

Paul Ian D. Capanas BSSE-3

Instructions:

1. Read the case study carefully and answer the questions based on the requirements described.
2. Use **ER diagrams**, **SQL schema definitions**, and written explanations where applicable.
3. Complete the exam by **12/11/2024 19:00**.

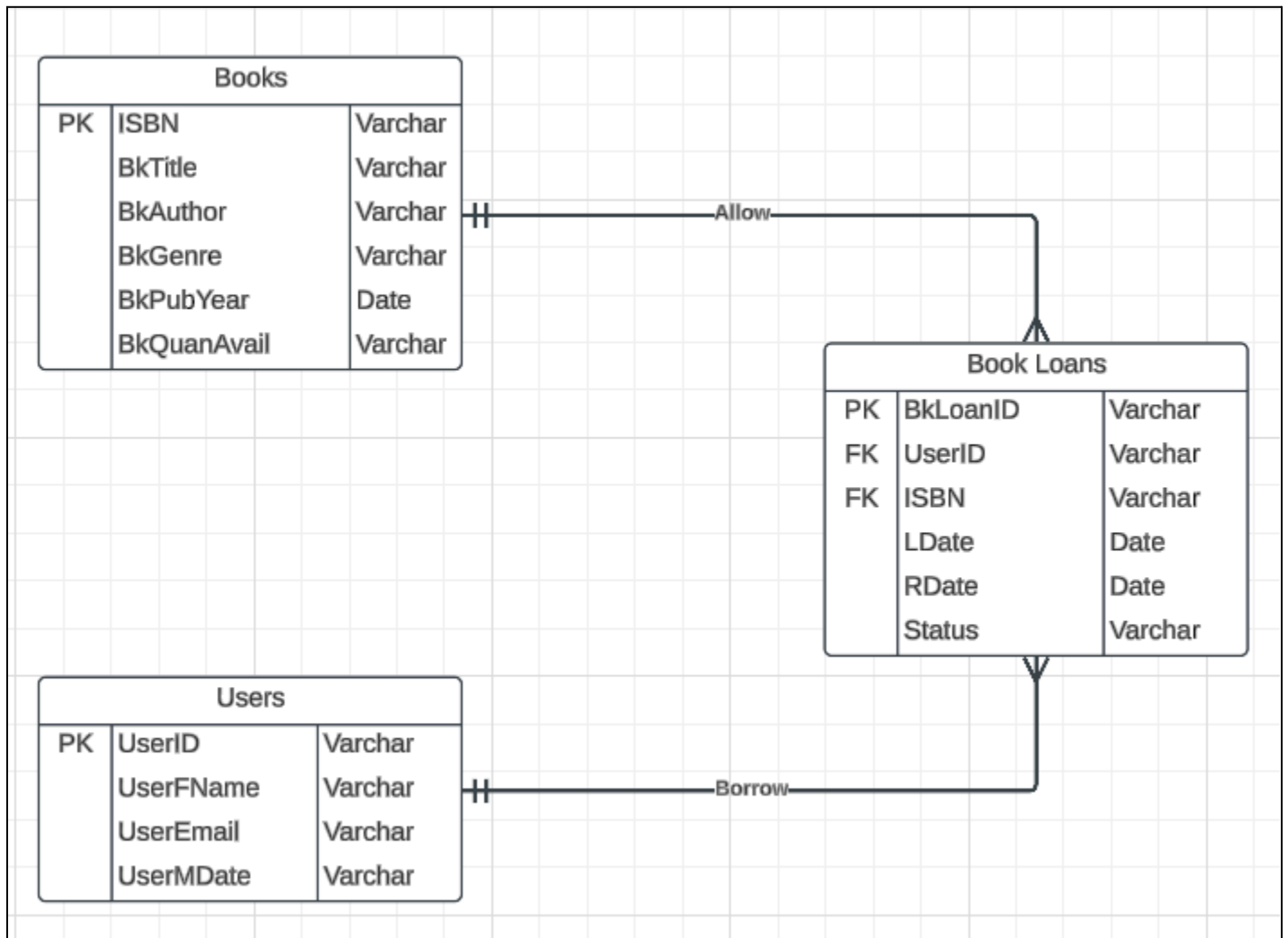
Case Study:

You have been tasked to design a database for an **Online Library Management System**. The system should keep track of books, users, and book loans. Below are the requirements:

1. **Books:** The library has a collection of books. Each book has the following details:
 - Title
 - Author
 - ISBN (unique identifier)
 - Genre (e.g., Fiction, Non-Fiction) ○
 - Published Year
 - Quantity Available
2. **Users:** Users of the library can borrow books. Each user has:
 - A unique ID
 - Full Name
 - Email Address
 - Membership Date
3. **Book Loans:** Users can borrow books. Each loan should record:
 - User ID
 - Book ISBN
 - Loan Date
 - Return Date
 - Status (e.g., "borrowed", "returned", "overdue")
4. **Rules:**
 - A user can borrow multiple books, but the loan status must be updated when books are returned.
 - The library should not allow loans for unavailable books (i.e., if all copies of a book are borrowed).

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)



I used Varchar on mostly uniformly on the data types, by means of usage so that it will perform faster rather than the other data types. The other benefit are its flexibility in terms of indexing. According to Harsh Priyadarshil VARCHAR typically performs better than TEXT because it needs less storage and provides quicker access to the data. While handling higher amounts of data, this speed benefit may be lost.

Feature	TEXT	VARCHAR
Data Length	0 - 4,294,967,295	0 - 65535
Storage	Data is stored in a static form	Data storage is variable
Performance	Slower	Faster
Indexing	Not Possible	Possible
Usage	To store larger strings of data	To store a shorter string of data

Comparison of Varchar and Text

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).

```
create table Books (  
ISBN      serial unique primary key,  
BkTitle   varchar(255) not null,  
BkAuthor  varchar(255) not null,  
BkGenre   varchar(255),  
BkPubYear date,  
BkQuanAvail varchar(255) not null default '0'  
)
```

Creation of Book table, I created the book table ISBN, as Primary Key, the required entries are the Book Title for every book stored in the library has an author, the Book Author as each book has its own writer, and the Book Quantity as it defaulted is missing is 0. I've allowed not required on Book Genre, as for me it is subjective on how will you depict a book. Book Published Year, as some books cannot be recalled on what year are they published ex. A Spellbook found in the 18th century doesn't have a published date rather it was only estimated.

```
create table Users (  
UserID serial unique primary key,  
UserFName varchar(255) not null,  
UserEmail varchar(255) not null unique,  
UserMDate date  
)
```

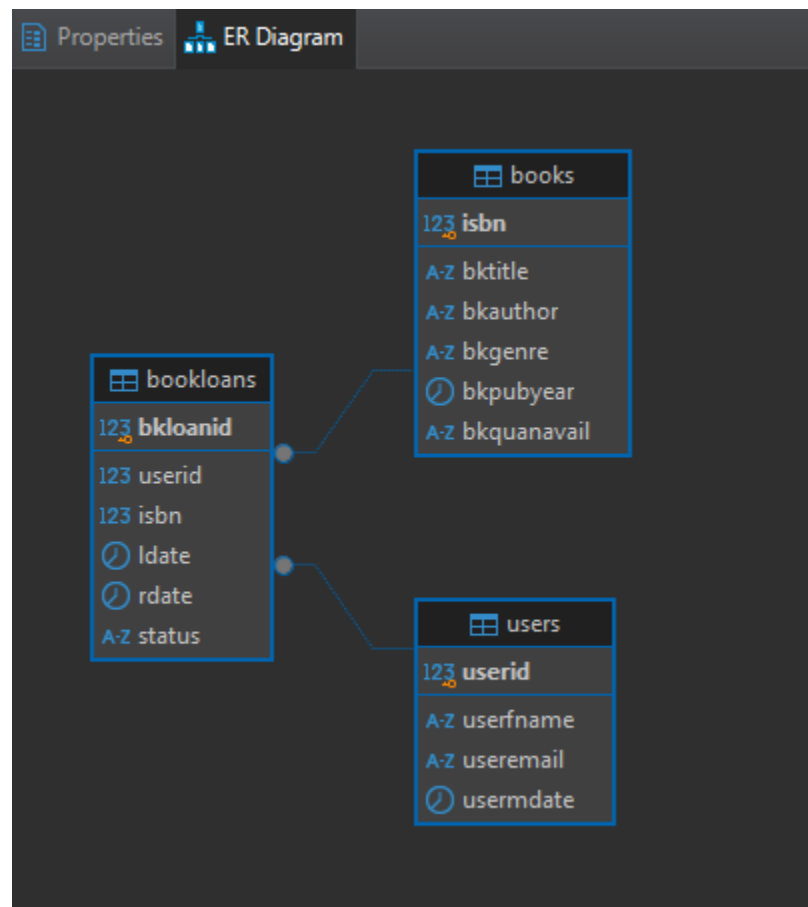
Creation of Users table, I created the book table with required primary key, I've implemented also that the Name must be not null as it is required on the borrow slip to have a name to be traced, as well as their contact number in order for them to be reached out if issues will arrive, making also their email unique as all emails must be unique and cannot be used by the same users. I've input also that the User can put default date depending on how it is depicted, can be an attendance type on the time he is in the library, as well as a membership which is not required, might be not a member but wants to borrow a book, a member of a library, or an expired membership but the librarian allows him to borrow books.

```

create table BookLoans(
BKLoanID serial primary key,
UserID int not null references users (UserID),
ISBN    int not null references books (ISBN),
LDate   date not null,
RDate   date not null,
Status  varchar(255) not null
)

