**Introduction:**

This project aims to design a library database that will enable librarians to easily manage and keep track of the books and other media available in the library. The database will cover all aspects of the library, including the collection size, the types and categories of data to be stored, and the media types and genres to be covered. The database will have a user-friendly interface allowing librarians to search for and locate materials, add new items to the library, update existing entries, and perform other necessary tasks. It will also include features such as tracking borrowed materials, who have borrowed them, and when they are due to be returned. The Library database’s main purpose is to keep track of books stored in a library. The problem arises when more people and books are coming and going at the same time. To keep up with returns, takeaways, and due dates the library database was made. It solves the problem by listing what is in the library's inventory, showing descriptions of books, and also who has signed out books. By storing and organizing this information it allows the library to efficiently and easily check to see what they have in stock and/or when they expect something to be returned.

**Database Description:**

**Physical Database:**

The above code is a SQL script that creates a database called "library\_project" and defines four tables within it: "books", "patrons", "book\_borrowing", and "book\_reservations".

The "books" table has six columns, including a primary key column called "isbn\_ID" that is of text type. Other columns in the table include "title", "author", "publication\_date", "language", and "number\_of\_copies".

The "patrons" table has eight columns, including a primary key column called "patron\_ID" that is of bigserial type. Other columns in the table include "first\_name", "last\_name", "address", "phone\_number", "email\_address", "library\_card\_number", and "account\_balance".

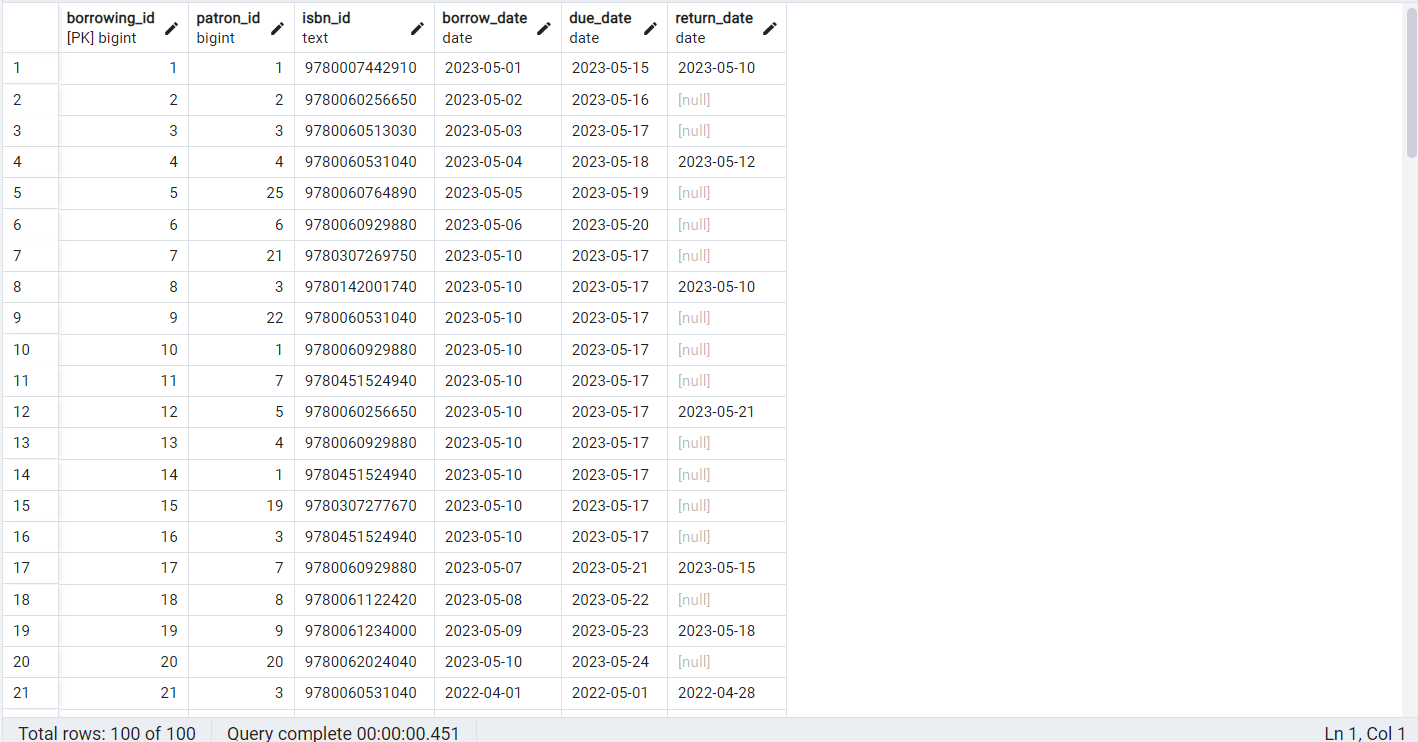
The "book\_borrowing" table has six columns, including a primary key column called "borrowing\_ID" that is of bigserial type. This table also includes two foreign key columns, "patron\_ID" and "isbn\_ID", which reference the primary keys of the "patrons" and "books" tables, respectively. Other columns in the table include "borrow\_date", "due\_date", and "return\_date".

