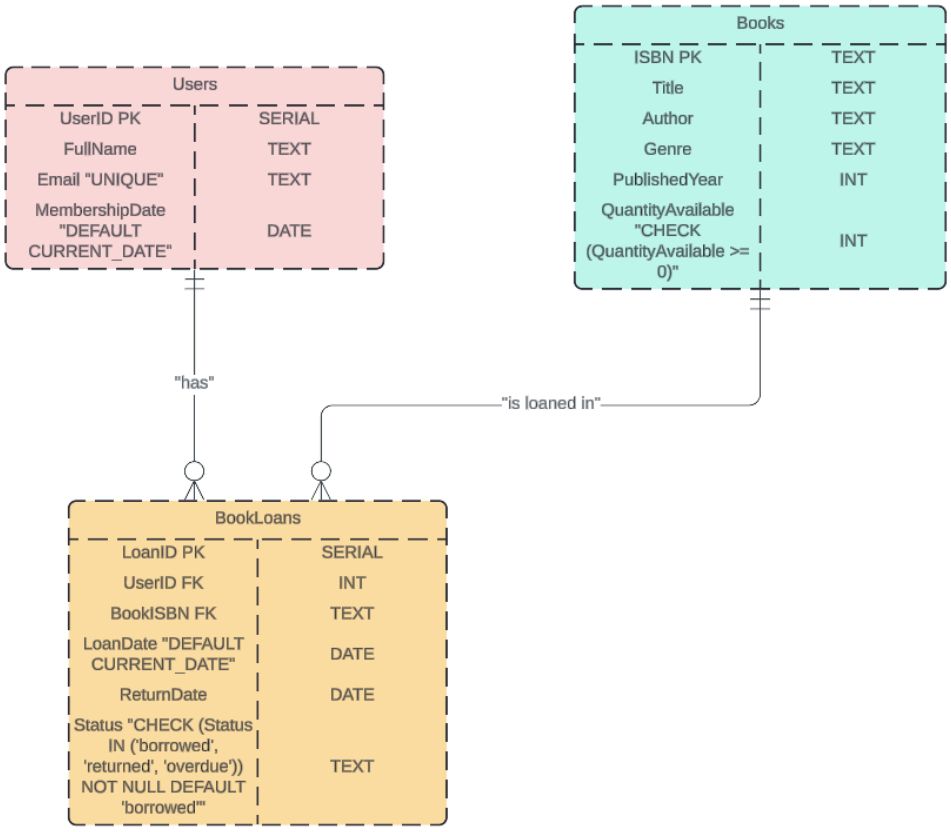
SE 2141 LABORATORY 4

REGINE THERESE N. BARTE **DECEMBER 11, 2024**

Part 1: Conceptual Design - 25pts

1. Draw an Entity-Relationship (ER) Diagram for the system based on the given requirements. Ensure you specify:

○ Entities
○ Attributes
○ Primary Keys
○ Relationships with cardinalities (e.g., one-to-many, many-to-many)



Part 2: Logical Design - 25pts

2. Translate the ER diagram into relational tables. Define:

○ Table schemas (list all attributes, data types, and constraints such as primary keys, foreign keys, and NOT NULL).

```sql
CREATE TABLE Books (
    ISBN TEXT PRIMARY KEY,
    Title TEXT NOT NULL,
    Author TEXT NOT NULL,
    Genre TEXT,
    PublishedYear INT,
    QuantityAvailable INT NOT NULL CHECK (QuantityAvailable >= 0)
);

CREATE TABLE Users (
    UserID SERIAL PRIMARY KEY,
    FullName TEXT NOT NULL,
    Email TEXT UNIQUE NOT NULL,
    MembershipDate DATE DEFAULT CURRENT_DATE
);

CREATE TABLE BookLoans (
    LoanID SERIAL PRIMARY KEY,
    UserID INT NOT NULL REFERENCES Users(UserID),
    BookISBN TEXT NOT NULL REFERENCES Books(ISBN),
    LoanDate DATE DEFAULT CURRENT_DATE,
    ReturnDate DATE,
    Status TEXT CHECK (Status IN ('borrowed', 'returned', 'overdue')) NOT NULL DEFAULT 'borrowed'
);

```

Part 3: SQL Queries

3. Write SQL queries for the following scenarios (15pts each):

○ a. Insert a new book into the library with a quantity of 5.

```
4
5   INSERT INTO Books (ISBN, Title, Author, Genre, PublishedYear,
    QuantityAvailable)
6   VALUES ('1112346782104', 'Geronimo Stilton Lost Treasure of the Emerald
    Eye', 'Geronimo Stilton',  'Fiction', 1887, 5);
7
```

○ b. Add a new user to the system.

```
4
5   INSERT INTO Users (FullName, Email, MembershipDate)
6   VALUES ('Regine Therese Barte', 'Chokiniko@gmail.com', CURRENT_DATE);
7
```

