

## Library Database Management System

Project Members: Fahim Tanvir, Jason Lopez, Khalid Issa, Jessica Chen, Abul Hasan

### Introduction:

One of our assignments in our database class was a Georgia-tech Library Case Study:

**Case Study: The Georgia Tech Library (GTL)** has approximately 16,000 members, 100,000 titles, and 250,000 volumes (an average of 2.5 copies per book). About 10% of the volumes are out on loan at any one time. The librarians ensure that the books that members want to borrow are available when the members want to borrow them. Also, the librarians must know how

many copies of each book are in the library or out on loan at any given time. A catalog of books is available online that lists books by author, title, and subject area. For each title in the library, a book description is kept in the catalog; the description ranges from one sentence to several pages. The reference librarians want to be able to access this description when members request information about a book. Library staff includes chief librarian, departmental associate librarians, reference librarians, check-out staff, and library assistants.

Books can be checked out for 21 days. Members are allowed to have only five books out at a time. Members usually return books within three to four weeks. Most members know that they have one week of grace before a notice is sent to them, so they try to return books before the grace period ends. About 5% of the members have to be sent reminders to return books.

Most overdue books are returned within a month of the due date. Approximately 5% of the overdue books are either kept or never returned. The most active members of the library are defined as those who borrow books at least ten times during the year. The top 1% of membership does 15% of the borrowing, and the top 10% of the membership does 40% of the borrowing.

About 20% of the members are totally inactive in that they are members who never borrow.

To become a member of the library, applicants fill out a form including their SSN, campus and home mailing addresses, and phone numbers. The librarians issue a numbered, machine-readable card with the member's photo on it. This card is good for four years. A month before a card expires, a notice is sent to a member for renewal. Professors at the institute are considered automatic members. When a new faculty member joins the institute, his or her information is pulled from the employee records and a library card is mailed to his or her campus address. Professors are allowed to check out books for three-month intervals and have a two-week grace period. Renewal notices to professors are sent to their campus address. The library does not lend some books, such as reference books, rare books, and maps. The librarians must differentiate between books that can be lent and those that cannot be lent. In addition, the librarians have a list of some books they are interested in acquiring but cannot obtain, such as rare or out-of-print books and books that were lost or destroyed but have not been replaced. The librarians must have a system that keeps track of books that cannot be lent as well as books that they are interested in acquiring. Some books may have the same title; therefore, the title cannot be used as a means of identification. Every book is identified by its International Standard Book Number (ISBN), a unique international code assigned to all books. Two books with the same title can have different ISBNs if they are in different languages or have different bindings (hardcover or softcover). Editions of the same book have different ISBNs.

The proposed database system must be designed to keep track of the members, the books, the catalog, and the borrowing activity.

We decided to do a project on the side which is a program written in Java(along with SQL for database) to manage a database similar to the case study. The following is a detailed analysis and summary of our implementation:

## **PART A: Database implementation.**

This is the main schema for the database:

### **1. Universal Schema**

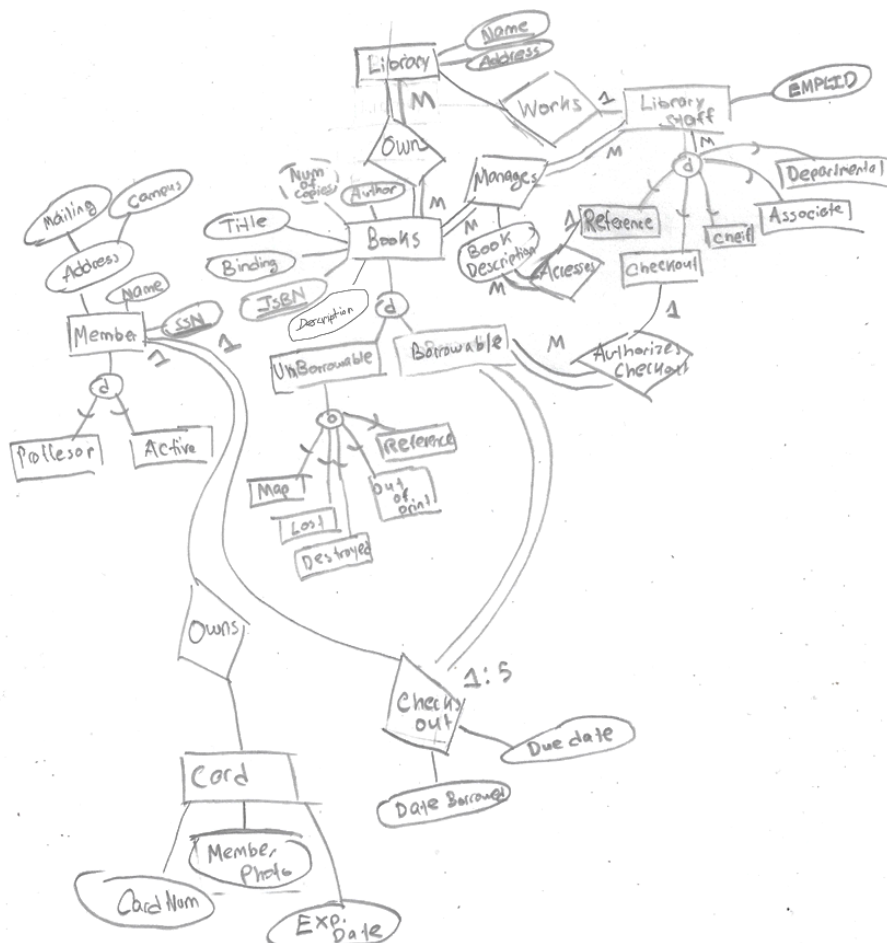
```
Library(  
    varchar ISBN,  
    varchar Author,  
    varchar Title,  
    int NumOfCopies,  
    varchar Binding,  
    varchar BookDescription,  
    varchar Language,  
  
    int SSN,  
    varchar MemberName,  
    varchar Address,  
    varchar Role,  
  
    int CardNumber,  
    varchar CardName,  
    Photo,  
    date ExpirationDate,  
  
    int EMPLID,  
    varchar Position,  
    varchar LibrarianName,  
  
    date CheckoutDate,  
    date DueDate  
)
```

## 2. Dependency Set (Functional Dependencies)

Here's the set of functional dependencies :

1. ISBN  $\rightarrow$  Author, Title, NumberOfCopies, Binding, Description, Language
2. BookISBN  $\rightarrow$  BookTitle
3. SSN, MemberID  $\rightarrow$  MemberName, Address, Role
4. CardNumber  $\rightarrow$  CardholderName, Photo, ExpirationDate
5. EMPLID  $\rightarrow$  Position, LibrarianName
6. COMemberName, AccountID  $\rightarrow$  COBookTitle, COISBN, CheckoutDate, ReturnDeadline

## 3.) ER DIAGRAM



#### 4.) Relationships for each table in schema

##### Relation 1: Books

Books(ISBN, Author, Title, NumOfCopies, Binding, Description, Language)

##### Relation 2: Member

Member(SSN, MemberName, Address, Role)

##### Relation 3: Card

Card(CardNumber, CardName, Photo, ExpirationDate)

##### Relation 4: Librarian

Librarian(EMPLID, Position, LibrarianName)

##### Relation 5: CheckDescription

CheckDescription(CDEMPLID, Position, CDLibrarianName, CDMemberName, Book Description, CDISBN)

##### Relation 6: CheckOut

CheckOut(MemberName, CardNumber, COBookTitle, COISBN, CheckoutDate, ReturnDeadline)



**Member Schema**

Column Name	Data Type	Primary Key	Foreign Key
SSN	Int	SSN	
Address	VarChar		
MemberName	Int		
Role	VarChar		

**Librarian Schema**

Column Name	Data Type	Primary Key	Foreign Key
EMPLID	Int	EMPLID	
Position	VarChar		
Librarian Name	VarChar		

**Card Schema**

Column Name	Data Type	Primary Key	Foreign Key
Card Number	Int	Card Number	SSN(In Member Schema)
Card Name	VarChar		MemberName(In Member Schema)
Photo			
Expiration Date	Date		

**Description Schema**

Column Name	Data Type	Primary Key	Foreign Key
-------------	-----------	-------------	-------------