Finally, the "book\_reservations" table has five columns, including a primary key column called "reservation\_ID" that is of bigserial type. This table also includes two foreign key columns, "patron\_ID" and "isbn\_ID", which reference the primary keys of the "patrons" and "books" tables, respectively. Other columns in the table include "reservation\_date" and "expiration\_date".

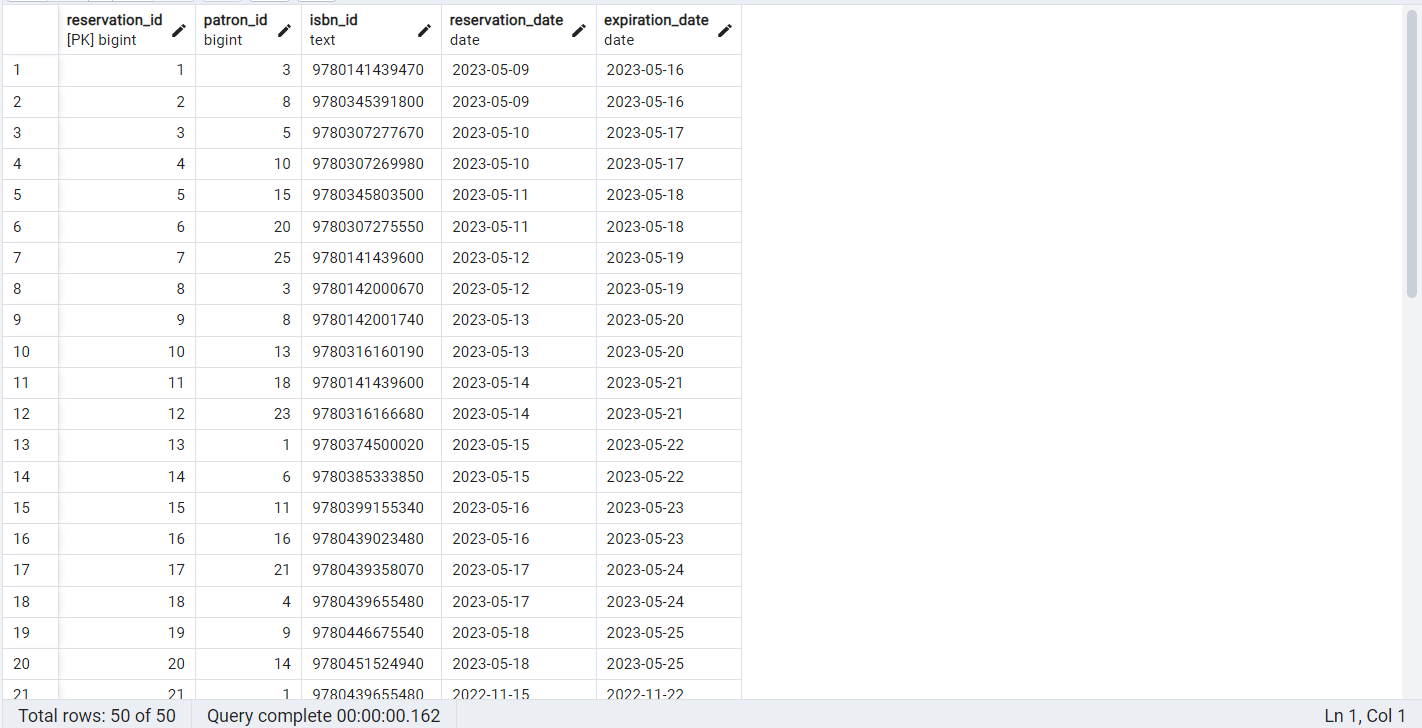
Overall, this database is built for a library system, with tables for books, patrons, book borrowing, and book reservations. The primary key and foreign key constraints help to maintain data integrity and prevent inconsistent data from being entered into the database.

