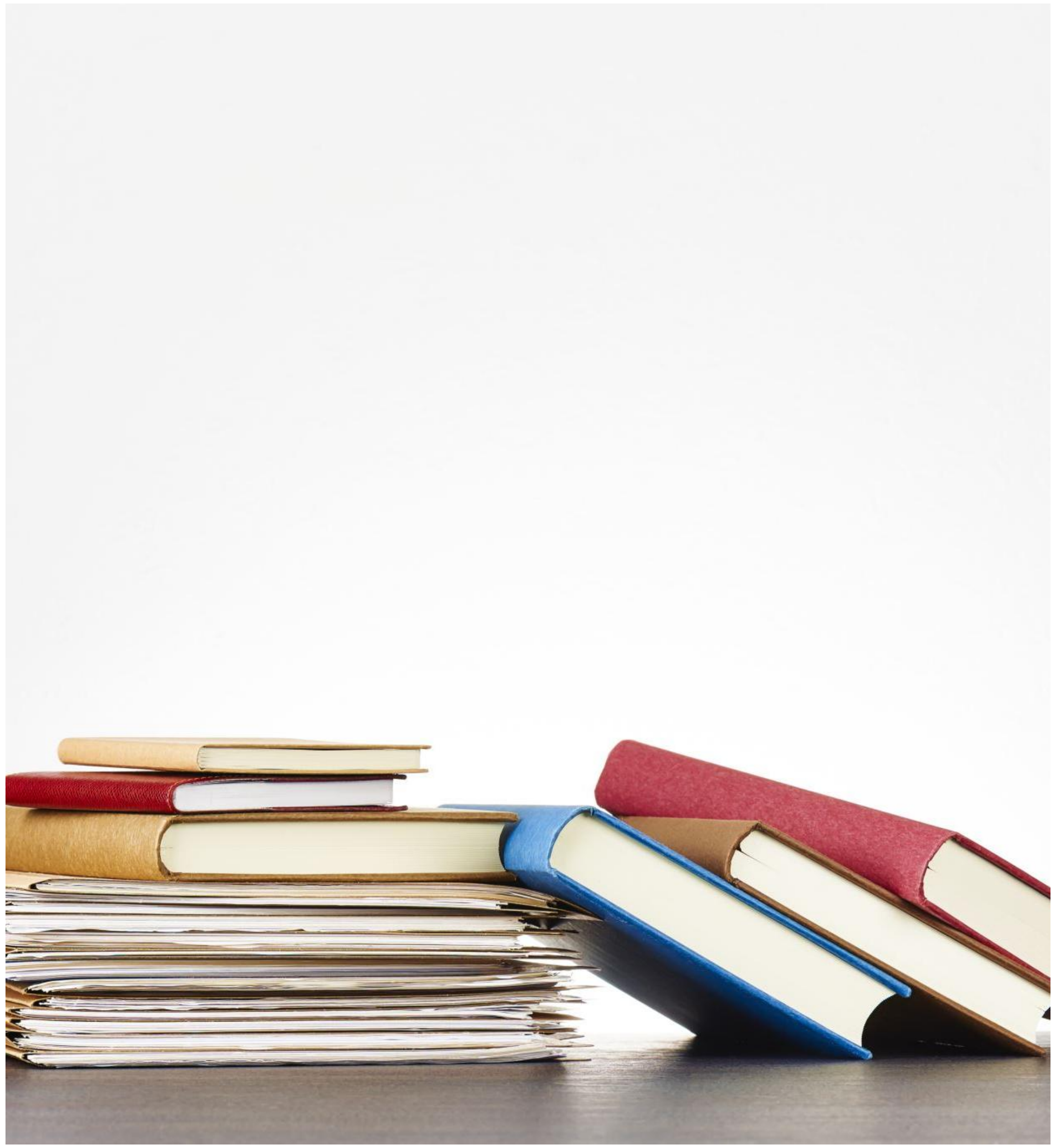
Group Members:

Abdulwadud Usman-Inenemo,

Alamin Adeleke,

Chukwunonso Daniel Ekweaga,

Emmanuel Adeyemi-Kings,
Marvellous Akinola



SYSTEM PURPOSE & KEY GOALS

- Build a reliable database to manage library resources and users.
- Track book borrowing, returns, and fines.
- Simplify user management and enhance efficiency.
- Enable quick search and generate useful reports.