EMPLID	Int	EMPLID	EMPLID(In Librarian Schema)
Position	VarChar		
CDLibrarianName	VarChar		Name(In Librarian Schema)
CDMember Name	VarChar		Member(In Member Schema)
Book Description	VarChar		Description(In Book Schema)

### **Book Schema**

Column Name	Data Type	Primary Key	Foreign Key
ISBN	Int	ISBN	
Author	VarChar		
Title	VarChar		
NumCopies	Int		
Binding	VarChar		
Description	VarChar		
Language	VarChar		

### **Checkout Schema**

Column Name	Data Type	Primary Key	Foreign Key
COMemberName	VarChar		MemberName( In Member Schema)
COMemberNumber	Int		
Book Title	VarChar		

COISBN	VarChar	COISBN	COISBN(In Book Schema)
CheckoutDate	Date		
Return Deadline	Date		

## 5. List of Features

The key features for our database is to manage records

- Creating Records will need these:
  - Books
  - Members
  - Cards
  - Librarians
  - Checkout Entries
  - Members near the end of an expiring membership
- Updating records will need these:
  - Modify details about the book
  - Member information details (such as address, phone number, etc.)
  - Card Expiration
  - Records of books being checked in and out of the library to see its availability
- Deleting records:
  - Remove outdated records of books the library may no longer be able to use(Unborrowable)
  - Delete expired membership cards (including faculty memberships if they no longer work there/expire)
  - Delete librarian information/access if they no longer work for the library
- Queries & Reports:
  - Able to filter or search for books by title or author information
  - Track books currently checked out, number of copies that are available
  - Create a list of overdue books based on ReturnDeadline.
  - Description of the books when members ask the reference librarians
  - If a member/applicant of membership is a professor or not(hence the role attribute in member) so they can get their cards automatically or by applying.
  - How many books members have borrowed and the due dates of the checkouts.
  - Generate information on most wanted books based on what is unborrowable.

## Part B: How it works as a application

### SQL IMPLEMENTATION

Hasan, Chen and Lopez helped set up SQL, albeit we have different operating systems, we have found a common ground working between the terminal and a AWS cloud server running MYSQL. After that we had to translate our schema into a fully working SQL database using line commands.

By setting up the mySQL Database, we managed to use the schema from the first part of the project as a blueprint of the entities and their relations between each other. Lopez worked on creating the Database on MacOS as a first step, while Hasan and Chen worked on MYSQL code translation. Chen worked on inserting book data. When established, the schema was then translated into SQL code. We began with creating a table for Books, Member, Card, Librarian, CheckDescription, & Checkout alongside their respective relations between themselves.

```
▷ Run
CREATE DATABASE librarydb

▷ Run | ⌘Select
CREATE TABLE Books (
  ISBN VARCHAR(20) PRIMARY KEY,
  Author VARCHAR(100),
  Title VARCHAR(150),
  NumOfCopies INT,
  Binding VARCHAR(50),
  Description TEXT,
  Language VARCHAR(50)
);
```

```
▷ Run | ⌘Select
CREATE TABLE Member (
  SSN INT PRIMARY KEY,
  MemberName VARCHAR(100),
  Address VARCHAR(200),
  Role VARCHAR(50)
);

▷ Run | ⌘Select
CREATE TABLE Card (
  CardNumber INT PRIMARY KEY,
  CardName VARCHAR(100),
  Photo BLOB,
  ExpirationDate DATE,
  SSN INT,
  FOREIGN KEY (SSN) REFERENCES Member(SSN)
);
```

```
▷ Run | ⌘Select
CREATE TABLE CheckOut (
  COMemberName VARCHAR(100),
  COMemberNumber INT,
  COBookTitle VARCHAR(150),
  COISBN VARCHAR(20),
  CheckoutDate DATE,
  ReturnDeadline DATE,
  PRIMARY KEY (COMemberNumber, COISBN),
  FOREIGN KEY (COISBN) REFERENCES Books(ISBN),
  FOREIGN KEY (COMemberNumber) REFERENCES Card(CardNumber)
);
```

```
CREATE TABLE Librarian (
  EMPLID INT PRIMARY KEY,
  Position VARCHAR(100),
  LibrarianName VARCHAR(100)
);

▷ Run | ⌘Select
CREATE TABLE CheckDescription (
  CDEmplID INT,
  CDLibrarianName VARCHAR(100),
  CDMemberName VARCHAR(100),
  BookDescription TEXT,
  CDISBN VARCHAR(20),
  Position VARCHAR(100),
  PRIMARY KEY (CDEmplID, CDISBN),
  FOREIGN KEY (CDEmplID) REFERENCES Librarian(EMPLID),
  FOREIGN KEY (CDISBN) REFERENCES Books(ISBN)
);
```

Tanvir and Issa would later test out the SQL for their Java implementation later on.

