# Assignment Title: Database System Design for a Library

## Introduction

### Overview of the Project

This project deals with the designing and implementation of an efficient database management system for a public library which is to be executed in MYSQL. The key tasks of the project are:

* Drawing of an Entity-Relationship (ER) diagram.
* The implementation of the database schema using MySQL, and
* Presenting of the advanced SQL queries to manage and query the data. Also, it provides a theoretical section on the CAP theorem concerning the database system.

### Importance of a Database System for a Public Library

The database system of a public library is an essential component since it facilitates the organization of the library’s activities. This system allows the organization to monitor the stock of the books, control the patron details, follow the borrowing and returning process and besides manage reservations. This means that when the processes are enacted, the improvement of service delivery to users is enhanced, the accountability of the availability of resources is ensured, and record-keeping is maintained. Besides, the correct organization of the database allows for the improvement of data quality, eliminates the problem of data duplication, and promotes the efficient use of data for decision-making.

### Objectives

The overall goal of this report is to gain hands on experience of creating and populating a database, and querying the data using MySQL. The tasks are structured to achieve the following goals:

* Design a Database System: Developing a detailed ER diagram that depicts the library’s data need and relationship.
* Implement the Database: Designing schema in MySQL basically by creating the tables and relations/ constraints and adding a set of sample data that fit the library environment at large.
* Write Advanced SQL Queries: Providing examples of using SQL in Writing data manipulation queries and Retrieval operations/Join operations/sub queries.
* Discuss Theoretical Concepts: Explaining the CAP theorem and some of the issues which will be relevant to the library’s database system, and how it is best to compromise one between consistency, availability, and partition tolerance.

## Database Design

### Scope and Objectives

#### Detailed Scope of the Database

This database focuses more specifically on who, what, where, how and why of various subjects. The scope of the database system for the public library encompasses the following areas:

* Book Management: Shelving of the books, keeping record on the copies in store, and processing of reserved books.
* Patron Management: Permanent registration of user details, Identification card details, correspondent address, Library membership type etc.
* Transaction Management: Cataloguing the usage of the borrowed books, reminding borrowers of the date the books need to be returned and collecting money for overdue books.
* Staff Management: Archiving all the library staff, their appointments, and dealing with the system.
* Reservation System: The processes involved in handling the reservation of the books by the patrons as well as handling the status of the reservation.

#### Goals Taking into Account of the Stakeholders, Needs, and Functions

#### Stakeholders

The key stakeholders of the public library database system include:

* Library Users (Patrons): Those who attend the library and make use of the available materials through borrowers’ check out, bookings, and use of other library services.
* Library Employees: This group consists of staffs who are involved in the selection, acquisition, processing, organization of collections, and providing user support that are basic to the functioning of a library as well as administrative and technical officers who are involved in the day-to-day running and co-ordination of the library.
* Library Executives: Community of leaders within the organization headed by a director who is charge of the over-all planning, direction, administration and setting the strategic vision and policies of a library.

#### Requirements

For a library environment and particularly a public library, the database system must be able to meet the following critical success factors for it to work smoothly: These include, improve library functions, and improve patrons experience and accuracy and consistencies in the data processing. The key requirements are:

1. Efficient Cataloging and Retrieval of Books: They include the ability to rapidly and efficiently capture books’ metadata by possibly indexing through barcode or ISBN number, the ability for the users to efficiently search the database to gain access to book details.
2. Accurate and Up-to-Date Records of Patrons and Their Borrowing History: Peculiar records of the patrons including the records of borrowed material is a very crucial aspect in library management and user services hence require up to date records.
3. Streamlined Transaction Processing for Borrowing and Returning Books: This is another operational requirement which implies that the use of the database in the processing of book borrowing, and returning transactions should not be time consuming.
4. Transparent Reservation System for Books: Self-service reservation should be easy and straightforward, as well as incorporate a system for book selection and follow up on book status.
5. Comprehensive Management of Staff Roles and Activities: The staff roles and activity to be performed in the library should be properly managed to enable the library run effectively through the direction of specific roles and efficient working of jobs.

