

# LIBRARY MANAGEMENT SYSTEM

IMPLEMENTATION OF DATABASE MANAGEMENT SYSTEM

AYUSHI ROY-RA2211026010439  
ANIRUDH ARCOT-RA2211026010449

# AGENDA

**01**

INTRODUCTION

**02**

ABSTRACT

**03**

PROBLEM DEFINITION

**04**

OBJECTIVE

**05**

EXECUTION PLAN

**06**

CONCLUSION

# INTRODUCTION

In the ever-evolving landscape of information management, libraries stand as bastions of knowledge dissemination and academic support. To uphold the efficiency and efficacy of library operations, the integration of advanced technological solutions is imperative. This project presents a sophisticated yet user-friendly Library Management System (LMS) meticulously crafted to meet the diverse needs of modern libraries.

At its core, this project embodies the fusion of cutting-edge database management principles with intuitive user interfaces, aiming to revolutionize the way librarians interact with their collections and patrons.



# ABSTRACT

This project presents a Library Management System (LMS) built upon robust database management principles to streamline library operations effectively. The system allows librarians to manage books, students, and transactions seamlessly through a user-friendly interface.

By integrating authentication mechanisms, it provides secure access control for librarians. This project exemplifies the transformative potential of database management systems in modernizing library management practices, offering a scalable and efficient solution for libraries.

# PROBLEM DEFINITION

The aim is to develop a library management app with a PostgreSQL backend that surpasses existing systems in efficiency, reliability, and scalability. Unlike current systems, which may suffer from data inconsistency and limited scalability, our solution ensures robust data integrity through PostgreSQL's advanced features like constraints and transactions. This guarantees accurate and reliable library records, leading to improved decision-making and user satisfaction. Additionally, the scalability of PostgreSQL enables seamless expansion to accommodate growing data volumes and user demands, eliminating the constraints faced by traditional systems.



# OBJECTIVE

- 1. Automate Library Operations:** Develop functionalities within the LMS to automate routine tasks such as book acquisitions, student registrations, statistics, and record-keeping.
- 2. Ensure Data Integrity:** Implement a robust database schema with proper constraints, relationships, and validation rules.
- 3. Enhance User Experience:** Design an intuitive and user-friendly interface for both librarians and students, facilitating easy navigation, quick access to resources, and personalized services.
- 4. Enable Secure Access Control:** PostgreSQL provides robust data integrity features such as constraints, foreign keys, and transactions, ensuring that your library data remains consistent and accurate.
- 5. Support Scalability and Adaptability:** PostgreSQL is highly scalable, allowing your library management system to grow along with your needs. It can handle large volumes of data and high numbers of concurrent users efficiently.



# EXECUTION

1

2

3

4

## **DATABASE INTEGRATION**

PostgreSQL has been used for database in this project. It contains 4 tables :- Book, Admin , Student and Record. It is a popular choice for RDBMS due to its ease of use, scalability, and performance.

## **AUTHENTICATION**

The Librarian to login using a username and a password to access the system. The Librarian can perform many functions , such as, adding a new book, new student, issuing or returning a book and view transactions

## **GUI USING SWING**

The GUI (Graphical User Interface) for this project, implemented in Java, provides a user-friendly environment for librarians to interact with the Library Management System .

## **BACKEND IMPLEMENTATION:**

Using JDBC (Java Database Connectivity) to interact with the MySQL database and perform CRUD operations.