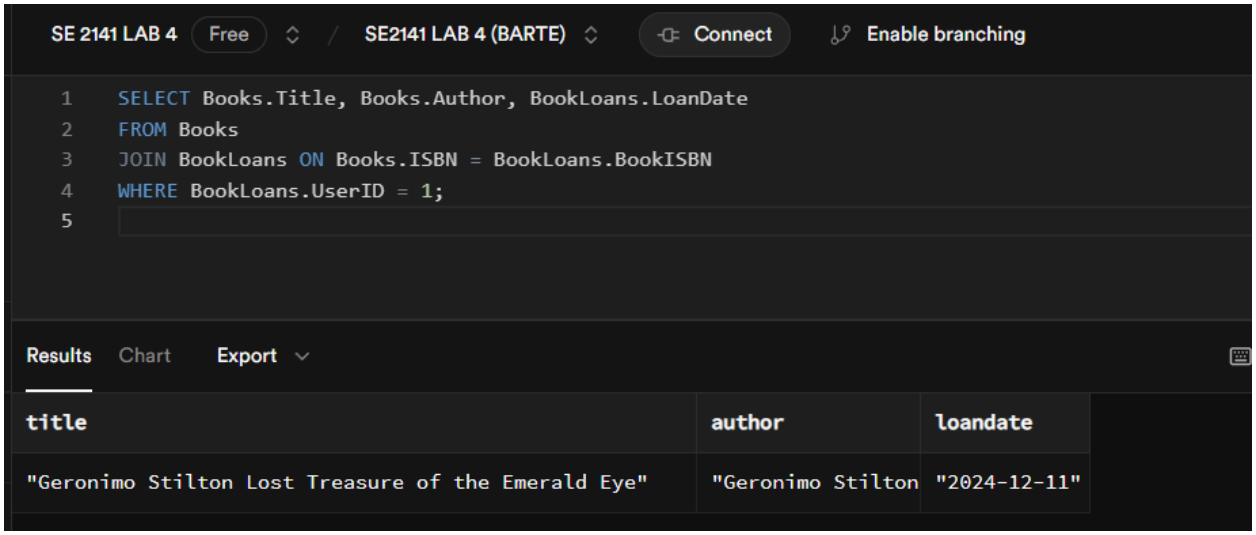
○ c. Record a book loan for a user.

```
1
2
3   INSERT INTO BookLoans (UserID, BookISBN, LoanDate, ReturnDate, Status)
4   VALUES (1, '1112346782104', CURRENT_DATE, '2024-12-10', 'borrowed');
5
```

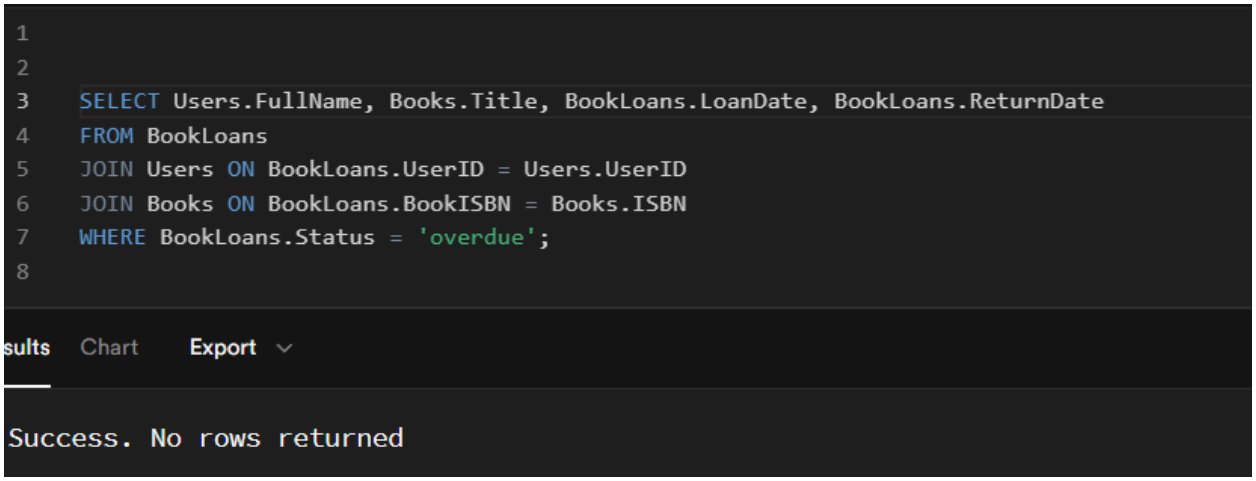
sults Chart Export ⋮ Source Primary Database role postgres