### Entity-Relationship (ER) Diagram

Explanation of the Entities, Attributes, Relationships, and Cardinalities

1. Entities:

- Book: ‘**BookID’**, ‘**Title’**, ‘**Author**’, ‘**Publisher**’, ‘**YearPublished**’, ‘**ISBN**’, ‘**Genre**’, ‘**CopiesAvailable’**

- Patron: ‘**PatronID**’, ‘**FirstName**’, ‘**LastName**’, ‘**DOB**’, ‘**Address**’, ‘**Phone**’, ‘**Email**’

- Staff: ‘**StaffID**’, ‘**FirstName**’, ‘**LastName**’, ‘**Position**’, ‘**Email**’, ‘**Phone**’

- Transaction: ‘**TransactionID**’, ‘**BookID**’, ‘**PatronID**’, ‘**StaffID**’, ‘**BorrowDate**’, ‘**ReturnDate**’, ‘**Fine**’

- Reservation: ‘**ReservationID**’, ‘**BookID**’, ‘**PatronID**’, ‘**ReservationDate**’, ‘**Status**’

2. Relationships:

1. Book - Transaction: One-to-Many (A book can be involved in multiple transactions)

2. Patron - Transaction: One-to-Many (A patron can have multiple transactions)

3. Staff - Transaction: One-to-Many (A staff member can handle multiple transactions)

4. Book - Reservation: One-to-Many (A book can have multiple reservations)

5. Patron - Reservation: One-to-Many (A patron can make multiple reservations)

3. Cardinalities:

- Each book can have multiple transactions, but each transaction involves only one book.

- Each patron can make multiple transactions, but each transaction involves only one patron.

- Each staff member can handle multiple transactions, but each transaction is handled by one staff member.

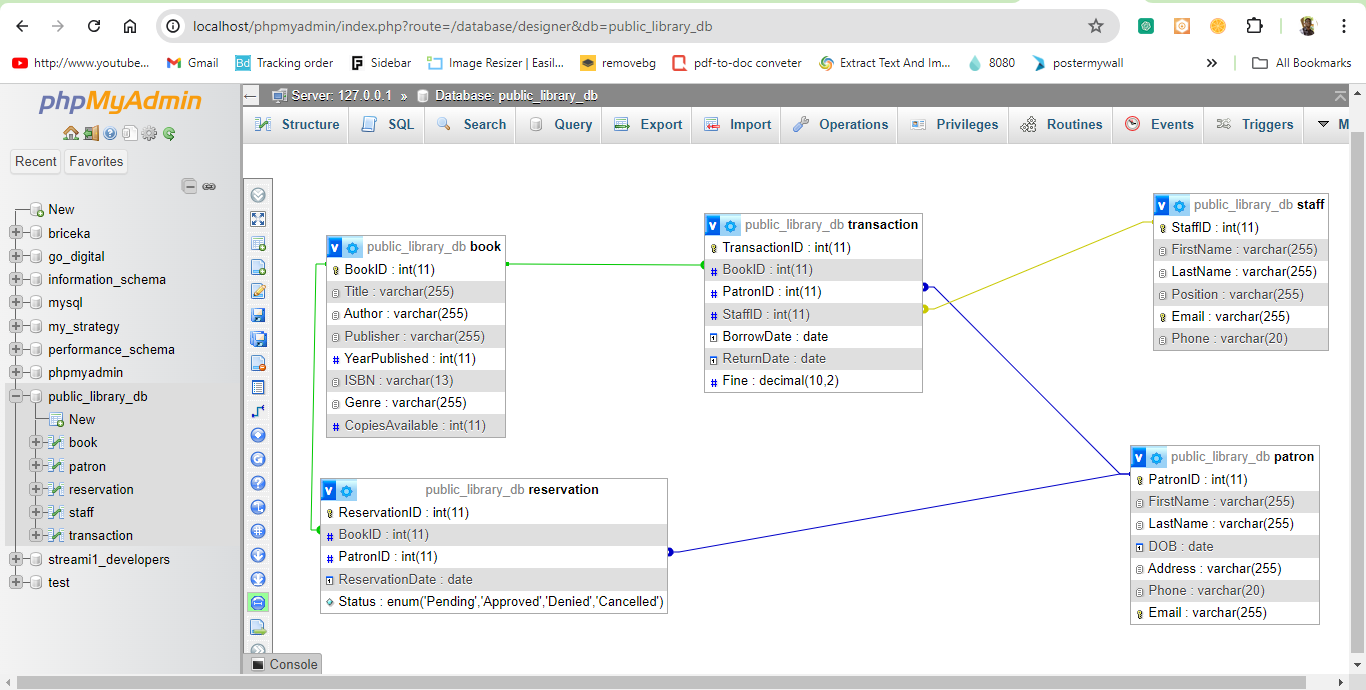
- Each book can have multiple reservations, but each reservation is for one book.