# TABLES IN POSTGRESQL

```
library_management_system=# \dt
      List of relations
 Schema |        Name         | Type | Owner
-----+---------------------+-----+-----
 public |      book          | table | postgres
 public | issue_book        | table | postgres
 public |      record         | table | postgres
 public | return_book        | table | postgres
 public |     student          | table | postgres
(5 rows)
```

# TABLES

```
library_management_system=# \d student
                                         Table "public.student"
  Column |          Type          | Collation | Nullable | Default
  _____+_____+_____+_____+_____
    sid  | integer           |           | not null | nextval('student_sid_seq' :: regclass)
   name  | character varying(20) |           | not null |
  course | character varying(20)|           | not null |
 branch | character varying(20)|           | not null |
   email | character varying(30)|           | not null |

Indexes:
  "student_pkey" PRIMARY KEY, btree (sid)

Referenced by:
  TABLE "return_book" CONSTRAINT "fk1" FOREIGN KEY (sid) REFERENCES student(sid)
  TABLE "record" CONSTRAINT "fk1" FOREIGN KEY (sid) REFERENCES student(sid)
  TABLE "issue_book" CONSTRAINT "fk2" FOREIGN KEY (sid) REFERENCES student(sid)
```

```
library_management_system=# \d book
                                         Table "public.book"
  Column |          Type          | Collation | Nullable | Default
  _____+_____+_____+_____+_____
   isbn  | integer           |           | not null | nextval('book_isbn_seq' :: regclass)
   title | character varying(20) |           | not null |
  author | character varying(20)|           | not null |
   price | integer           |           |           | 0
 quantity | integer           |           |           | 0

Indexes:
  "book_pkey" PRIMARY KEY, btree (isbn)

Check constraints:
  "book_price_check" CHECK (price >= 0)
  "book_quantity_check" CHECK (quantity >= 0)

Referenced by:
  TABLE "issue_book" CONSTRAINT "fk1" FOREIGN KEY (isbn) REFERENCES book(isbn)
  TABLE "return_book" CONSTRAINT "fk2" FOREIGN KEY (isbn) REFERENCES book(isbn)
  TABLE "record" CONSTRAINT "fk2" FOREIGN KEY (isbn) REFERENCES book(isbn)
```