Success. No rows returned

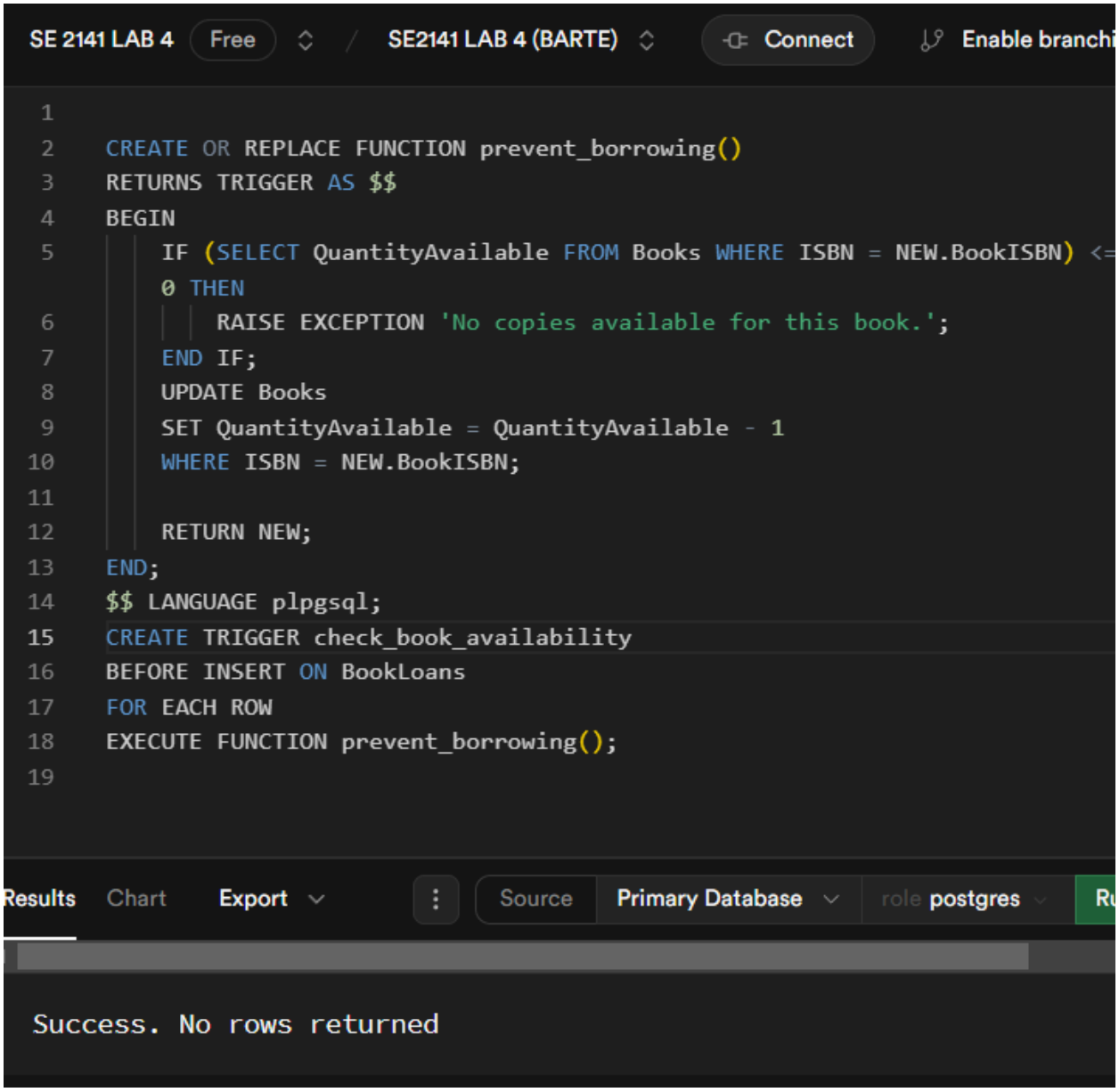
○ d. Find all books borrowed by a specific user.



○ e. List all overdue loans.



Part 4: Data Integrity and Optimization



- First, it makes a new trigger function prevent_borrowing() that will be executed before inserting a new loan into the book loans table. It will send out a trigger then if there are any copies of the books available to borrow by querying the books table for the quantity available.
- if the available quantity is less than or equal to zero then it raises an error that says “No copies available for this book”
- it would also check the availability and would loop to check if there are still books available to prevent loans if the book is not available anymore.

```sql

CREATE INDEX idx_bookloans_status ON BookLoans(Status);

SELECT Users.FullName, Books.Title, BookLoans.LoanDate, BookLoans.
ReturnDate
FROM BookLoans
JOIN Users ON BookLoans.UserID = Users.UserID
JOIN Books ON BookLoans.BookISBN = Books.ISBN
WHERE BookLoans.Status = 'overdue';

```

- This creates an index on status column where indexes improve query performance for filtering or searching by specific columns.
- Then selecting from the required column joining the book borrowed and the user, using the ISBN to get the title of the book that the user wants to borrow.
- Filtering then is where book loans are being separated to be classified specifically the overdue ones.

Part 5: Reflection (25 pts)

5. What challenges might arise when scaling this database to handle millions of users and
books? Suggest one solution for each challenge.

- I think the difficulty when handling millions of users would be the capacity of the database to carry out each information and give it more time to process each time it needs to be accessed. The consistency of data may also occur as loading of information through time may change and be updated so as the operations require it to process fast and accurately..