- Each patron can make multiple reservations, but each reservation is made by one patron.

### Justification of Design Choices Based on Library Needs

Just as requirements of the operation of the public library are met when designing the system, the design of the database system for the public library is unique to the operation of the library. This includes the choice of the entities involved, the characteristics of these entities, the relationships between the entities, and the varying cardinalities displayed so as to address all the basic functional needs of a library.

* Entity Selection: More of the entities represent the fundamental functioning of the library with regards to its books, patrons, monetary transactions and employees. These entities provide on the key areas that are needed in the efficient management of a library.
* Attributes: For attributes four or five attributes are chosen for each entity to record as much relevant information as needed. For example, for book details one can provide author and publisher while details of the patron can include the phone number in order to reach the latter.
* Relationships: They depict contact or interactions within and between the entities within the context of the library. For instance, the interaction between the books and the transactions guarantees that the book borrowing and returning processes have been recorded correctly.
* Cardinalities: The cardinalities also aid in the capacity of the database to accommodate multiple interactions for the likes of transactions for a specific book and reservations to a specific patron by more than one person.



Title: Visual Representation of the ER Diagram

### Principles of Database Design

Fundamental Guidelines of Database Design are general rules that need be followed to ensure databases are well designed, with accurate data arrangement and timely performance. They tell where, how, and even under what circumstances data must be placed and how relationships between entities must be defined to ensure the database’s flexibility and scalability. When applied appropriately, the presented principles can be used to construct efficient and scalable database systems which deliver data managerial and utilization capabilities suitable to organizations’ requirements. The principles are as follows:

1. Normalization: Specifically, the design of the database does not contain any redundancy, which is consistent with normalization theories. For instance, while patronizing a store, it is recommended to isolate patron and transactions data into two tables to eliminate repetition through storing space and increasing general upkeep.
2. Data Integrity: It uses types of constraints to keep referential integrity,; Primary and foreign key constraints. These restrictions guarantee that every transactional process, book reservations, and operations performed by the staff will be connected to a patron at the same time reducing the formation of orphan records in the database.
3. Scalability: Read access scalability is achieved through generalized record definitions that allow for the insertion of new volumes with their details into the system without the need to redesign the database. This scalability is particularly important for expanding business operations and enhancing services to meet the increasing demands of library operations.
4. Flexibility: We ensure that changes in the course of executing the design are well encompassed and implemented adequately. It enables one to introduce new attributes to entities or create new relationships between tables without expanding most of the structure of the database model. This flexibility is important in coping with changes in functional specifications and business rules of the Library.
5. Performance: Occasionally significant features like BookID, PatronID, and TransactionID are indexed to aid in faster querying of the database. Indexing provides skills and means for orientation and manipulation of data in storage and retrieval processes as well as information necessary for effective library functioning including book stock, patron records, and transactions.

Through the application of the principles involved in the design of the database, the management of operations of a public library is made to be optimized, enhanced in terms of stability and made scalable. Data accuracy or standardization is ensured; duplication of work is prevented; scalability is provided; the data center can easily accommodate changes and increase efficiency, hence improving organizational utility and customer satisfaction.