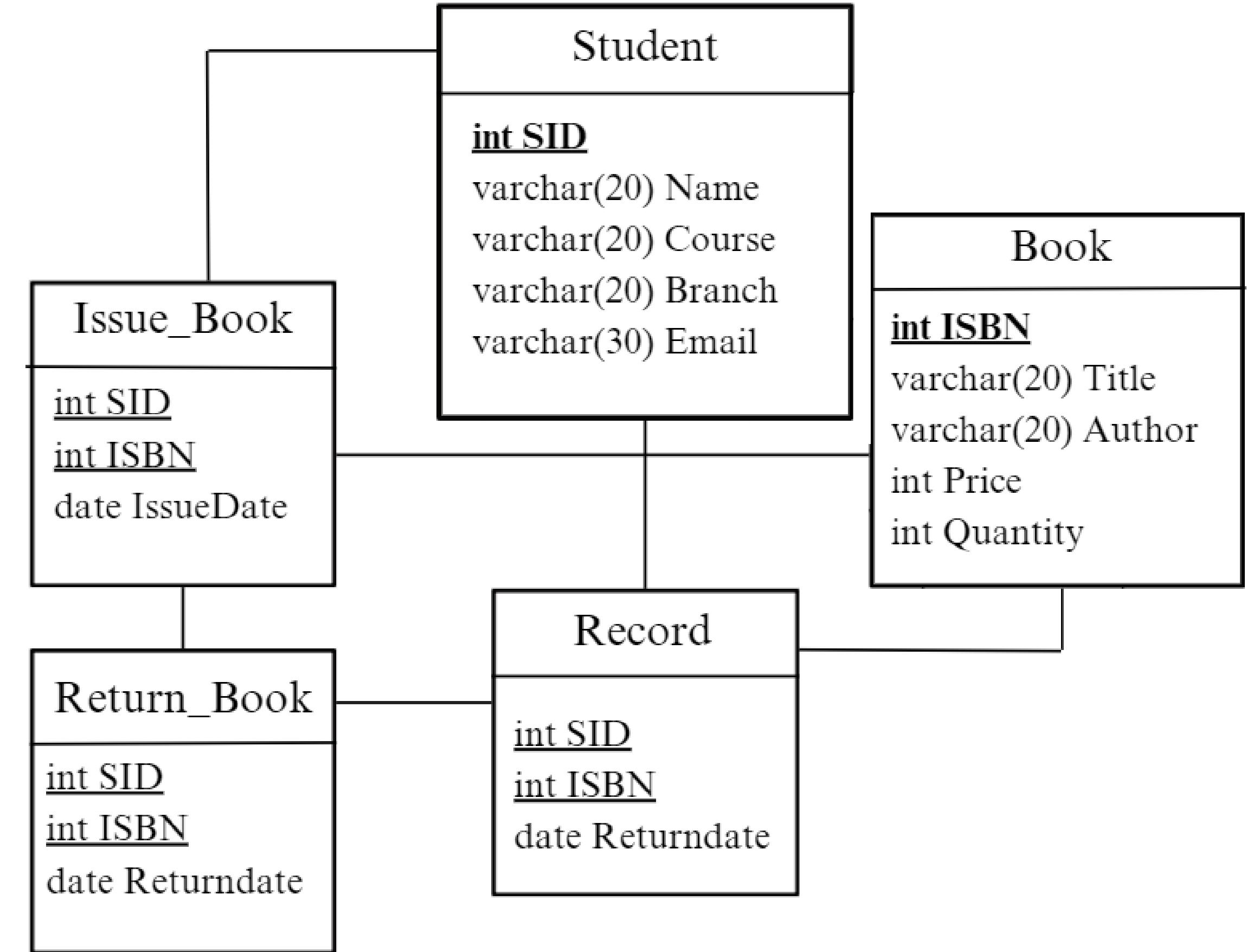
# TABLES

```
library_management_system=# \d return_book
Table "public.return_book"
 Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+
 sid | integer | | not null | nextval('return_book_sid_seq'::regclass)
 isbn | integer | | not null | nextval('return_book_isbn_seq'::regclass)
 return_date | date | | not null |
Foreign-key constraints:
 "fk1" FOREIGN KEY (sid) REFERENCES student(sid)
 "fk2" FOREIGN KEY (isbn) REFERENCES book(isbn)
```

```
library_management_system=# \d issue_book
Table "public.issue_book"
 Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+
 isbn | integer | | not null | nextval('issue_book_isbn_seq'::regclass)
 sid | integer | | not null | nextval('issue_book_sid_seq'::regclass)
 issue_date | date | | not null |
Foreign-key constraints:
 "fk1" FOREIGN KEY (isbn) REFERENCES book(isbn)
 "fk2" FOREIGN KEY (sid) REFERENCES student(sid)
```

```
library_management_system=# \d record
Table "public.record"
 Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+
 sid | integer | | not null | nextval('record_sid_seq'::regclass)
 isbn | integer | | not null | nextval('record_isbn_seq'::regclass)
 issue_date | date | | |
 return_date | date | | |
Foreign-key constraints:
 "fk1" FOREIGN KEY (sid) REFERENCES student(sid)
 "fk2" FOREIGN KEY (isbn) REFERENCES book(isbn)
```

# RELATIONAL SCHEMA



# CONCLUSION

In conclusion, the Library Management System project, implemented using Java for the GUI and MySQL for the backend database, represents a step towards modernizing library operations and enhancing user experiences.

By leveraging Java's GUI libraries and SQL's powerful database management capabilities, the project delivers a comprehensive solution for librarians to efficiently manage library resources.

The project's scope encompasses essential functionalities such as adding new books and students, issuing and returning books, and viewing transaction history. The user-friendly GUI provides librarians with a seamless interface to perform these tasks effectively, while also ensuring data integrity and scalability through MySQL's relational database management system.



# **THANK YOU!**

**AYUSHI ROY- 439  
ANIRUDH ARCOT-449**