SYSTEM ENTITIES & RELATIONSHIPS

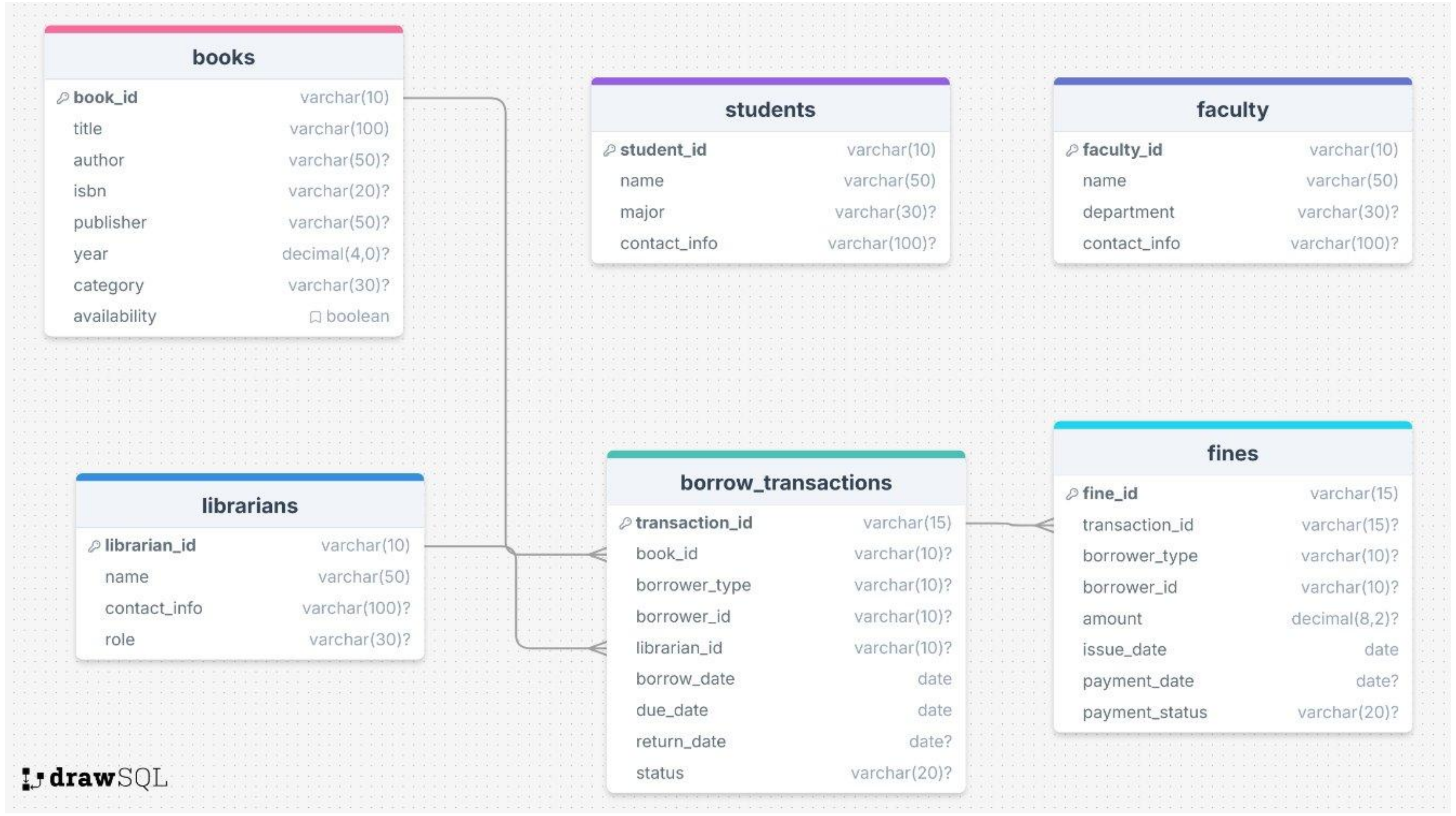
Entities:

- Books, Students, Faculty, Librarians, Borrow_Transactions, Fines

Relationships:

- A student or faculty member can borrow many books.
- A book can only be borrowed by one user at a time.
- Borrow transactions connect users, books, and librarians.
- Fines are linked to specific borrow transactions.

ENTITY RELATIONSHIP DIAGRAM



books	
book_id	varchar(10)
title	varchar(100)
author	varchar(50)?
isbn	varchar(20)?
publisher	varchar(50)?
year	decimal(4,0)?
category	varchar(30)?
availability	boolean

students	
student_id	varchar(10)
name	varchar(50)
major	varchar(30)?
contact_info	varchar(100)?

faculty	
faculty_id	varchar(10)
name	varchar(50)
department	varchar(30)?
contact_info	varchar(100)?

librarians	
librarian_id	varchar(10)
name	varchar(50)
contact_info	varchar(100)?
role	varchar(30)?

borrow_transactions	
transaction_id	varchar(15)
book_id	varchar(10)?
borrower_type	varchar(10)?
borrower_id	varchar(10)?
librarian_id	varchar(10)?
borrow_date	date
due_date	date
return_date	date?
status	varchar(20)?

fines	
fine_id	varchar(15)
transaction_id	varchar(15)?
borrower_type	varchar(10)?
borrower_id	varchar(10)?
amount	decimal(8,2)?
issue_date	date
payment_date	date?
payment_status	varchar(20)?

BOOK MANAGEMENT FEATURES

- Add new books with full bibliographic data
- Update availability in real-time upon borrow/return
- Categorize books by genre, year, or publisher
- View all books in the library

USER MANAGEMENT FEATURES

1. Register student and faculty profiles
2. Associate borrowing privileges and limits
3. Track book borrowing history
4. Distinguish between student and faculty user types