**Sample Data:**

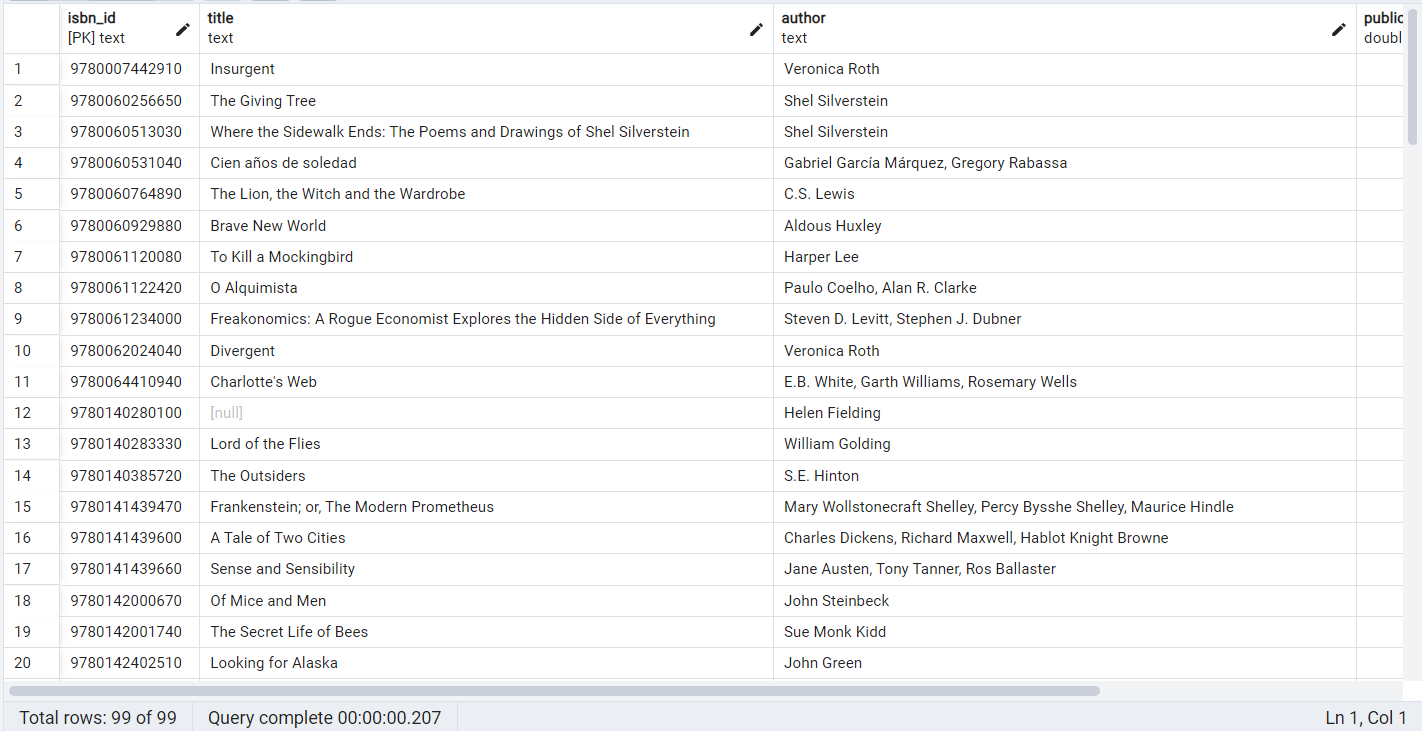
Here are some screenshots of SELECT queries running on our tables populate with our sample data:

****

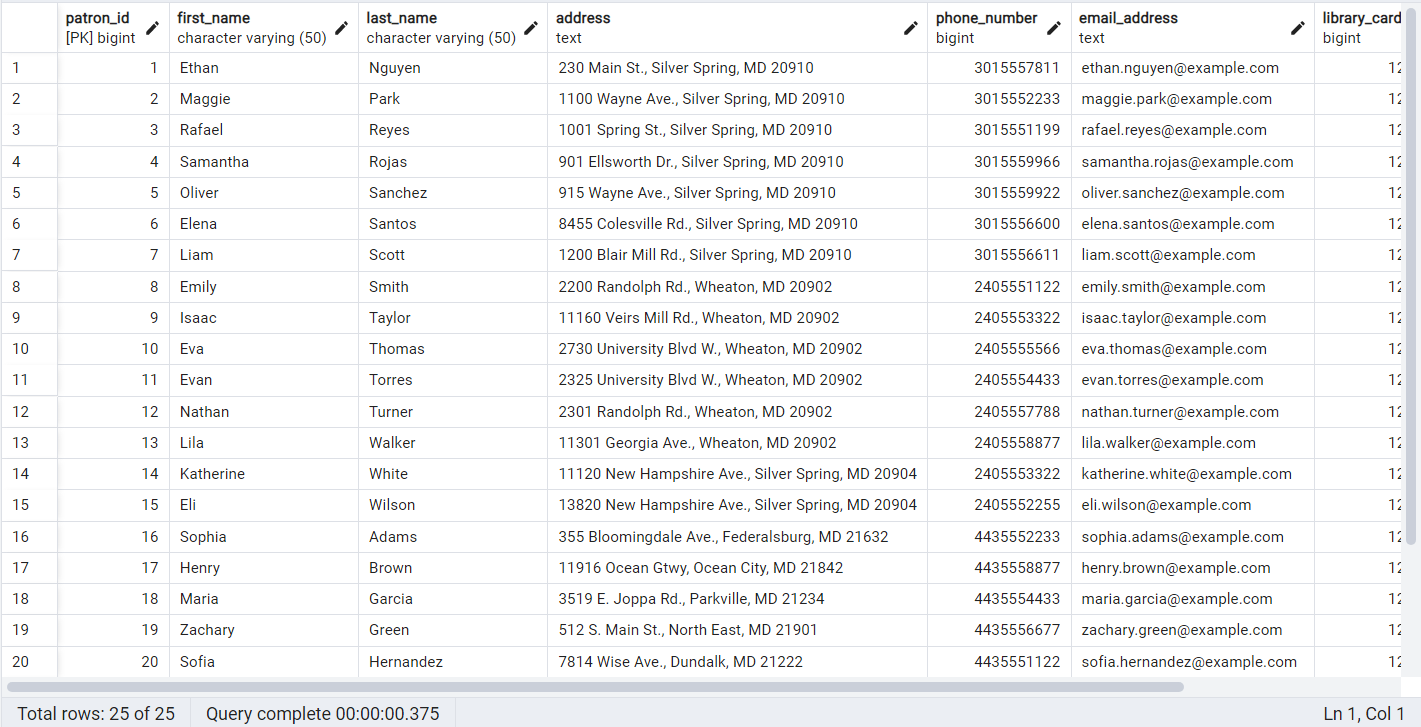
"book\_borrowing" table

****

"book\_reservations" table

****

"books" table

****

"patrons" table

**Queries:**

All the queries here are numbered by their order in the ‘team\_03\_library\_queries.txt’ file which has been submitted along with this report.

