## **Introduction**

### **Analysis of the Problem**

The project's main issues focus on the need for a centralized, effective system for managing library operations. Key problem points include:

* **Manual Processes**: Existing systems require significant manual intervention, leading to inefficiencies and inaccuracies.
* **Poor Data Tracking**: Inadequate tracking of which students have borrowed which books and the absence of organized student preferences.

By establishing a relational database system with clearly defined tables and relationships, this project aims to address these problems.

### **Discussion of Data**

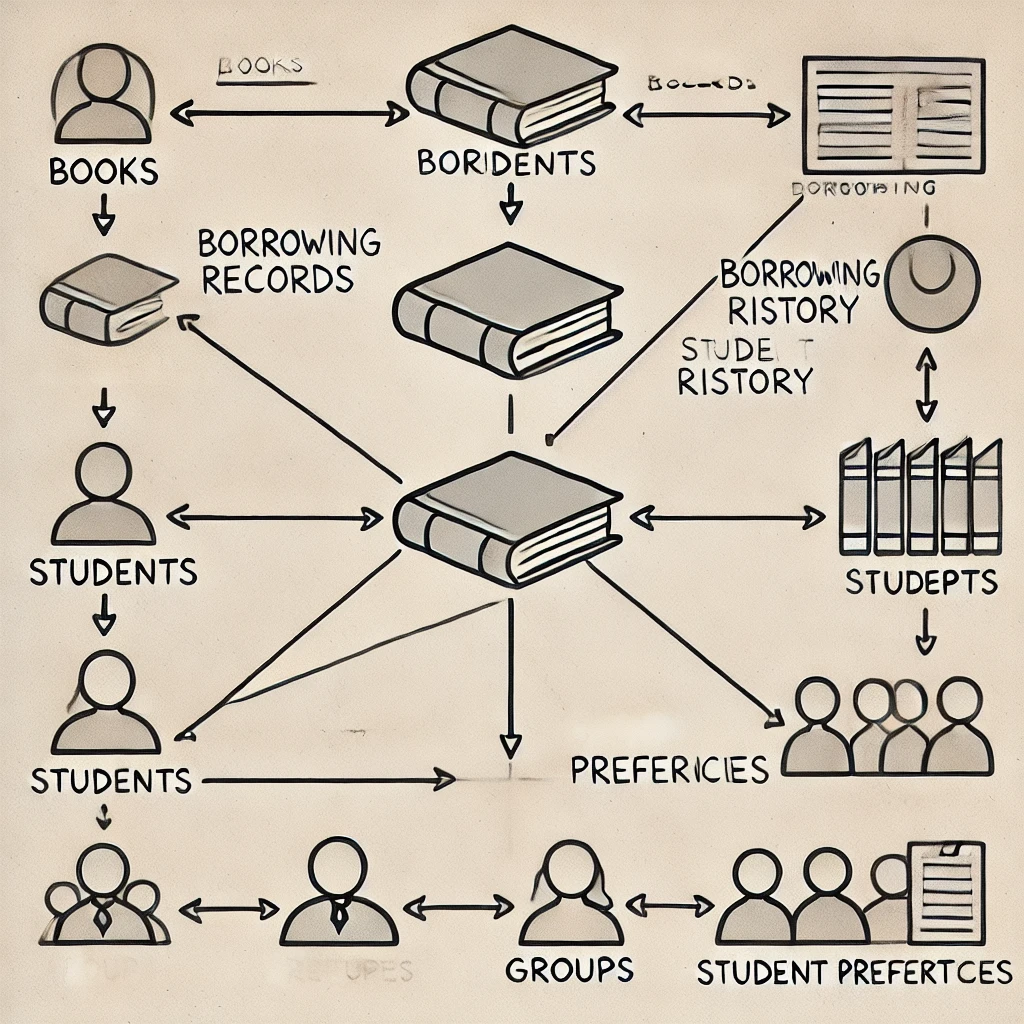
The database will primarily manage four categories of data:

1. **Books**: Details such as title, author, publication year, genre, and availability status.
2. **Students**: Information like student ID, name, and associated borrowing records.
3. **Groups**: Organizational units that group students based on shared preferences or academic needs.
4. **Preferences**: Records of students’ preferred genres or book types, which inform group assignments and library resource planning.

### **Database Design**

The database design is based on an Entity-Relationship diagram, which illustrates the key entities and their relationships:

* **Books**: The central entity, linked to borrowing records and student preferences.
* **Students**: Associated with borrowing history and preferences.
* **Groups**: Connected to students and preferences to reflect collective reading interests.



### **Explanation of Tables and Relationships**

1. **Books Table**: Contains fields like BookID, Title, Author, Genre, PublicationYear, and AvailabilityStatus.
2. **Students Table**: Includes StudentID, Name, Email, and GroupID.
3. **Groups Table**: Contains GroupID and GroupName, serving as a link to students.
4. **Preferences Table**: Maps student preferences to genres, using StudentID and Genre.
5. **Borrowing Table**: Tracks BorrowID, StudentID, BookID, BorrowDate, and ReturnDate.