To begin with our process of implementation, we begin with the Books table, the Member table, & the Librarian table as they have no foreign key. The Member table uses the SSN as a primary key, which is then referenced by the Card table. With these tables initially set up, we set up the relations



between the entities, specifically the ones with foreign keys referenced by the tables we just created. The CardNumber attribute in the Card table is referenced by the SSN attribute. Following that, the CheckDescription table was created with two foreign keys referencing the Librarian table (EMPLID) as well as the Books table (ISBN). The final table created was the Checkout table, then referencing the Books table (ISBN) and the Card table (CardNumber).

With these tables in place we have used the schema from part 1 into a fully working database using mySQL, set up along with the relationships between these entities so that the user can set up any queries to add a member, checkout a book, or inquire about details over a cer

## JAVA IMPLEMENTATION

Issa worked with Tanvir in the Java implementation. They both assisted each other in programming it and Issa helped test it out using the “MySQL extension” on VScode. When the Java application is loaded, it launches a graphical user interface or a GUI designed to manage a library system. This enabled direct interaction with the database to verify functionality and ensure reliable behavior of the application. This resulted in a user-friendly, interactive system that integrates the GUI components with real-time database access.

These resources were used for help and configuration:

[https://www.youtube.com/watch?v=ZhljL\\_0gm04](https://www.youtube.com/watch?v=ZhljL_0gm04)(Provided by Fahim Tanvir)

<https://www.tutorialspoint.com/jdbc/jdbc-sample-code.htm>(Provided by Fahim Tanvir)

<https://www.geeksforgeeks.org/java-database-connectivity-with-mysql/>(Provided by Khalid Issa)

<https://www.youtube.com/watch?v=AHFBPxWebFQ>(Provided by Khalid Issa)

It uses swing to display data generated in our database. It constructs and executes SQL queries to retrieve, insert, or remove data, ensuring operations that involve multiple parts (like creating a new member and their card) are handled within transactions to maintain data consistency, and provides feedback. It looks at the foreign and partial keys so if a change were to conflict with that(let's say someone wants to delete a member but that member is apart of a checkout order, that change won't happen). It took us a bit to have JDBC(the thing that connects java and sql server together) as we were having trouble working with it as we had to manually figure out where to configure it in our buildpath. Without it, the sql still worked fine, it would just give an error as it could've found the JDBC driver.

The image below shows the book's overall report like the ISBN, Author, Title, Number of copies, Binding, Description and the Language it is written in

File Reports						
Search:			Search Books			
ISBN	Author	Title	NumOfCopies	Binding	Description	Language
978-0-123456-47-2	Alice Johnson	Sample Book 3	10	Ebook	Yet another sample book descri...	English
978-0987654321	George R.R. Martin	A Game of Thrones	3	Paperback	The first book in the A Song of Ic...	English
978-1-234-56789-7	Jane Smith	Sample Book 2	3	Paperback	Another sample book description.	English
978-1111111111	J.R.R. Tolkien	The Hobbit	4	Hardcover	A hobbit's adventure through Mi...	English
978-1234567890	J.K. Rowling	Harry Potter and the Sorcerer's S...	5	Hardcover	A young wizard's journey begins.	English
978-3-16-148410-0	John Doe	Sample Book 1	5	Hardcover	A sample book description.	English

Library Management System - (3) Overdue Books!			
File Reports			
Search:		Search Books	
MemberName	BooksCheckedOut	LatestDueDate	Status
Alice Johnson	4	2025-05-15	OK
John Doe	1	2023-10-15	OK
Jane Smith	1	2023-10-20	OK
Bob Smith	0		OK

For the above image, it displays the Member's name, the amount of books they checked out along with the due date and the status of their membership