## Implementation

### Database Implementation in MySQL

#### SQL Code for Creating the Database and Tables

```sql

-- Create the database if it doesn't exist

CREATE DATABASE IF NOT EXISTS public\_library\_db;

-- Use the created database

USE public\_library\_db;

-- Table: Book

CREATE TABLE Book (

BookID INT PRIMARY KEY AUTO\_INCREMENT,

Title VARCHAR(255) NOT NULL,

Author VARCHAR(255) NOT NULL,

Publisher VARCHAR(255) NOT NULL,

YearPublished INT,

ISBN VARCHAR(13),

Genre VARCHAR(255),

CopiesAvailable INT DEFAULT 0

);

-- Table: Patron

CREATE TABLE Patron (

PatronID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(255) NOT NULL,

LastName VARCHAR(255) NOT NULL,

DOB DATE,

Address VARCHAR(255),

Phone VARCHAR(20),

Email VARCHAR(255) UNIQUE

);

-- Table: Staff

CREATE TABLE Staff (

StaffID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(255) NOT NULL,

LastName VARCHAR(255) NOT NULL,

Position VARCHAR(255) NOT NULL,

Email VARCHAR(255) UNIQUE,

Phone VARCHAR(20)

);

-- Table: Transaction

CREATE TABLE Transaction (

TransactionID INT PRIMARY KEY AUTO\_INCREMENT,

BookID INT NOT NULL,

PatronID INT NOT NULL,

StaffID INT,

BorrowDate DATE NOT NULL,

ReturnDate DATE,

Fine DECIMAL(10,2),

FOREIGN KEY (BookID) REFERENCES Book(BookID),

FOREIGN KEY (PatronID) REFERENCES Patron(PatronID),

FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)

);

-- Table: Reservation

CREATE TABLE Reservation (

ReservationID INT PRIMARY KEY AUTO\_INCREMENT,

BookID INT NOT NULL,

PatronID INT NOT NULL,

ReservationDate DATE NOT NULL,

Status ENUM('Pending', 'Approved', 'Denied', 'Cancelled'),

FOREIGN KEY (BookID) REFERENCES Book(BookID),

FOREIGN KEY (PatronID) REFERENCES Patron(PatronID)

);

```

### Description of the Tables and their Links

* Book: Contains data such as the title of the book, the author, the publisher, the year of publication, the ISBN number, the type of the book and the amount of stock available.
* Patron: Contains information about the patron including name, date of birth, physical address, phone number, and email.
* Staff: Stores the staff details like name, post, email id, and contact no.
* Transaction: Documents the borrowing and returning of books with details that include – `BookID`, `PatronID`, `StaffID, date borrowed and returned` and fines paid.
* Reservation: Record book reservation whereby particulars such as reserved book, patron name and identification number (`PatronID`), date of reservation and reservation status (`Status`) are documented.