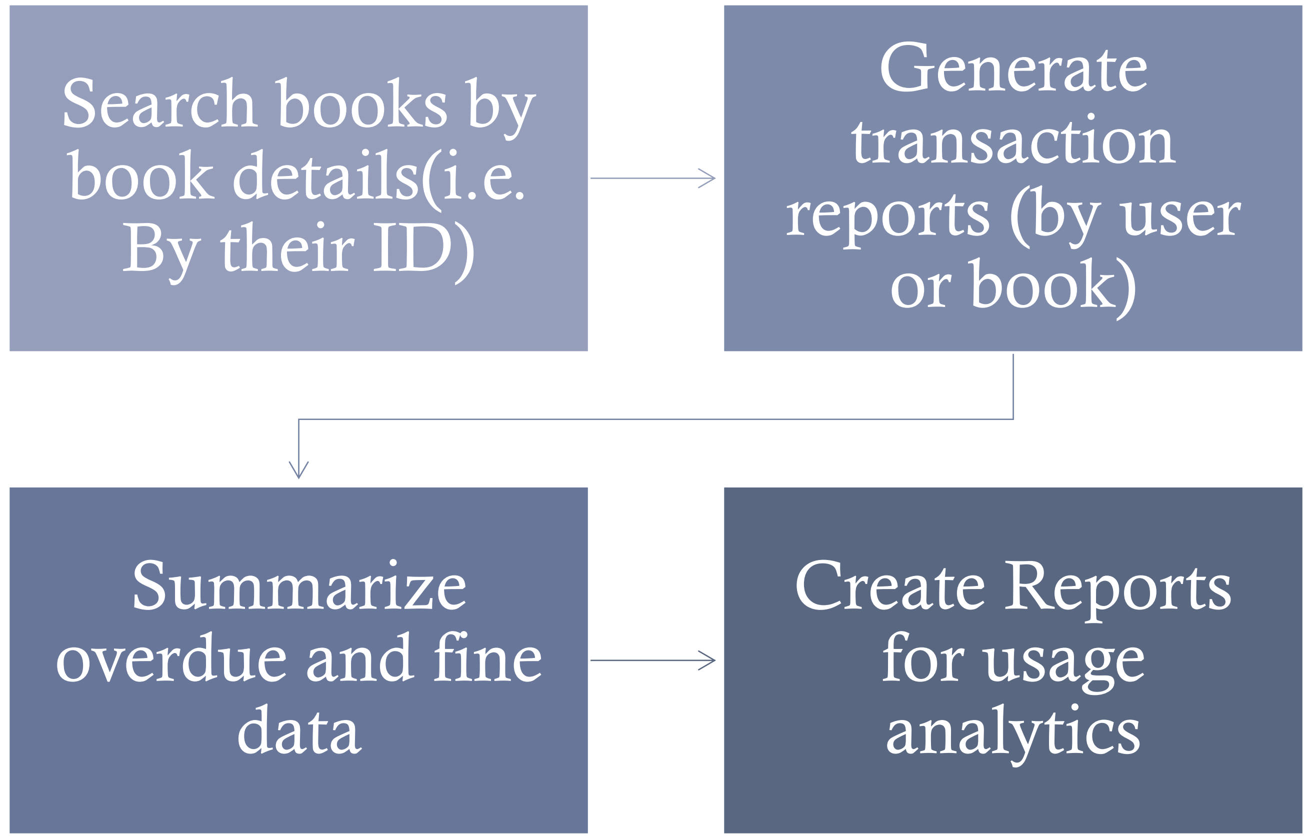
BORROW & RETURN SYSTEM

- Enable borrowing with due date assignment
- Return process updates book availability
- Record timestamps of transactions
- Ensure borrowing policy compliance

FINE MANAGEMENT SYSTEM

- Detect overdue books automatically
- Calculate fines based on delay duration
- Update payment status (paid/unpaid)
- Generate fine summaries per user

SEARCH & REPORTING FEATURES


Search books by book details(i.e. By their ID)
Generate transaction reports (by user or book)
Summarize overdue and fine data
Create Reports for usage analytics