Library Management System - (3) Overdue Books!						
File Reports						
Search:		Search Members + Cards				
SSN	MemberName	Role	Address	CardNumber	CardName	ExpirationDate
101	Alice Johnson	Professor	123 Maple St	201	Alice Johnson	2025-12-31
102	Bob Smith	Student	456 Oak Ave	202	Bob Smith	2024-11-30
123456789	John Doe	Student	123 Main St, Cityville	1001	John Doe	2025-12-31
555555555	Alice Johnson	Staff	789 Oak St, Villageville	1003	Alice Johnson	2023-11-15
987654321	Jane Smith	Faculty	456 Elm St, Townsville	1002	Jane Smith	2024-06-30

Library Management System - (3) Overdue Books!		
File Reports		
Search:		Search Librarians
EMPLID	Position	LibrarianName
1	Head Librarian	Emily Brown
2	Assistant Librarian	Michael Green
3	Library Technician	Sarah White

For the above two images, the top image displays the member's personal information such as their SSN, role, address, card number and expiration date. The bottom image displays their EMPLID, Positions and their name

Library Management System - (3) Overdue Books!

File Reports

Search:  Search CheckDescriptions

CDEMLID	CDLibrarianName	CDMemberName	BookDescription	CDISBN	Position
1	Emily Brown	John Doe	Sample book description for Jo...	978-3-16-148410-0	Head Librarian
2	Michael Green	Jane Smith	Sample book description for Ja...	978-1-234-56789-7	Assistant Librarian
3	Sarah White	Alice Johnson	Sample book description for Ali...	978-0-123456-47-2	Library Technician

Library Management System - (3) Overdue Books!

File Reports

Search:  Search CheckOuts

COMemberName	COMemberNumber	COBookTitle	COISBN	CheckoutDate	ReturnDeadline	ReturnDate
Alice Johnson	201	Harry Potter and the Sorce...	978-1234567890	2025-05-01	2025-05-15	
John Doe	1001	Sample Book 1	978-3-16-148410-0	2023-10-01	2023-10-15	
Jane Smith	1002	Sample Book 2	978-1-234-56789-7	2023-10-05	2023-10-20	
Alice Johnson	1003	Sample Book 3	978-0-123456-47-2	2023-10-10	2023-10-25	

For the above two images, the images display the checkout for overdue books with the top image containing the EMPLID, Librarian Name, Member Name, the description, ISBN and position. The bottom image has the member's name, number, the book title they borrowed, the ISBN from the specific books, the checkout date and return deadline.

Library Management System - (3) Overdue Books!						
File Reports						
Search:			Search	Overdue Books		
COMemberName	COMemberNumber	COBookTitle	COISBN	CheckoutDate	ReturnDeadline	Address
John Doe	1001	Sample Book 1	978-3-16-148410-0	2023-10-01	2023-10-15	123 Main St, Cityville
Jane Smith	1002	Sample Book 2	978-1-234-56789-7	2023-10-05	2023-10-20	456 Elm St, Townsville
Alice Johnson	1003	Sample Book 3	978-0-123456-47-2	2023-10-10	2023-10-25	123 Maple St
Alice Johnson	1003	Sample Book 3	978-0-123456-47-2	2023-10-10	2023-10-25	789 Oak St, Villageville

Library Management System - (3) Overdue Books!						
File Reports						
Search: Tolkien			Search	Books		
ISBN	Author	Title	NumOfCopies	Binding	Description	Language
978-1111111111	J.R.R. Tolkien	The Hobbit	4	Hardcover	A hobbit's adventure throu...	English

Library Management System - (3) Overdue Books!						
File Reports						
Search:			Search	Overdue Books		
COMemberName	COMemberNumber	COBookTitle	COISBN	CheckoutDate	ReturnDeadline	Address
John Doe	1001	Sample Book 1	978-3-16-148410-0	2023-10-01	2023-10-15	123 Main St, Cityville
Jane Smith	1002	Sample Book 2	978-1-234-56789-7	2023-10-05	2023-10-20	456 Elm St, Townsville
Alice Johnson	1003	Sample Book 3	978-0-123456-47-2	2023-10-10	2023-10-25	123 Maple St
Alice Johnson	1003	Sample Book 3	978-0-123456-47-2	2023-10-10	2023-10-25	789 Oak St, Villageville

For the above three images, the first one contains the checkout information like the member's name, number, book title, ISBN, checkout date, return deadline and their address. The second one just contains the basic information like the book's title, author, ISB, Binding , description and the language it is written in. The third one contains the checkout information much like the first image.