### Sample Data

```sql

-- Place dummy data into Book table

INSERT INTO Book (Title, Author, Publisher, YearPublished, ISBN, Genre, CopiesAvailable)

VALUES

('The Great Gatsby', 'F. Scott Fitzgerald', 'Scribner', 1925, '9780743273565', 'Classic Fiction', 5), ('To Kill a Mockingbird', 'Harper Lee', 'Harper Perennial Modern Classics', 1960,'9780061120084', 'Classic Fiction', 3),

('1984', 'George Orwell', 'Signet Classic', 1949, '9780451524935', 'Dystopian Fiction', 7),

('Pride and Prejudice', 'Jane Austen', 'Penguin Classics', 1813, '9780141439518', 'Classic Fiction', 4);

-- Populate Patron table with example data

INSERT INTO Patron (FirstName, LastName, DOB, Address, Phone, Email) VALUES

VALUES

John, Doe, 1990-05-15, 123 Main St, Anytown, +1234567890, john. doe@email. com

VALUES ( ‘Jane’, ‘Smith’, ‘1985-09-20’, ‘456 Elm St, Othertown’, ‘+1987654321’, ‘jane. smith@email. com’ );

-- Populate Staff table with sample data

SELECT FirstName, LastName, Position, Email, Phone FROM Staff

VALUES

Emily, Jones, Librarian, emily. jones@email. com , +1122334455

,(‘Michael’, ‘Brown’, ‘Assistant Librarian’, ‘michael. brown@email. com’, ‘+9988776655’);

-- Place the sample data into the Transaction table

TRANSACTION: (BookID, PatronID, StaffID, BorrowDate, ReturnDate, Fine)

VALUES

‘(1, 1, 1, ‘2024-06-15’, ‘2024-06-22’, 0. 00)’

INSERT INTO appointments VALUES (3, 2, 2, ‘2024-06-18’, NULL, 0. 00);

-- Populate Reservation table with sample data

INSERT INTO Reservation (BookID, PatronID, ReservationDate, Status)

VALUES

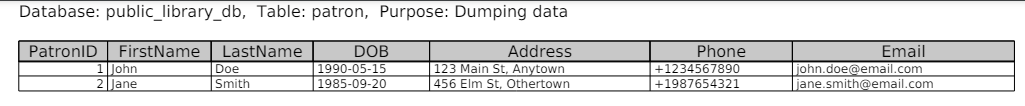
(2, 1, '2024-06-20', 'Pending'),

(4, 2, '2024-06-21', 'Approved');

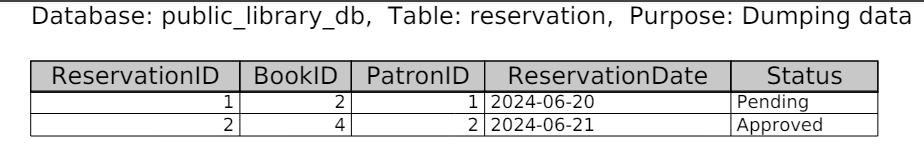
```



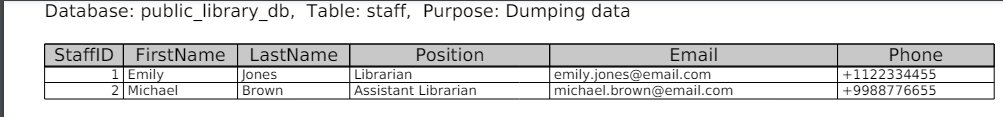
Title: Book table



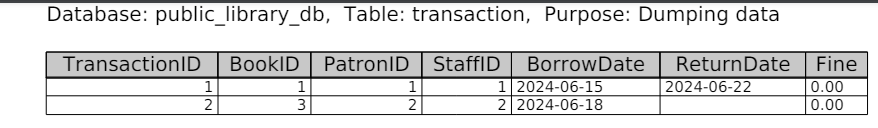
Title: Patron table



Title: Reservation table



Title: Staff table



Title: Transaction table

### SQL Queries

Here are examples of SQL queries that manipulate and retrieve data from the library system

1. Retrieve all books currently available:

```sql

-- All Books Available in the Store:

SELECT \* FROM Book WHERE CopiesAvailable > 0;

```

**Purpose**: Shows all books that are in a circulation to patrons of the facility.

2. Retrieve details of all overdue transactions:

```sql

SELECT T.TransactionID, B.Title, P.FirstName, P.LastName, T.BorrowDate, T.ReturnDate, T.Fine

FROM Transaction T

INNER JOIN Book B ON T.BookID = B.BookID

INNER JOIN Patron P ON T.PatronID = P.PatronID

WHERE T.ReturnDate IS NULL AND T.BorrowDate < CURDATE() - INTERVAL 14 DAY;

```

**Purpose**: Lists transactions where books have not been returned within 14 days, showing details of the book, patron, borrowing and return dates, and any fines (`Fine`) incurred.

3. Count the number of reservations per book:

```sql

SELECT B.Title, COUNT(\*) AS ReservationCount

FROM Reservation R

INNER JOIN Book B ON R.BookID = B.BookID

GROUP BY B.Title;

```

**Purpose**: Provides a count of how many times each book has been reserved, aiding in understanding popular titles.

4. Update the status of a reservation to 'Confirmed:

```sql

UPDATE Reservation

SET Status = 'Approved'

WHERE ReservationID = 1;

```

**Purpose**: Changes the status of a specific reservation (identified by `ReservationID`) from 'Pending' to 'Approved' once the book is ready for pickup.