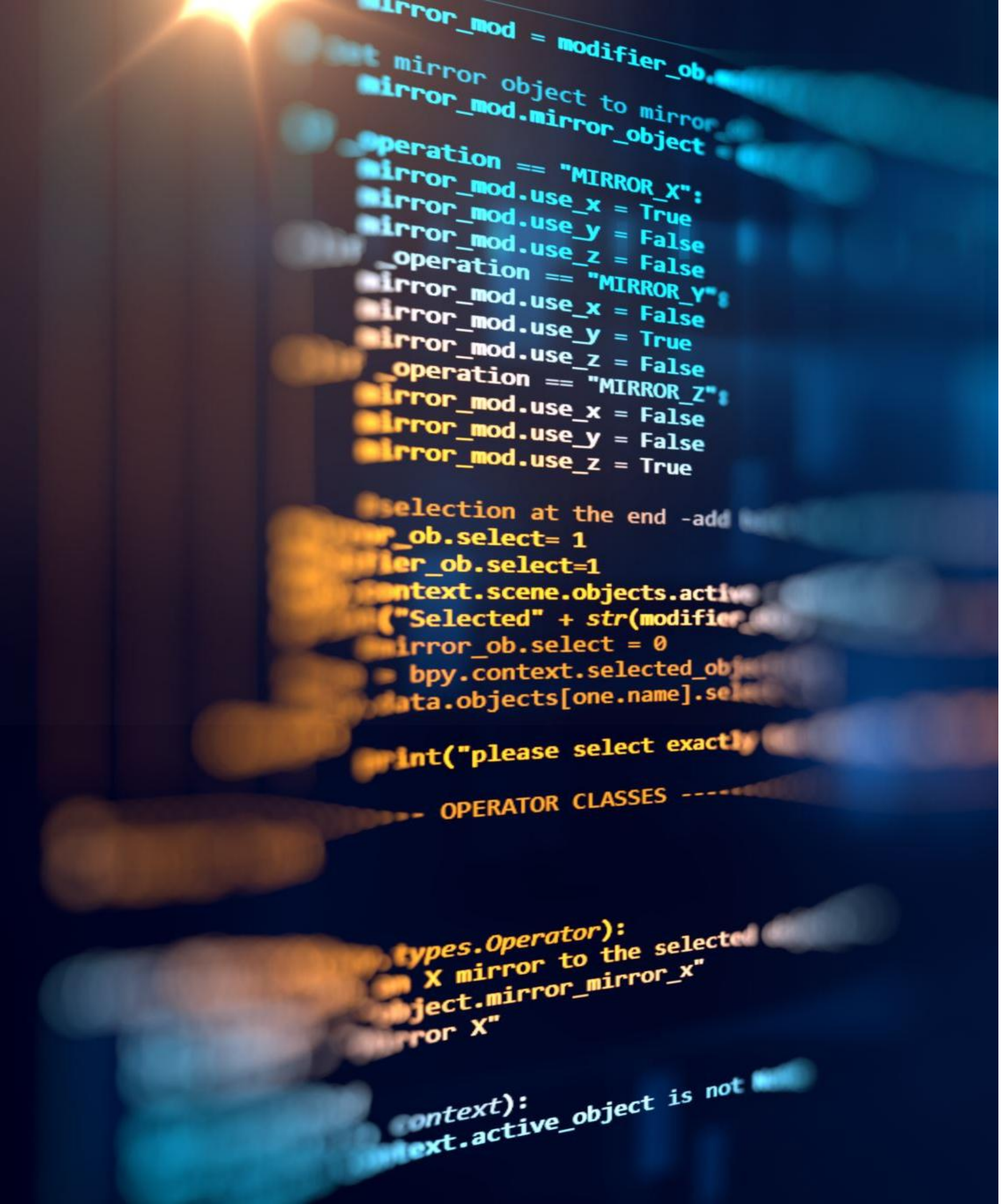
TECHNICAL IMPLEMENTATION

DBMS: MySQL / PostgreSQL

Programming Language: Java

Connection Library: JDBC

IDE & Tools: BlueJ , GitHub for version control



GITHUB REPOSITORY

Repository: https://github.com/danekweaga/Database-Project

Will include:

- Java source code
- SQL DDL scripts
- Sample data and inserts
- ER diagram
- Documentation and reports