Library Management System - (3) Overdue Books!						
File Reports						
Search:			Search	CheckOuts		
COMemberName	COMemberNumber	COBookTitle	COISBN	CheckoutDate	ReturnDeadline	
Alice Johnson	201	Harry Potter and the Sorcerer's Stone	978-1234567890	2025-04-01	2025-04-15	
hald isa	203	211 notes	211	2025-05-10	2025-05-11	
hald isa	203	211 notes	211	2025-05-01	2025-05-01	
hald isa	203	Culinary Arts	2025-05-11	2025-05-11	2025-05-25	
hald isa	203	Introduction	2025-05-09	2025-05-11	2025-05-25	
hald isa	203	Harry Potter	2025-05-11	2025-05-11	2025-05-25	

Member Name:

hald isa

Card Number (integer):

203

Book Title:

240

Book ISBN:

240 notes

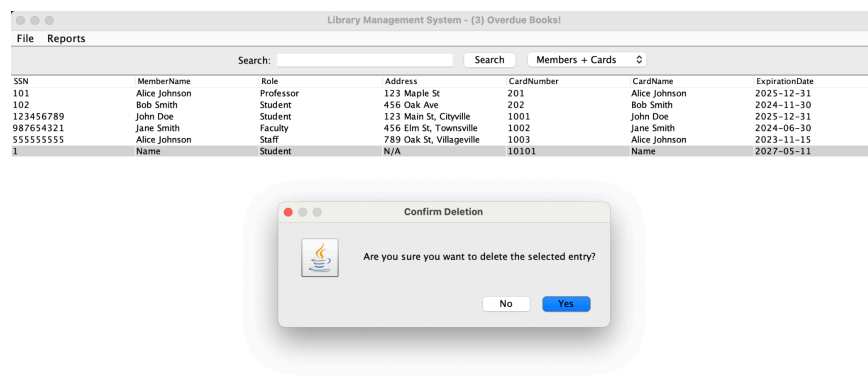
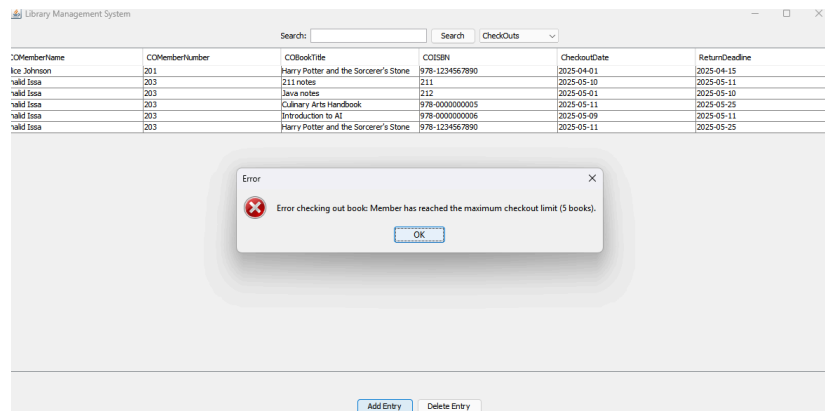
Checkout Date (YYYY-MM-DD):

2025-05-11

Return Deadline (YYYY-MM-DD):

2025-05-25

OK Cancel



For the above three images, the bottom displays the SSN, the Member's name, role, address, card number and the expiration date for their card/membership. It also shows how members are limited to 5 books. The top displays a checkout situation with a name( in this case 'Khalid') as an example.

In conclusion, some of the stuff could've been implemented better or more streamlined. There was also stuff we wanted to include but didn't get around to such as an expiration and renew system for member cards. Also, we discovered the code can still delete some stuff that we don't want deleted such as overdue books(which will delete the checkout order in conjunction with that). But so far, some of our goals regarding functionality within our database have been met and each member contributed a fair amount.