These tables' connections ensure data integrity and enable efficient querying. The Students table and the Borrowing table, for example, have a one-to-many link, however the Books table and Borrowing have a similar relationship.

SELECT \* FROM library4.students;

create table Books(

book\_id int,

isbn int,

book\_title TEXT,

publisher\_id int,

book\_format TEXT,

pages int,

published DATE,

year TEXT,

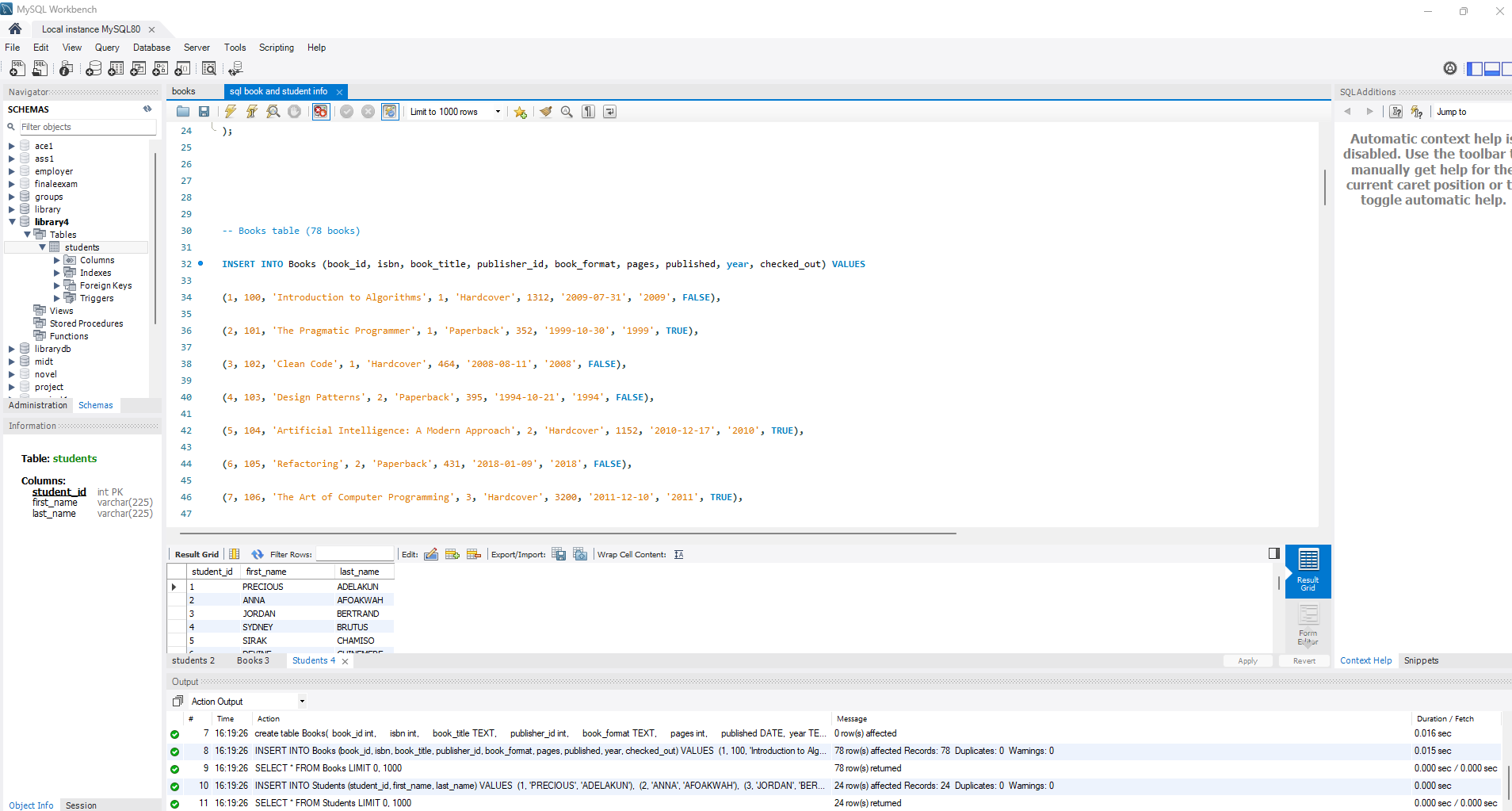
checked\_out bool

);

**Implementation**

### **Tools Used**

1. **Node.js**:A runtime environment for creating server-side apps that are quick, scalable, and event-driven. It was chosen because of its dynamic features, which provide effective management of database transactions and queries.
2. **Mysql**: A lightweight, self-contained SQL database engine, ideal for small-to-medium-sized projects. It was chosen for its simplicity and ability to handle structured data effectively.



1. **Visual Studio Code**: A powerful integrated development environment (IDE) used for coding, debugging, and testing the application. It supports extensions for this project which made things less complicated for the development process.
2. **GitHub**: For version control and collaboration, ensuring that the project remains well-organized and easily accessible for future enhancements.