5. Calculate total fines collected:

```sql

SELECT SUM(Fine) AS TotalFines

FROM Transaction

WHERE ReturnDate IS NOT NULL;

```

**Purpose**: Computes the total amount of fines collected from returned books, providing financial oversight for the library.

Each query fulfills specific operational needs within the library system, from managing inventory (Query 1) and tracking overdue books (Query 2) to assessing popularity through reservation counts (Query 3) and financial oversight via fine calculations (Query 5).

## CAP Theorem Discussion

### Components of the CAP Theorem

The CAP theorem, proposed by Eric Brewer, states that in a distributed data store, only two out of the following three guarantees can be achieved simultaneously: consistency, availability, and partition tolerance.

**Consistency**: One of the crucial aspects of data management is to make sure that all nodes are subscribed to the same data. Read any operand in a consistent system and what you get is the result of the latest write.

**Availability**: This helps to give a feedback to every request, to show that the system is still functional even though some nodes are down. It ensures that it is always available to accept the service requests.

**Partition Tolerance**: Partition tolerance option makes sure that the system can operate even if there are partitions on the network. It means that it can operate correctly regardless of the state of partitioning of the network.

### Relevance to the Library’s Database System

Awareness of the CAP theorem components’ relation to the library database system aids in the creation of an effective and optimal system.

1. Consistency: To ensure the records of books’ availability and borrowers are accurate, consistency for the library is the key. To avoid problems like double bookings, or if a fine has to be levied, ensures all users/staff entering the data view exactly the same.
2. Availability: High availability also means that the services of the library are available to patrons and to the staff consistently. It should allow the users to search, make bookings and even make transactions on books without any interruption.
3. Partition Tolerance: Although network partition occurs more frequently in a centralized library system, it is important to have the ability to handle such issues to ensure that the system is still functional during the network breakage.

### Impact on Design and Implementation

The creation of a library database system requires the consideration of trade-offs in the light of CAP theorem components. Decisions made will determine how the system manages data consistency, its availability, and tolerance to partitioning.

#### Trade-offs and Decisions

- Consistency vs. Availability: The library system sometime requires consistency and sometime availability. Authorizing strong consistency may decrease availability during peak periods or network complications, which is why moderation is important.

- Partition Tolerance: However, splitting the network is rare in a centralized structure; nonetheless, it is crucial to work on making the system ready for such cases and deliver uninterrupted service.

#### Strategies for Ensuring Data Consistency, System Availability, and Handling Partition Tolerance

- Data Consistency: Make use of high levels of ACID properties in transaction to guarantee soundness of data. Foreign keys and constraints ensure the maintenance of referential integrity and accurate representation of data relationships.

- System Availability: Use of redundancy and failover to increase availability. Daily backups of the system and distributed replicas of the database can assist in maintaining the working status of the system in failures.

- Handling Partition Tolerance: Always design the system so that it can bear transient faults and be at the final consistent state. Ensure application availability in the face of network partitions through data replication and conflict resolution techniques.

Taking into account the above-stated aspects of the CAP theorem, it will be possible to create the foundation for the library’s database system, which will be characterized by high availability, reliability, consistency, and immunity to possible network problems.

## Conclusion

### Summary of the Project

This project aimed to develop and integrate a fully functional database management system in a public library system utilizing MySQL. Major activities included the development of the ER diagram, creating the actual database design, populating the same with dummy data, and showcasing simple SQL queries for operations on the database. Moreover, the concept of CAP theorem was introduced with regard to the library system.

### Key Learnings and Takeaways

This particular project provided good practice in the area of database design, database implementation and in particular Advanced SQL Queries. We got to know about normalization, data integrity and the pros cons of CAP theorem talk. We also talked about how to enforce consistency of the data and availability of the system in the practical scenario of real project.

### Database Design and Management and Its Relevance to Practical Applications

Data base design plays an important role in planning and organizing data so that it might be stored, accessed and retrieved in an efficient manner. Database applications in real-world contexts, for example, a library system, have well-designed databases facilitating operations, optimizing users’ experiences, and handling accurate information, leading to the general efficiency and dependability of a system.

## References