* 11 Queries Total
* No Two Queries Are Variations Of Each Other
* We only use the wildcard where it is necessary.

| **Query Number** | **Involve Multiple Tables (A)** | **Involve Filtering**  **(B)** | **Involve Aggregation**  **(C)** | **Involves Subqueries**  **(D)** |
| --- | --- | --- | --- | --- |
| **1** | **✅** | **✅** | **❌** | **❌** |
| **2** | **✅** | **✅** | **✅** | **❌** |
| **3** | **✅** | **✅** | **✅** | **❌** |
| **4** | **✅** | **✅** | **✅** | **✅** |
| **5** | **✅** | **✅** | **❌** | **❌** |
| **6** | **❌** | **❌** | **❌** | **❌** |
| **7** | **❌** | **✅** | **❌** | **❌** |
| **8** | **❌** | **✅** | **✅** | **❌** |
| **9** | **❌** | **✅** | **❌** | **❌** |
| **10** | **✅** | **❌** | **✅** | **❌** |
| **11** | **❌** | **❌** | **✅** | **❌** |

**Changes From Original Design:**

Since we submitted our proposal, our team refined our database design and implemented the schema. We received feedback from our mentor and instructors and adjusted our initial plans based on their suggestions. We had several meetings to discuss the feedback and how to incorporate it into our project. We also consulted with the instructors and mentors when we encountered difficulties.

We also revisited the initial proposal and rethought and reworked the design of our database. Other than minor changes, such as changing the names of columns, the only significant difference was our decision to include only ISBN 13. We also decided to remove the late fee column for books as we decided that calculating the late fee was more of an administrative thing than a database-related question. Also, since the date when the book was borrowed was already included in the database, it was trivial to calculate the late fee after-the-fact. We also concluded that the author's table which listed date of birth, date of death, nationality, and so on did not provide significant use and would not be useful since the only important factors a library would take into account were the author's name, book language, and the genre of the book. The authors' table simply felt like an extra addition that did not serve any specific purpose since a library's main focus lay in the books rather than the authors themselves.

We planned to populate the database with more sample data and ensured all tables were correctly linked using primary and foreign keys. We also wrote queries to retrieve information from the database, such as finding books by author or publisher. We created a database backup process to ensure the safety of the data, and we compiled a final report documenting our work on the project. We distributed the remaining tasks among team members based on their skills and availability and successfully completed the project.

**Lessons Learned:**

The lessons learned during this project mostly falls under coding aspects. One example is using the JOIN query. When working on the required queries for the project there were many times where JOIN had to be used in order to show how information related across the tables in the database. The tricky bit was figuring out ways to match multiple tables together to show specific information that a librarian might need. Another lesson learned is the importance of database upkeep. As my teammates have stated, populating the database by hand is both exhausting and time-consuming and we only have a fraction of what a real library would have. As the number of books, patrons, and readers increases it is mandatory to have a database so library staff can find the information they need immediately.

**Potential Future Plans:**

There are several potential future plans that could be pursued to further develop the library database. One possible area for improvement is the user interface. While the current design is functional and user-friendly, there may be ways to streamline the interface and make it even more intuitive for librarians to use. Another area for improvement could be the addition of more advanced search capabilities, such as the ability to search by keyword or subject matter. It may also be beneficial to integrate the database with other library management systems, such as cataloging or interlibrary loan systems, to create a more comprehensive and cohesive library management solution. Additionally, the database could be expanded to include additional media types, such as audio or video materials, or to support multi-language collections. Finally, ongoing maintenance and updates will be important to ensure the database remains relevant and useful over time, including regular backups, security updates, and bug fixes as needed.

**Works Cited:**

<https://github.com/zygmuntz/goodbooks-10k/blob/master/samples/books.csv>