```

The Creation of BookLoans table has an Fk of both UserID from the Users table for whom it is the person that will borrow the book as well as the ISBN is the book that the person will be borrowing. Also, the Loan Date and Return Date is the pact that will be inputted in a borrow slip so it must not be null, also the status of it must be updated according to the presence of the book on the library. Lastly, the BKLoanID is the primary key and a unique identifier for the transaction.



As seen on DBeaver, all my inputs are connected with each other as I've queried the correct correspondent queries.

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.

The screenshot shows a SQL query being executed in a database client. The query is:

```
INSERT INTO Books (BkTitle, BkAuthor, BkGenre, BkPubYear, BkQuanAvail)
VALUES ('The Great Les Paul', 'Les Paul', 'History', '1925-04-10', '5');
```

Below the query editor, the results pane shows the following details:

Name	Value
Query	INSERT INTO Books (BkTitle, BkAuthor, BkGenre, BkPubYear, BkQuanAvail) VALUES ('The Great Les Paul', 'Les Paul', 'History', '1925-04-10', '5')
Updated Rows	1
Execute time	0.001s
Start time	Wed Dec 11 17:25:19 PST 2024
Finish time	Wed Dec 11 17:25:19 PST 2024

Below the results pane, a table view shows the data for the 'Books' table:

isbn	bktitle	bkauthor	bkgenre	bkpubyear	bkquanavail
1	The Great Les Paul	Les Paul	History	1925-04-10	5

- b. Add a new user to the system.

The screenshot shows a SQL query being executed in a database client. The query is:

```
Insert into Users (UserFName, UserEmail, UserMDate)
values ('Sidryl Pudge Stormbreaker', 'SidrylHotdog49@mailers.com', '2024-12-10');
```

Below the query editor, the results pane shows the following details:

Name	Value
Query	Insert into Users (UserFName, UserEmail, UserMDate) values ('Sidryl Pudge Stormbreaker', 'SidrylHotdog49@mailers.com', '2024-12-10')
Updated Rows	1
Execute time	0.003s
Start time	Wed Dec 11 17:29:50 PST 2024
Finish time	Wed Dec 11 17:29:50 PST 2024

Below the results pane, a table view shows the data for the 'Users' table:

userid	username	useremail	usermdate
1	Sidryl Pudge Stormbreaker	SidrylHotdog49@mailers.com	2024-12-10

- c. Record a book loan for a user.