SCHEMA Q&A - STRUCTURE

1. What tables exist in the database?

books, students, faculty, librarians, borrow_transactions, fines

2. Columns in "books" table:

book_id, title, author, isbn, publisher, year, category, availability

3. Primary key for "students" table:

student_id

SCHEMA Q&A - RELATIONSHIPS

4. Table Relationships:

borrow_transactions.book_id → books.book_id

borrow_transactions.borrower_id → students.student_id or faculty.faculty_id

borrow_transactions.librarian_id → librarians.librarian_id

fines.transaction_id → borrow_transactions.transaction_id

5. Data type for "year" in books table:

INT

BOOK FUNCTIONALITY Q&A

Retrieve

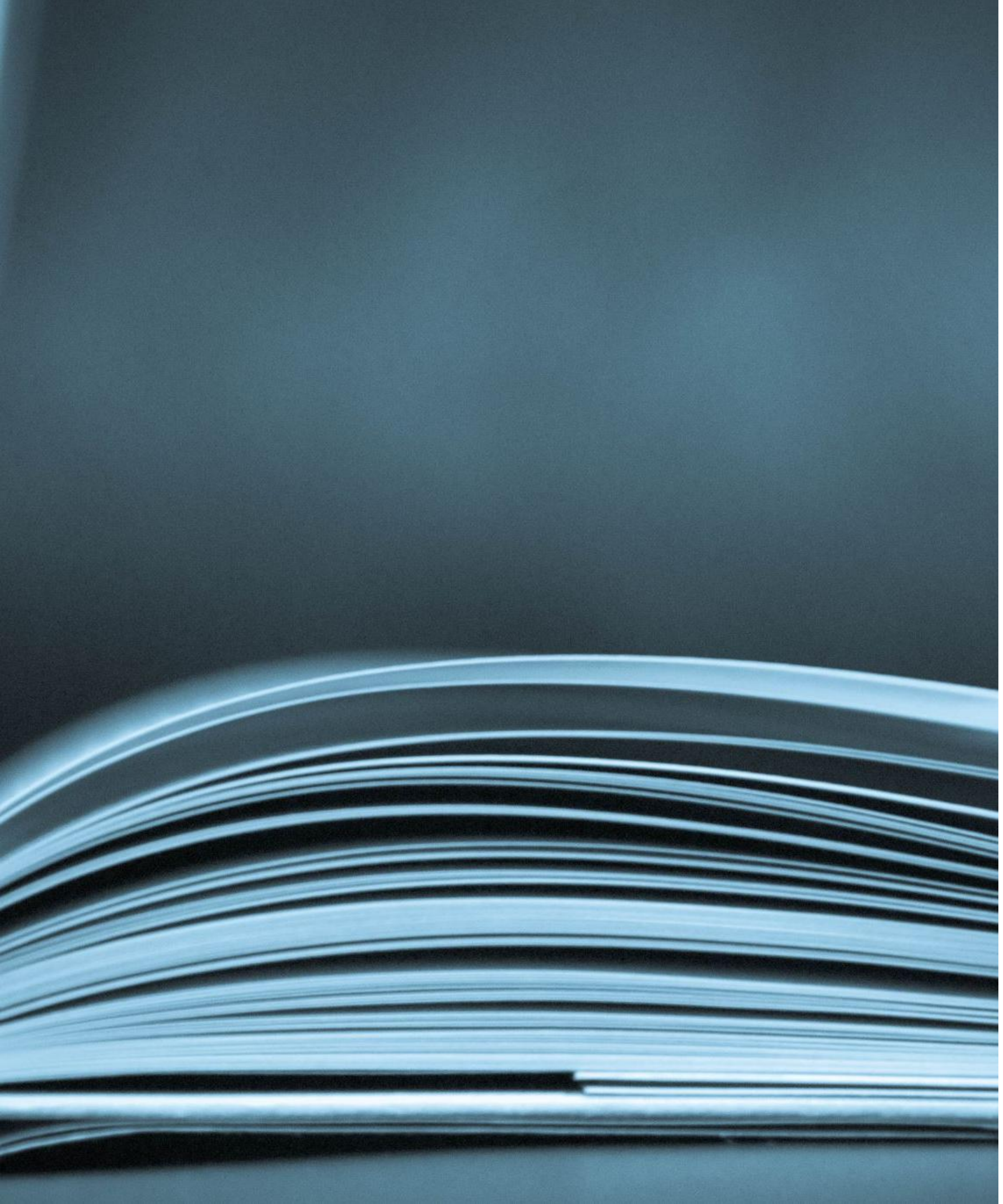
Retrieve books sorted by title:

- SELECT * FROM books ORDER BY title;

Find

Find a specific book by ID:

- SELECT * FROM books WHERE book_id = ?;



BOOK CRUD Q&A

8. Information to add a book:

book_id, title, author, isbn, publisher, year, category, availability

9. How is availability tracked?

availability column (boolean)

10. Update book availability:

UPDATE books SET availability = ? WHERE book_id = ?;

FINAL SUMMARY

- A complete, scalable system to automate library tasks
- Structured database with clear entity relationships
- Real-time transaction handling with fine tracking
- Strong technical foundation using Java and SQL
- Clear path for implementation and collaboration on GitHub