<

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

```
select * from bookloans
where UserID = 1
```

1

Results 2

Statistics 3

Statistics 4

Statistics 5

Results 6

S

from bookloans where UserID = 1 | Enter a SQL expression to filter results (use Ctrl+Space)

123 bkloanid	123 userid	123 isbn	🕒 ldate	🕒 rdate	A-Z status
1	1	1	2024-12-11	2025-01-11	borrowed

Let's just say referring to what I've created above, in which Mr. Sidryl Pudge Stormbreaker borrowed book, so basically just find the UserID on the Users table in which it was 1 as it is a Foreign to the BookLoans, therefore just select all from the table of bokloans where te UserID = 1.

- e. List all overdue loans.

```
select * from BookLoans
where RDate < CURRENT_DATE and Status = 'overdue';
```

5	1	1	2024-12-01	2024-12-09	overdue
---	---	---	------------	------------	---------

Basically the query of this is subjective as for example the date was input on a specific date, just enter on a date that is relative and after the deadline date, let's just say also that the librarian is the one who also monitors the que of the books everyday as it's her job to monitor the status of the books.

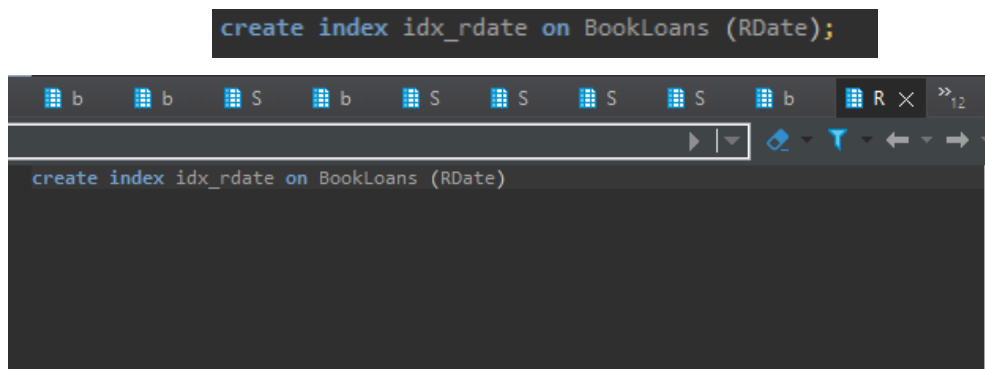
Part 4: Data Integrity and Optimization

4. Explain how you would ensure:
- The prevention of borrowing books when no copies are available. (15 pts)

```
update Books
set BkQuanAvail = 0
where BkTitle = 'The Great Les Paul';
```

I would like to address that as a Librarian, you're the one who is in charge on the book availability in the library, so it's your job that you are the one that will update the availability of the books in the library and as the one who is in charge in database, then so for make use of the **UPDATE statement**.

- Fast retrieval of overdue loans. (20 pts - with CODE and actual screenshot of performance)



First Create an Index that will filter the Book Loans depending on their Return Date also this also helps for faster query execution


```
select * from BookLoans
where RDate < CURRENT_DATE;
```

123 bkloanid	123 userid	123 isbn	ldate	rdate	A-Z status
5	1	1	2024-12-01	2024-12-09	overdue
4	1	1	2024-12-01	2024-12-05	overdue

As a librarian, you're the one checking the dates the overdue as always on the current day, every day. So as a Librarian, it is your task to check the Return Dates that are before your current date to see the books that are simply overdue.

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.

The rising problem I could foresee on this database is by means the querying of people within the database within the table. In can also improve availability in I can suggest partitioning them by years in terms of Book Borrowed so that you can improve the performance in terms of locating a particular query of the related to the attribute of the book that was loaned. According to Charlie Cluster (2023), The partitioning of the database by means can improve the scalability, by partitioning them into years, to make the data easily locatable, improve availability as well as it can be seen on what year did you query, and performance by means of the fast runtime of query and it will not go through the whole database.

Resources:

Cluster(2023).Database Partitioning.
<https://www.cockroachlabs.com/blog/what-is-data-partitioning-and-how-to-do-it-right/>

Harsh Priyadarshi(2024). Benefits in using Varchar.
<https://www.scaler.com/topics/varchar-vs-text-mysql/>